

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

VL COLLECTIVE IP LLC,
Patent Owner.

Case No. IPR2023-00891

U.S. Patent No. 8,605,794 B2

**DECLARATION OF DR. BENJAMIN B. BEDERSON, PH. D.
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,605,794 B2**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	QUALIFICATIONS	2
III.	SCOPE OF OPINIONS.....	13
IV.	MATERIALS REVIEWED AND CONSIDERED	14
V.	LEVEL OF ORDINARY SKILL IN THE ART.....	15
VI.	STATEMENT OF LEGAL PRINCIPLES	16
	A. Claim Construction	16
	B. Anticipation.....	16
	C. Obviousness	17
VII.	TECHNOLOGY BACKGROUND AND DISCUSSION OF THE PRIOR ART	18
	A. Background of Multimedia Systems for Synchronizing Content.....	18
	B. U.S. Patent No. 7,386,782 to <i>Comps</i>	20
	C. U.S. Patent Application Publication No. 2004/0024898 to <i>Wan</i>	24
	D. U.S. Patent Application Publication No. 2004/0098398 to <i>Ahn</i>	26
	E. U.S. Patent Application Publication No. 2003/0101364 to <i>Kim</i>	28
	F. Other Evidence Regarding the State of the Art	29
VIII.	THE '794 PATENT	29
	A. Summary of the '794 Patent	29
	B. The '794 Patent File History.....	32
	C. Challenged Claims	35
	D. Claim Construction (37 C.F.R. § 42.104(b)(3)).....	37

IX.	PRIORITY DATE OF THE '794 PATENT	42
X.	APPLICATION OF THE PRIOR ART TO CLAIMS 1–22	43
A.	Ground 1: <i>Comps</i> Anticipates Claims 1-5, 9-13, 15-17, and 20-21 of the '794 Patent	43
1.	Claim 1	43
2.	Claim 2	53
3.	Claim 3	55
4.	Claim 4	57
5.	Claim 5	60
6.	Claim 9	61
7.	Claim 10	62
8.	Claim 11	63
9.	Claim 12	63
10.	Claim 13	64
11.	Claim 15	64
12.	Claim 16	65
13.	Claim 17	68
14.	Claim 20	68
15.	Claim 21	69
B.	Ground 2: <i>Comps</i> and <i>Wan</i> Render Obvious Claims 6-8, and 22	70
1.	Claim 6	70
2.	Claim 7	72
3.	Claim 8	74
4.	Claim 22	76
2.	Obviousness Based on <i>Comps</i> and <i>Wan</i>	80
C.	Ground 3: <i>Comps</i> and <i>Ahn</i> Render Obvious Claim 14 of the '794 Patent	83
1.	Claim 14	83

2.	Obviousness Based on <i>Comps</i> and <i>Ahn</i>	84
D.	Ground 4: <i>Comps</i> and <i>Kim</i> Render Obvious Claims 18 and 19 of the '794 Patent	87
1.	Claim 18	87
2.	Claim 19	88
3.	Obviousness Based on <i>Comps</i> and <i>Kim</i>	88
XI.	SECONDARY CONSIDERATIONS	90
XII.	CONCLUSION	91

EXHIBIT LIST

Exhibit Number	Description
1001	Patent-at-Issue: U.S. Patent No. 8,605,794 to Hellwagner, “Method for Synchronizing Content-dependent Data Segments of Files,” filed on March 16, 2006 (“’794Pat”).
1002	File History of the ’794 patent (Application No. 11/918,276 (“’794FH”).
1003	Declaration of Benjamin B. Bederson, Ph.D.
1004	U.S. Patent No. 7,386,782 to <i>Comps</i> , “Method for Synchronizing a Multimedia File,” filed on September 13, 2001 (“ <i>Comps</i> ”).
1005	U.S. Patent Application Publication No. 2004/0024898 to <i>Wan</i> , “Delivering Multimedia Descriptions,” filed on July 5, 2001 (“ <i>Wan</i> ”).
1006	U.S. Patent Application Publication No. 2004/0098398 to <i>Ahn</i> , “Method and Apparatus for Delivery of Metadata Synchronized to Multimedia Contents,” filed on January 30, 2002 (“ <i>Ahn</i> ”).
1007	U.S. Patent Application Publication No. 2003/0101364 to <i>Kim</i> , “Method of Ensuring Synchronous Presentation of Additional Data with Audio Data Recorded on a Rewriteable Recording Medium,” filed on November 27, 2002 (“ <i>Kim</i> ”).
1008	EP 1051008A2 to Jones, “Method and Apparatus for Media Data Transmission” filed on January 13, 1999 (“ <i>Jones</i> ”).
1009	U.S. Patent No. 7,827,297 to Kristjansson, “Multimedia Linking and Synchronization Method, Presentation and Editing Apparatus,” filed on January 18, 2003 (“ <i>Kristjansson</i> ”).
1010	U.S. Patent Application Publication No. 2005/0190872 to Seong, “Transcoding System and Method for Maintaining Timing Parameters before and after Performing Transcoding Process,” filed on February 10, 2005 (“ <i>Seong</i> ”).

1011	U.S. Patent No. 7,627,811 to Thong, “Content-based Synchronization Method and System for Data Streams,” filed on March 4, 2005 (“ <i>Thong</i> ”).
1012	U.S. Patent Application Publication No. 2003/0061369 to Aksu, “Processing of Multimedia Data,” filed on September 20, 2002 (“ <i>Aksu</i> ”).
1013	U.S. Patent Application Publication No. 2005/0165941 to Eytchison, “Methods and Apparatuses for Streaming Content,” filed on January 22, 2004 (“ <i>Eytchison</i> ”).
1014	U.S. Patent No. 5,724,605 to Wissner, “Method and Apparatus for Representing and Editing Multimedia Compositions Using a Tree Structure,” filed on March 31, 1995 (“ <i>Wissner</i> ”).
1015	U.S. Patent No. 5,515,490 to Buchanan, “Method and System for Temporally Formatting Data Presentation in Time-dependent Documents,” filed on November 5, 1993 (“ <i>Buchanan</i> ”).
1016	U.S. Patent Application Publication No. 2004/0064831 to Abbott, “System and Method for Media Stream Indexing and Synchronization,” filed on October 1, 2003 (“ <i>Abbott</i> ”).
1017	<i>Starz Entm’t, LLC v. VL Collective IP, LLC</i> , Amended Joint Claim Construction Chart, pages 1-16 (Dkt. 80).
1018	<i>Starz Entm’t, LLC v. VL Collective IP, LLC</i> , Joint Claim Construction Brief filed by Plaintiffs, pages 1-73 (Dkt. 78).
1019	<i>Starz Entm’t, LLC v. VL Collective IP, LLC</i> , Claim Construction Order (Dkt. 88)

I. INTRODUCTION

1. My name is Benjamin B. Bederson, Ph.D. I am currently Professor Emeritus of Computer Science at the University of Maryland. I have been retained by counsel for Netflix Inc. (“Netflix” or “Petitioner”) to submit this Declaration in support of Netflix’s petition for *Inter Partes* Review (“IPR”) and cancellation of claims 1–22 (the “Challenged Claims”) of U.S. Patent No. 8,605,794 (“the ’794 patent”) (Ex. 1001), which I understand has been assigned to VL Collective IP LLC (“VL” or “Patent Owner”).

2. I have been asked to opine on whether the ’794 patent is anticipated and/or rendered obvious by the prior art. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials. The materials that I considered for this Declaration include all exhibits cited in this Declaration.

3. I may rely upon these materials, my knowledge and experience, and/or additional materials to rebut arguments raised by Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

4. My analysis of the materials produced in this investigation is ongoing and I will continue to review any new material as it is provided. This Declaration represents only those opinions I have formed to date. I reserve the right to revise,

supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

5. I am being compensated at my standard consulting rate for my time spent working on issues in this matter. I am also being reimbursed for any reasonable and customary expenses associated with my work and testimony. I have no direct financial interest in, or affiliation with, Petitioner or Patent Owner. My compensation is not dependent upon the outcome of, or my testimony in, the present IPR or any litigation proceedings.

II. QUALIFICATIONS

6. My qualifications for forming the opinions set forth in this report are summarized here and are addressed more fully in my CV, which is attached hereto as Appendix A. I have included my referenced papers from this section in this appendix.

7. Based on my education, research, and work experience, I am knowledgeable about the subject matter of the '794 patent and the related prior art. I am currently Professor Emeritus of Computer Science at the University of Maryland ("UMD"). From 2014 to 2018, I was the Associate Provost of Learning Initiatives and Executive Director of the Teaching and Learning Transformation Center at the UMD. I am a member and previous director of the Human-Computer

Interaction Lab (“HCIL”), the oldest and one of the best-known Human-Computer Interaction (“HCI”) research groups in the country.

8. Last year, I co-founded the J.S. Bryant School, a therapeutic high school to open in 2025. I was also co-founder and Chief Scientist of Zumobi, Inc. from 2006 to 2014, a Seattle-based startup that is a publisher of content applications and advertising platforms for smartphones. I am also co-founder and co-director of the International Children’s Digital Library (“ICDL”), a web site launched in 2002 that provides the world’s largest collection of freely available online children’s books from around the world with an interface aimed to make it easy for children and adults to search and read children’s books online. I was also co-founder and prior Chief Technology Officer of Hazel Analytics from 2014-2023, a data analytics company to improve food safety and better public health whose product sends alerts in warranted circumstances. From 1992-1994, I worked in the Computer Graphics and Interactive Media group at Bell Communications Research – which is where I started my research in multimedia, including the development of Pad++.

9. In addition, I have for more than 25 years consulted for numerous companies in the area of user interfaces, including EPAM, Hillcrest Labs, Lockheed Martin, Logitech, Microsoft, NASA Goddard Space Flight Center, the Palo Alto Research Center, and Sony.

10. For more than 30 years, I have studied, designed, and worked in the field of computer science and HCI. That includes teaching, research, entrepreneurship, and administration in HCI and the software and technology underlying interactive computing systems. I have worked on the design and implementation of hardware and software systems including the use of capacitive, resistive, optical and IR among other sensors, and interactive applications on a range of devices, including embedded systems, controllers, smart phones, and PDAs for 2D and 3D interfaces.

11. In 1992, I completed my Ph.D. dissertation at New York University titled “A Miniature Space-Variant Active Vision System: Cortex-I” in which I built a custom small motor that aimed a custom camera along with a custom DSP-based computing system. Collectively, these worked together to create a miniature computer vision system that one could carry in one hand – dramatically smaller than other computer vision systems of the day. This system included the ability to generate a real-time video signal with the result of the image processing.¹

12. Starting in the mid-1990s and going through the 2000s, I worked on a range of “zoomable user interfaces” (ZUIs) that supported users in navigating large information spaces by zooming in and out of the information on the screen. I built

¹ Bederson, B. B. (1992). A Miniature Space-Variant Active Vision System: Cortex-I. Doctoral dissertation, New York University, New York, NY.

several different ZUI systems over the years that supported multimedia, including Pad++, Jazz, and Piccolo. For example, in my 1996 paper describing Pad++, on page 30, I explained “To this end, we are building a set of tools that will allow non-technical visual designers to create interactive zoomable multimedia systems.”² Starting in the early 2000s, I adapted that work to work on mobile devices including PDAs, tablets, and mobile phones.

13. Starting in 2001 and continuing for several years, I worked on photo management systems. I wrote a paper in 2001 describing PhotoMesa, a zoomable image browser using a novel 2D layout algorithm I called “quantum treemaps” to position the images on the screen in a way that attempted to fill the screen while keeping groups of related images together.³ As indicated in the figure below from that paper, PhotoMesa could display many hundreds of images at a time by showing small thumbnails, by showing larger thumbnails when the mouse hovered over a small thumbnail, and by allowing a user to zoom in to see a high resolution version

² Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, 3-31. <https://doi.org/10.1006/jvlc.1996.0002>

³ Benjamin B. Bederson. 2001. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST '01)*. Association for Computing Machinery, New York, NY, USA, 71–80. <https://doi.org/10.1145/502348.502359>

of the images. In later work, I worked on adding support for video and other media to PhotoMesa. I also worked on other multimedia browsers, as I described in this paper where I said “The approach started in ZIB was continued in a new project that is creating an interface for elementary school-aged children to find multimedia information in a digital library.” *Id.*, p. 73.

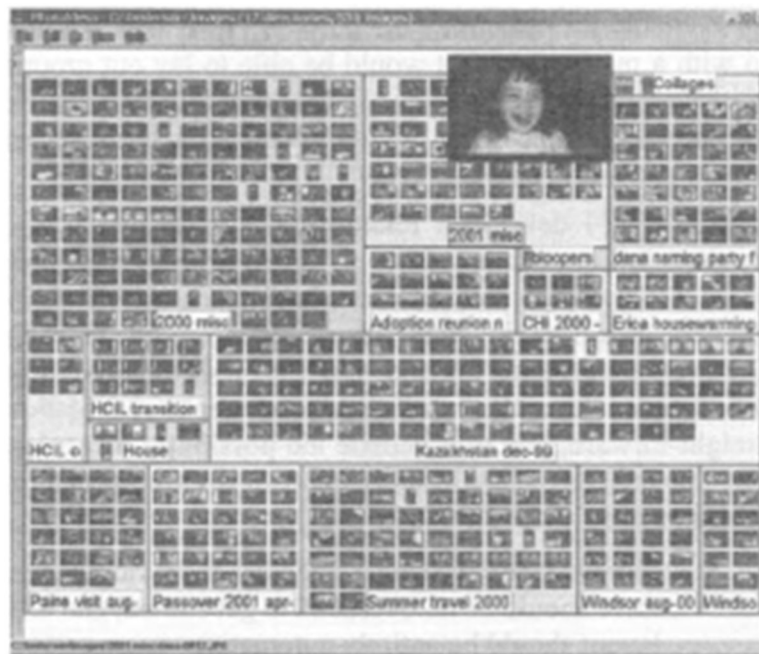


Figure 1: Screen snapshot of PhotoMesa with over 500 images in 17 groups.

14. In 2003, I wrote a paper that performed image processing to identify salient portions of images and then automatically crop those images to show those portions.⁴ In this way, image thumbnails could focus attention on faces or other

⁴ Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. 2003. Automatic thumbnail cropping and its effectiveness. In Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03). Association for Computing Machinery, New York, NY, USA, 95–104. <https://doi.org/10.1145/964696.964707>

objects of a photo without showing all of the background. In 2006, I wrote a paper that explained some of the trends in photo management software including rich support for annotation, browsing and sharing of photos.⁵ In 2007, I published another paper that provided a review of innovative personal photo management work going on at that time.⁶ That paper describes a version of my PhotoMesa software that by 2007 as shown in the following figure, included the ability to annotate photos with the location and name of people in photos.

⁵ Ben Shneiderman, Benjamin B. Bederson, and Steven M. Drucker. 2006. Find that photo! interface strategies to annotate, browse, and share. *Commun. ACM* 49, 4 (April 2006), 69–71. <https://doi.org/10.1145/1121949.1121985>

⁶ Hyunmo Kang, Benjamin B. Bederson & Bongwon Suh (2007) Capture, Annotate, Browse, Find, Share: Novel Interfaces for Personal Photo Management, *International Journal of Human–Computer Interaction*, 23:3, 315-337, DOI: 10.1080/10447310701702618



FIGURE 1 The revised direct annotation mechanisms as implemented in PhotoMesa, the successor to PhotoFinder. *Note.* Users can add a caption under a photo, or add a particular attribute such as favorite (a yellow star on the bottom left of photo) or hidden. In addition, labels can be dragged from the list of people (on the top left) or from the user-defined category tree (on the bottom left). A label can be directly placed on the photo to represent where the individuals or objects are located within the photo.

15. I was heavily involved in the research and development of mobile device applications during the early 2000's. For example, I envisioned, designed and built DateLens in 2002-2004 to create a richer calendar for the Microsoft PocketPC platform as well as desktops as described in a 2004 paper.⁷ One of its innovations

⁷ Benjamin B. Bederson, Aaron Clamage, Mary P. Czerwinski, and George G. Robertson. 2004. DateLens: A fisheye calendar interface for PDAs. ACM Trans.

was its design that enabled it to scale between small (mobile) and large (desktop) computers. It also had a special double-headed scroll bar that allowed a user to touch scroll and also control how much information was presented on the screen at a time. I started DateLens as a research project and eventually sold it commercially. DateLens displayed calendar data that came from the Microsoft PocketOutlook database and allowed modification and viewing of appointments. The research website from that time period that describes DateLens is still available at <http://www.cs.umd.edu/hcil/datelens/> which shows screenshots and a picture of me demonstrating DateLens to Bill Gates. The commercial website from that time period is also available at <http://www.windsorinterfaces.com/datelens.shtml>, which shows a number of news articles reviewing DateLens at that time.

16. In 2004-2005, with graduate student Amy Karlson and John SanGiovanni of Microsoft, I worked on a prototype software called LaunchTile. LaunchTile allowed zooming and scrolling of multiple applications on a handheld touchscreen PDA. The interface offered a grid of applications that could be zoomed and scrolled. It included several different methods of scrolling, including gestures

where the scrolling continued after a finger was lifted from the touch screen, and then slowed to a stop.⁸

17. That work led to my co-founding of Zumobi, where I was responsible for investigating new software platforms and developing new user interface designs that provide efficient and engaging interfaces for mobile touchscreen platforms including the iPhone and Android-based devices. For example, I designed and implemented software called “Ziibii,” a “river” of news for iPhone, software called “ZoomCanvas,” a zoomable user interface for several iPhone apps, and iPhone apps including “Inside Xbox” for Microsoft and Snow Report for REI. All of these applications included touch-based zooming and scrolling.

18. At UMD, I am focused primarily on the area of HCI, a field that relates to the development and understanding of computing systems to serve users’ needs. Researchers and practitioners in this field are focused on making universally usable, useful, efficient, and appealing systems to support people in their wide range of activities. My approach is to balance the development of innovative technology that serves people’s practical needs. Example systems following this approach that I have built include Cortex-I (1992 embedded computer vision system that sensed licensed

⁸ Amy K. Karlson, Benjamin B. Bederson, and John SanGiovanni. 2005. AppLens and launchTile: two designs for one-handed thumb use on small devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05). Association for Computing Machinery, New York, NY, USA, 201–210. <https://doi.org/10.1145/1054972.1055001>

plates with custom motor, camera, and controller), Audio Augmented Reality (1995 embedded system for sensing a user's location and playing audio suited to that location), Fisheye Menus (2000 software for sensing movement within and selection of linear list of items in a menu), PhotoMesa (2001 software for end users to browse personal photos), DateLens (2002 software for end users to use their mobile devices to efficiently access their calendar information), SlideBar (2005 linear sensor to control scrolling), LaunchTile (2005 "home screen" software for mobile devices to allow users to navigate apps in a zoomable environment), SpaceTree (2001 software for end users to efficiently browse very large hierarchies), ICDL (as described above), and StoryKit (a 2009 iPhone app for children to create stories).

19. My work has been published extensively in more than 160 technical publications, and I have given about 100 invited talks, including 9 keynote lectures. I have won a number of awards including the Brian Shackel Award for "outstanding contribution with international impact in the field of HCI" in 2007, and the Social Impact Award in 2010 from the Association for Computing Machinery's ("ACM") Special Interest Group on Computer Human Interaction ("SIGCHI"). ACM is the primary international professional community of computer scientists, and SIGCHI is the primary international professional HCI community. I have been honored by both professional organizations. I am an "ACM Distinguished Scientist," which "recognizes those ACM members with at least 15 years of professional experience

and 5 years of continuous Professional Membership who have achieved significant accomplishments or have made a significant impact on the computing field.” I am a member of the “CHI Academy,” which is described as follows: “The CHI Academy is an honorary group of individuals who have made substantial contributions to the field of HCI. These are the principal leaders of the field, whose efforts have shaped the disciplines and/or industry, and led the research and/or innovation in human-computer interaction.” The criteria for election to the CHI Academy are: (1) cumulative contributions to the field; (2) impact on the field through development of new research directions and/or innovations; and (3) influence on the work of others. I have received two “Test of Time” awards from IEEE InfoVis for my 2001 and 2002 work on visualizing hierarchies, including those containing digital photos.

20. I have appeared on radio shows numerous times to discuss issues relating to user interface design and people’s use and frustration with common technologies, web sites, and mobile devices. My work has been discussed and I have been quoted by mainstream media around the world over 120 times, including by the New York Times, the Wall Street Journal, the Washington Post, Newsweek, the Seattle Post Intelligencer, the Independent, Le Monde, NPR’s All Things Considered, New Scientist Magazine, and MIT’s Technology Review.

21. I have designed, programmed, and publicly deployed dozens of user-facing software products that have cumulatively had millions of users. My work is

cited by several major companies, including Amazon, Apple, Facebook, Google, and Microsoft. I am a named inventor on 14 U.S. patents and 20 U.S. patent applications. The patents are generally directed to user interfaces and experience.

22. I received a B.S. degree in Computer Science with a minor in Electrical Engineering in 1986 from the Rensselaer Polytechnic Institute. I received M.S. and Ph.D. degrees in Computer Science in 1989 and 1992, both from New York University.

III. SCOPE OF OPINIONS

23. I have been asked to provide my opinions with respect to claims 1-22 of the '794 patent (the "challenged claims"), specifically regarding whether:

- Claims 1-5, 9-13, 15-17, and 20-21 of the '794 patent are anticipated by *Comps* (Ex. 1004) (Ground I);
- Claims 6-8, and 22 of the '794 patent would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention in view of *Comps* (Ex. 1004) and *Wan* (Ex. 1005) (Ground II);
- Claim 14 of the '794 patent would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention in view of *Comps* (Ex. 1004) and *Ahn* (Ex. 1006) (Ground III); and

- Claims 18 and 19 of the '794 patent would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention in view of *Comps* (Ex. 1004) and *Kim* (Ex. 1007) (Ground IV).

24. This Declaration, including the exhibits cited herein, sets forth my opinions on these topics. In short, the challenged claims are unpatentable because they would have been anticipated or obvious to a POSITA at the time the '794 patent was filed. Specifically, it is my opinion that *Comps* anticipates claims 1-5, 9-13, 15-17, and 20-21 of the '794 patent; *Comps* and *Wan* render obvious claims 6-8, and 22 of the '794 patent; *Comps* and *Ahn* render obvious claim 14 of the '794 patent; and *Comps* and *Kim* render obvious claims 18 and 19 of the '794 patent.

25. I reviewed and contributed to the Petition's explanation as to why the challenged claims are unpatentable. The Petition sets forth my reasons for my opinion that the challenged claims are anticipated or would have been obvious. Below I elaborate on certain points raised in the discussion of the Petition from the perspective of a POSITA at the time of the '794 patent's priority date.

IV. MATERIALS REVIEWED AND CONSIDERED

26. The materials I considered and relied upon in preparing my Declaration and forming my opinions include all exhibits cited to in this Declaration, including the '794 patent, the '794 file history, and the relevant prior art, as I discuss below.

27. I also have relied on my academic and professional experience in reaching the opinions expressed in this Declaration.

V. LEVEL OF ORDINARY SKILL IN THE ART

28. In rendering the opinions set forth in this Declaration, I was asked to consider the patent claims and the prior art through the eyes of a POSITA. The “art” is the field of technology to which a patent is related. I understand that the purpose of using the viewpoint of a POSITA is for objectivity.

29. I considered factors such as the educational level and years of experience of those working in the pertinent art; the types of problems encountered in the art; the teachings of the prior art; patents and publications of other persons or companies; and the sophistication of the technology. I understand that a POSITA is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors discussed above.

30. Taking these factors into consideration, it is my opinion that a POSITA as of the time of the alleged invention of the '794 patent would have had a bachelor's degree in electrical or computer engineering, or a closely related scientific field such as computer science, and two years of work experience with multimedia systems. Alternatively, any lack of experience could be remedied with additional education (*e.g.*, a master's degree), and likewise, a lack of education can be remedied with additional work experience (*e.g.*, 4-5 years).

VI. STATEMENT OF LEGAL PRINCIPLES**A. Claim Construction**

31. I understand that terms appearing in the patent claims are to be interpreted according to their “ordinary and customary meaning” in an IPR proceeding. In determining the ordinary and customary meaning, the words of a claim are first given their plain meaning as they would have been understood by a POSITA at the time of the alleged invention, in light of the specification and file history. I understand that treatises and dictionaries may be consulted, albeit under limited circumstances, to determine the meaning attributed by a POSITA at the time of the alleged invention. I have followed this approach in my analysis and have applied the ordinary and customary meaning of those terms throughout my analysis in this Declaration.

32. I understand that the words of the claims should be interpreted as they would have been understood by a person of ordinary skill in the art at the time the alleged invention was made (not today).

B. Anticipation

33. I understand that if each and every element of a claim is disclosed in a single prior art reference, then the claimed invention is anticipated and not patentable under pre-AIA 35 U.S.C. § 102. In order for the invention to be anticipated, each element of the claimed invention must be described or embodied, either expressly or inherently, in a single prior art reference. I also understand that a reference

inherently discloses a claim element when that claim element is necessarily present in the reference.

C. Obviousness

34. I have been informed that a patent claim is invalid as “obvious” under pre-AIA 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to one of ordinary skill in the art, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so-called “secondary considerations” of non-obviousness, which include: (i) “long felt need” for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others.

35. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known methods is

likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

VII. TECHNOLOGY BACKGROUND AND DISCUSSION OF THE PRIOR ART

36. It is my opinion that all of the elements of claims 1-22 were already known in the prior art before the priority date of the '794 patent. Accordingly, I conclude that there is nothing novel or non-obvious about the alleged invention disclosed in the '794 patent.

37. I have had input into, reviewed, and endorse the discussions in the Background of Multimedia Systems for Synchronizing Contents and Identification of Prior Art sections of the Petition (sections IV.B and V), which I largely include here.

A. Background of Multimedia Systems for Synchronizing Content

38. Systems for synchronizing multimedia content (*e.g.*, video and audio content) were well-known by the mid-2000s. Prior to 2005, multiple techniques for synchronizing multimedia contents existed, including synchronizing different types of multimedia data including two or more separate data files. *Comps*, 1:4-5, 2:6-9; *Wan*, [0065]; *Ahn*, [0003]; *Kim*, [0002]; *Kristjansson*, 9:64-67; *Seong*, [0003]; *Wissner*, 4:10-19; *Abbott*, [0056]. There were techniques by which metadata are synchronized with associated multimedia contents. *Ahn*, [0003]; *Seong*, [0003].

There were also techniques by which descriptions are synchronized with associated audio files and video files using an internal time graph constructed by an encoder. Ex. 1005 at [0043-0045]. Such techniques included temporal and non-temporal synchronization methods. *Comps*, 1:41-41, 3:44-47, 4:19-36; *Wan*, [0049]; *Kim*, [0018]; *Thong*, Title, 1:62-2:13. Some techniques used predefinable rules for assigning one type of data segments to another type of data segments. *Comps*, 4:3-17, Fig. 3; *Kristjansson*, 7:12-24; *Buchanan*, 11:59-67. Prior to the '794 patent, those in the art understood there was a need for temporal and non-temporal synchronization methods, and such methods were already developed. *Comps*, 1:26-41, 2:20-23, 2:48-51, 3:44-47, 4:19-36, Abstract.

SMIL (Synchronized Multimedia Integration Language) has been a W3C standard since 1998 when version 1.0 was defined (<https://www.w3.org/TR/SMIL1/>). Version 2.1 was defined by December 2005 (<https://www.w3.org/TR/2005/REC-SMIL2-20051213/>). The standard defines “an XML-based language that allows authors to write interactive multimedia presentations” including “temporal behaviour of a multimedia presentation” that can “represent timing and synchronization.” (Abstract). The standard includes the ability to describe markers that indicate scene boundaries as shown in an example in section 8.5. The Timing and Synchronization Module (<https://www.w3.org/TR/2005/REC-SMIL2-20051213/smil-timing.html>) describes

how SMIL solves “fundamental media synchronization problems and defined a powerful way of choreographing multimedia content” (10.2 Introduction). A basic part of SMIL's timing description is the ability to describe how media elements are synchronized by playing them in sequence, in parallel, or without specified order - in a way that does not require the use of timestamps. For example, SMIL could synchronize a video and audio element to play in parallel without using a timestamp. (Section 10.4.1 Attributes).

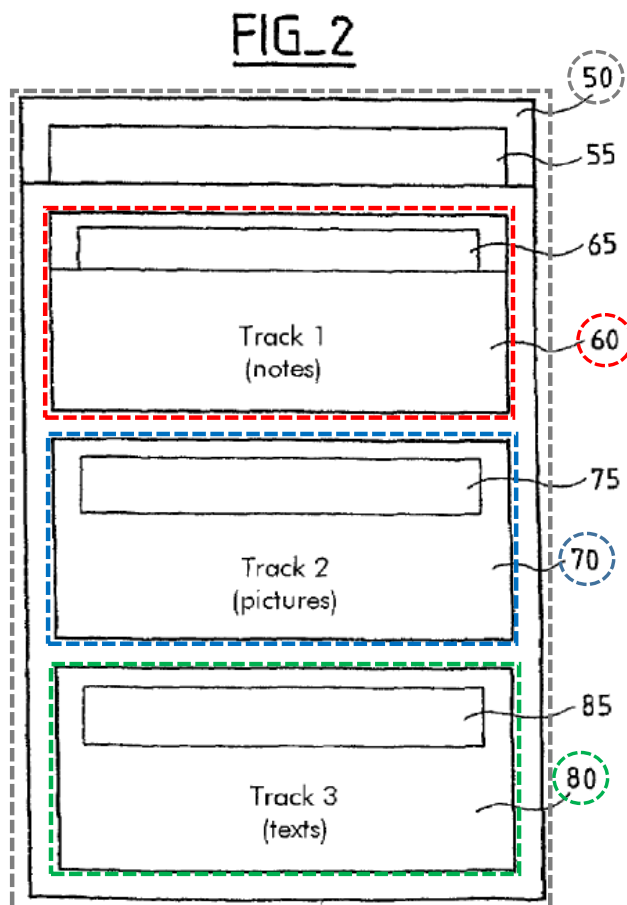
```
<par endsync="last">  
  <video src="intro.mpg" begin= "5s" dur="30s" fill="freeze" />  
  <audio src="intro.au" begin= "2s" dur="40s"/>  
</par>
```

B. U.S. Patent No. 7,386,782 to *Comps*

39. *Comps* is prior art under pre-AIA § 102(a) and (e) because it was filed on September 13, 2001 and issued on June 10, 2008. I understand that *Comps* was not disclosed to or considered by the Patent Office during prosecution of the '794 patent.

Comps discloses “a method of synchronizing different types of data in a multimedia file.” *Comps*, 1:4-5; *see also id.* at Abstract. *Comps* describes a method of synchronizing multimedia data including “at least two separate data files (track1, track2) referred to as the first file, the second file, etc.” *Id.*, 2:6-9. In an embodiment shown in Fig. 2, *Comps* discloses that a multimedia file 50 includes

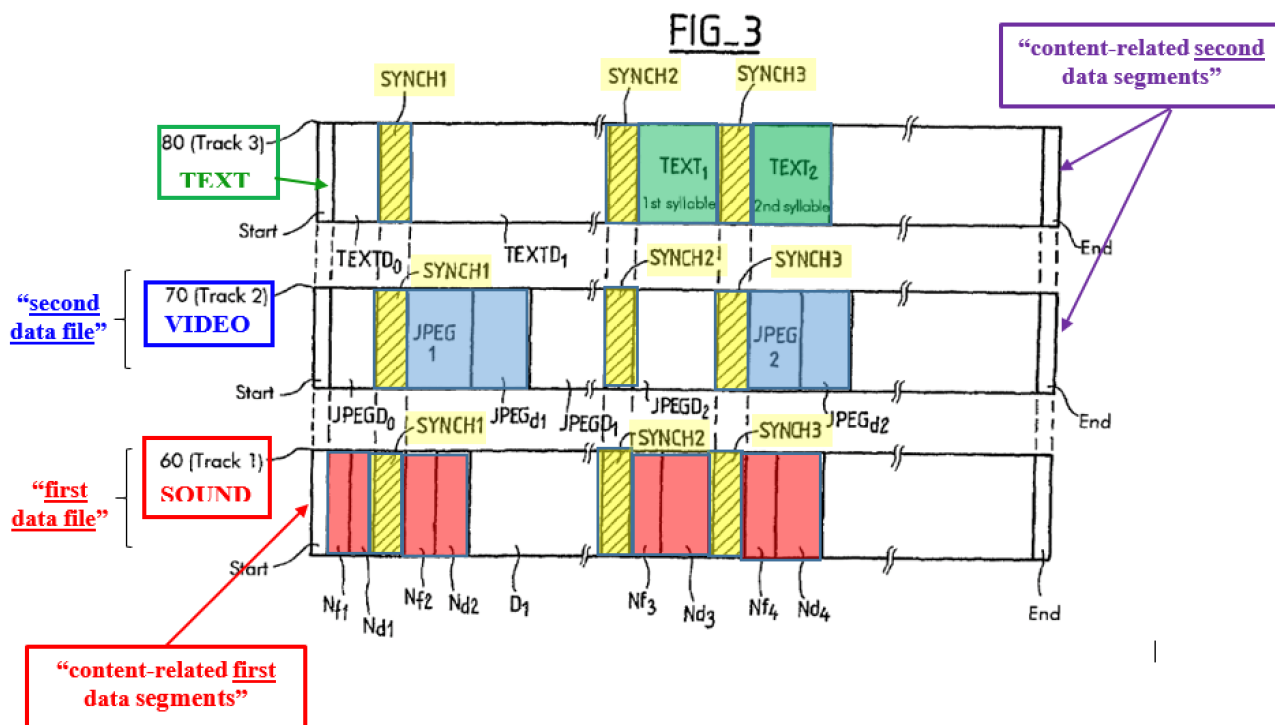
three tracks 60, 70, 80. *Id.*, 2:54-55. Each track contains a different type of content such as *e.g.*, the track 60 for a sequence of musical notes, the track 70 for a sequence of pictures (or video data), and the track 80 for a sequence of texts. *Id.*, 2:60-64, 3:8-19, 3:48-49, Fig. 2. The tracks 60, 70, 80 include synchronization commands related to events included in the track (note, picture, text, etc.) that are executed sequentially by a microprocessor to, for example, play music and display a picture or a text. *Id.*, 2:47-50, 3:5-7; 3:48-49. Each track 60, 70, 80 has a respective header 65, 75, 85 that includes an indicator of the type of data contained in the track. *Id.*, 3:1-3. Figure 2 (reproduced below) from *Comps* illustrates the above-discussed structure of the multimedia file 50.



Comps, Fig. 2 (annotated).

40. As shown in Figure 3 (reproduced below), each track 60, 70, 80 may include corresponding data segments, *e.g.*, sound segments as represented by fields $Nf_1:Nd_1$, $Nf_2:Nd_2$, $Nf_3:Nd_3$, $Nf_4:Nd_4$ for the sound track 60, picture segments as represented by fields $JPEG_1:JPEG_{d1}$, $JPEG_2:JPEG_{d2}$ for the picture track 70, and text segments as represented by fields $TEXT_1$, $TEXT_2$ for the text track 80. *Id.*, 3:22-41. Nf_i and Nd_i represent the frequency of i -th note and its duration, respectively; $JPEG_i$ and $JPEG_{di}$ represents i -th JPEG image for a given time represented by $JPEG_{di}$; and

TEXT_i represents i-th text message. *Id.* The synchronization commands are represented by fields SYNCH_i for i for 1 to n. *Id.*, 3:42-43.



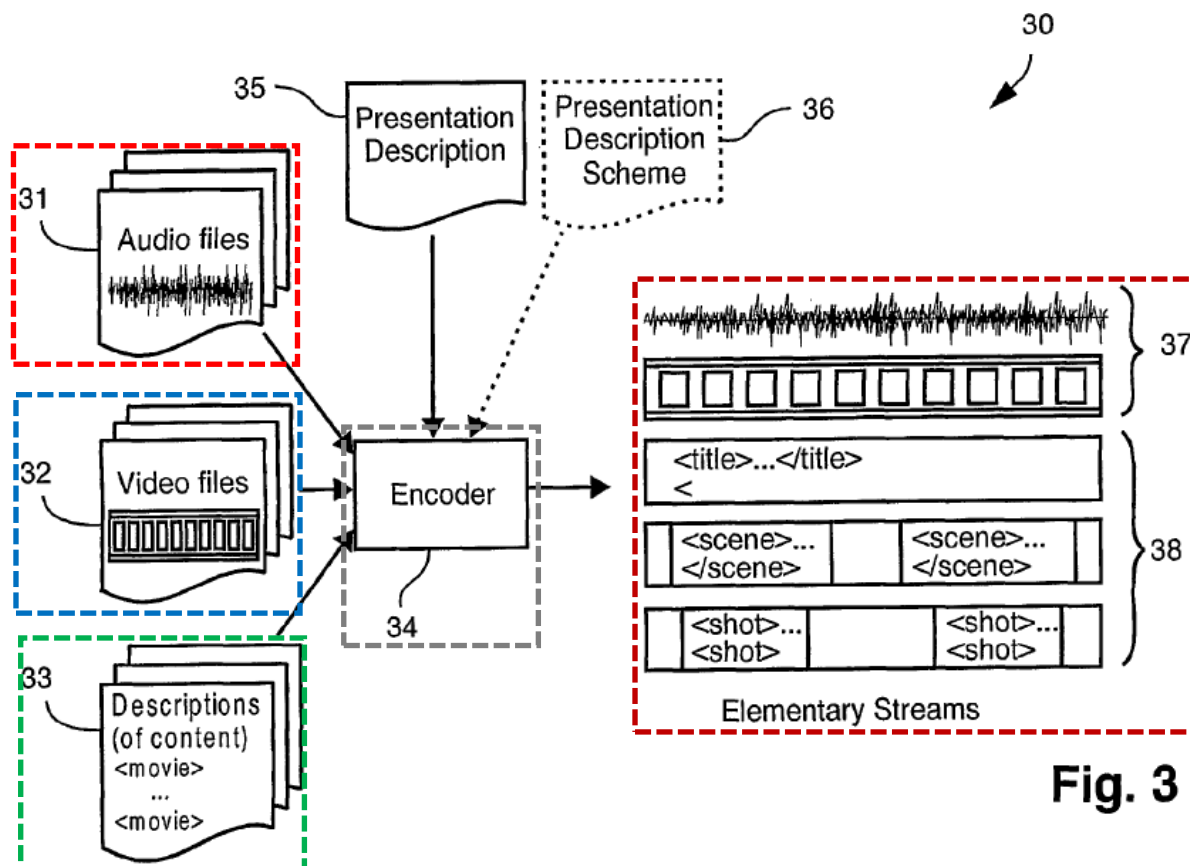
Comps, Fig. 3 (annotated).

41. As shown in Figure 3, based on the "SYNCH₁" command, the second note (Nf₂:Nd₂) must be heard at the moment the first picture (JPEG₁) is displayed; based on the "SYNCH₂" command, the third note (Nf₃:Nd₃) must be heard at the moment the first syllable (TEXT₁) is displayed; and based on the "SYNCH₃" command, the fourth note (Nf₄:Nd₄) must be heard at the moment the second picture (JPEG₂) and the second syllable (TEXT₂) are simultaneously displayed. *Id.*, 3:57-4:2.

C. U.S. Patent Application Publication No. 2004/0024898 to *Wan*

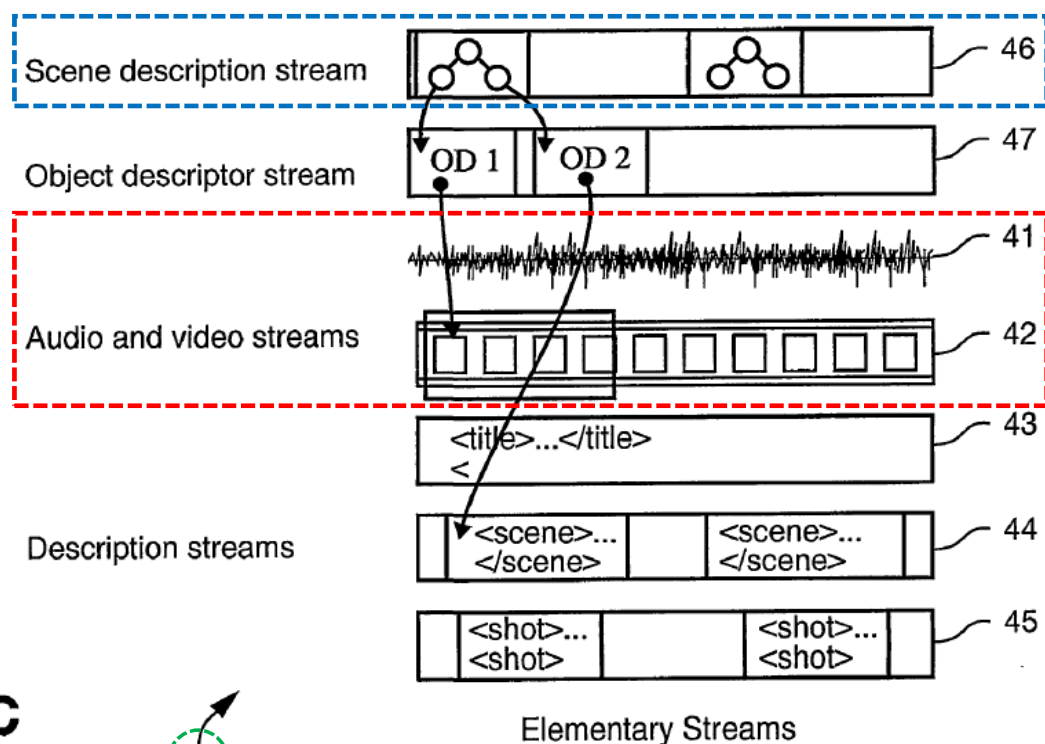
42. *Wan* is prior art under pre-AIA § 102(a), (b), and (e) because it was filed on July 5, 2001 and published on February 5, 2004. *Wan* was not disclosed to or considered by the Patent Office during prosecution of the '794 patent.

43. *Wan* is directed to “provid[ing] . . . streaming descriptions with AV (audio-visual) content.” *Wan*, [0010]. *Wan* discloses, referring to Figure 3 (reproduced below), that descriptions 33 are synchronized with associated audio files 31 and video files 32 using an internal time graph constructed by an encoder 34. *Id.*, [0043-0045]. Based on this time graph, the encoder 34 schedules the delivery of the required components and then generates elementary data streams 37, 38 (*e.g.*, for the audio and video data) to be transmitted. *Id.*, [0045]. The encoder 34 also splits the descriptions 33 of the content into multiple data streams 38. *Id.*

**Fig. 3**

Wan, at Fig. 3 (annotated).

44. As shown in Fig. 4C (reproduced below), a multimedia stream 40 includes an audio stream 41, a video stream 42, and a scene description stream 46. *Id.*, [0053]. *Wan* discloses that the scene description stream 46 can be used for transmitting scene graphs and updates to scene graphs. *Id.*, [0047]. When the description has some temporal relationship with the content (*e.g.*, a description of a particular scene in a movie that provides for multiple camera angles to be viewed), the temporally irregular relationship between description and content may be accommodated through a temporal object description structured into the streams. *Id.*, [0053-0054].

**Fig. 4C**

Wan, Fig. 4C (annotated).

D. U.S. Patent Application Publication No. 2004/0098398 to *Ahn*

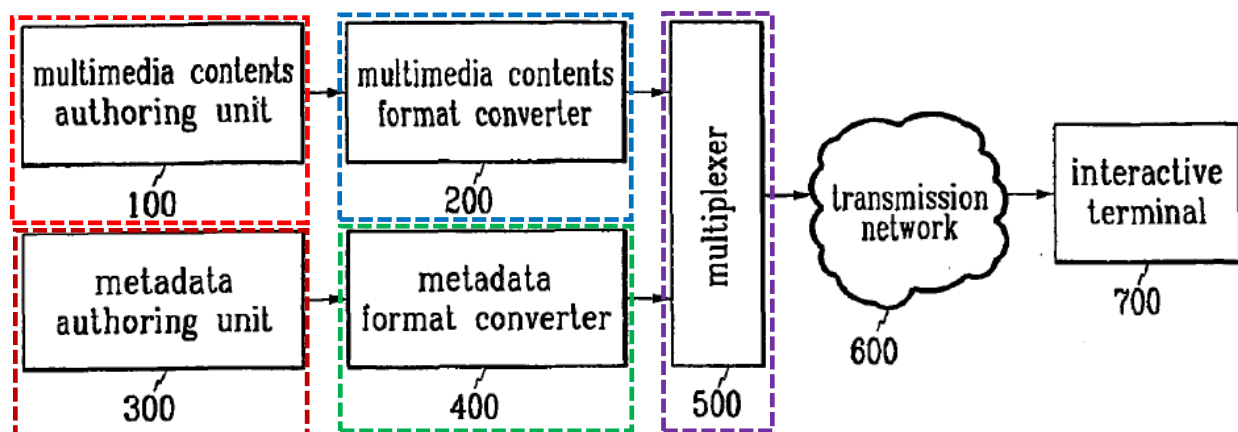
45. *Ahn* is prior art under pre-AIA § 102(a) and (e) because it was filed on January 30, 2002 and published on May 20, 2004. *Ahn* was not disclosed to or considered by the Patent Office during prosecution of the '794 patent.

46. *Ahn* is directed to a “[m]ethod and apparatus for delivery of metadata synchronized to multimedia contents.”⁹ *Ahn*, Title. *Ahn* discloses, referring to Figure 1 (reproduced below), a metadata transmission system that includes a

⁹ The specification of the '794 patent describes that “video or audio data, and/or metadata” are examples of media data. '794Pat, 3:4-8.

multimedia contents format converter 200 and a metadata format converter 400. *Id.*, [0019]. The multimedia contents format converter 200 compresses the multimedia contents input from a multimedia contents authoring unit 100, converts them into transmission format data for synchronization and transmission, and outputs them to a multiplexer 500. *Id.*, [0021]. The metadata format converter 400 converts the metadata input from the multimedia contents authoring unit 100 into binary codes, converts them into a transmission format for synchronization and transmission, and outputs them to the multiplexer 500. *Id.*, [0025]. A metadata synchronization format converter 420 converts the XML-language metadata transmitted from a metadata authoring unit 300 into binary codes, and converts them into a synchronization format. *Id.*, [0030].

FIG. 1

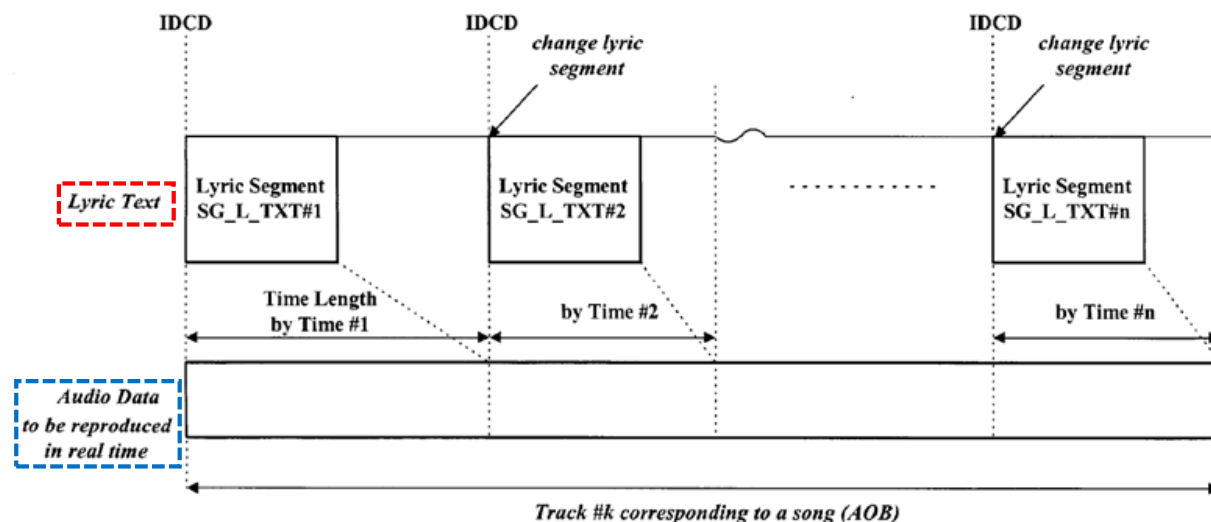


Ahn, Fig. 1 (annotated).

E. U.S. Patent Application Publication No. 2003/0101364 to *Kim*

47. *Kim* is prior art under pre-AIA § 102(a), (b) and (e) because it was filed on November 27, 2002 and published on May 29, 2003. *Kim* was not disclosed to or considered by the Patent Office during prosecution of the '794 patent.

48. *Kim* is directed to a “a method of recording additional data such as l[lyric and user input data to be in synchronization with audio data on a rewritable recording medium, and of reproducing them synchronously therefrom.” *Kim*, [0002]. *Kim* discloses that lyric or user data related with a song recorded on a rewritable disk such as DVD-RW is segmentized into several units. *Id.*, [0030]. The segmentized lyric or user data may be recorded in a file separated from the management information file storing management information for controlling reproduction of all of recorded audio data. *Id.*, cl. 7. Fig. 5 (reproduced below) illustrates continuous and synchronous display of a series of lyric segments together with reproduced audio data.

FIG. 5

Kim, Fig. 5 (annotated).

F. Other Evidence Regarding the State of the Art

49. The prior art in Exhibits 1008-1016 further reflects the state of the art, level of ordinary skill in the art, common knowledge in the art, and/or common sense in the art, and is therefore also relevant to the background of the invention and the invalidity analysis herein.

VIII. THE '794 PATENT

A. Summary of the '794 Patent

50. I have had input into, reviewed, and endorse the discussions in the Summary section of the Petition (section IV.B), which I include here.

51. The '794 patent summarizes the invention as "a method or [] a device which enables content-related first and second data segments to be synchronized in a simple and standard-compliant manner." '794Pat, 2:24-27. According to the

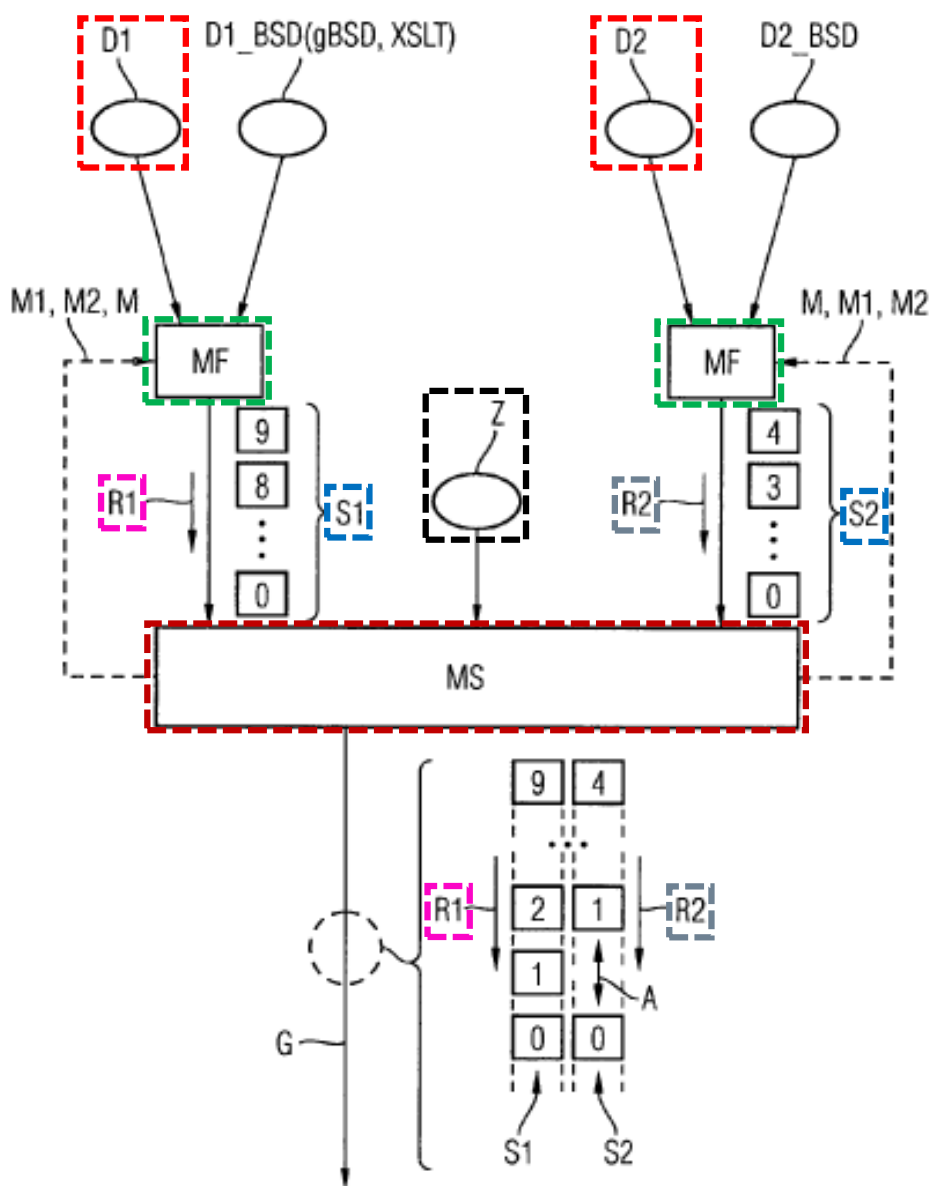
patent, the content-related first and second data segments are respectively output in sequence according to their chronological order based on a predeterminable assignment rule. *Id.*, 2:28-37; *see also id.*, Abstract. Every independent claim of the '794 patent requires that each of content-related second data segments be output together with an associated one of the content-related first data segments based on an assignment rule for assigning each one of the content-related second data segments to one of the content-related first data segments. *Id.*, Claims 1, 9, 22; *see also id.*, 2:28-37.

52. More specifically, the patent describes, referring to Fig. 1, that a video data stream and an audio data stream are synchronized using the method of the purported invention. *Id.*, 3:65-4:8. Content-related first and second data segments S1, S2 are obtained from a first data file D1 (*e.g.*, video data stream) and a second data file D2 (*e.g.*, audio data stream), respectively. *Id.*, 4:1-14. As shown in Fig. 1, the first data segments S1 are passed by a fragmentation module MF to a synchronization module MS in the same chronological sequence R1 in which they are to be output. *Id.*, 4:42-49. Similarly, the second data segments S2 are passed by the fragmentation module MF to the synchronization module MS in the same chronological sequence R2 in which they are to be output. *Id.*, 5:5-9. Subsequently, the first and second data segments S1, S2 are synchronized on the basis of a predefinable assignment rule Z. *Id.*, 5:10-13. The specification of the '794 patent

describes that this method is advantageous since the assignment is carried out without the use of timestamps. *Id.*, 2:36-43; *see also id.*, Claims 20, 21.

53. Figure 1 (reproduced below) from the patent illustrates above-discussed elements:

FIG 1



'794*Pat*, Fig. 1 (annotated).

54. The patent describes that the assignment rule can be described “merely by specification of the number” such that a second data segment S2 is assigned precisely to a first data segment S1 and a further assignment is carried out following a number of sequentially succeeding first data segments S1. *Id.* at 2:47-54. The specification of the '794 patent describes that this method is advantageous since the very low signaling overhead is required for signaling the number. *Id.*

B. The '794 Patent File History

55. I have had input into, reviewed, and endorse the discussions in the Prosecution History section of the Petition (section IV.C), which I include here.

56. The '794 patent issued from U.S Patent Application No. 11/918,276 (“’276 Application”), filed October 11, 2007, which was a national stage application of PCT/EP2006/060808, filed March 16, 2006, which claims priority to a German Patent Application, DE 10 2005 016 866, filed April 12, 2005. *See* '794*Pat*, field (21, 22, 30, 86); '794*FH*, 445.

57. In a non-final Office Action dated March 7, 2012, claims 1-17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over EP 1494430 (“Shin”) in view of U.S. Patent No. 5,661,523 (“Yamane”). '794*FH*, 280. Claims 1-4 and 7 were rejected under 35 U.S.C. § 101. *Id.*, 278. In response to the rejections, the applicant amended independent claims 1 and 9 to recite that the assignment rule was

for “assigning each second data segment to a first data segment” (*id.*, 262, 263), and argued that the cited art did not disclose this element (*id.*, 268–271). The applicant added new claims 18-21. *Id.*, 264.

58. In a final Office Action dated June 29, 2012, claims 1-21 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,598,352 (“Rosenau”) in view of Yamane. *Id.*, 226. Claims 1-4, 7, 16, 18, and 20 were rejected under 35 U.S.C. § 101. *Id.*, 224.

59. On October 10, 2012, the applicant conducted an interview with the Examiner to discuss the rejections, in which a proposed amendment to claim 1 was presented. *Id.*, 209-215. The Examiner stated that both Rosenau and a newly identified prior art reference, U.S. Patent No. 7,099,848 (“Bratton”) each disclosed the proposed amendment to claim 1 that recited “wherein the assignment rule is formed in such a way that, for each second data segment, a first data segment is assigned by specification of a number, the number representing a position of the first data segment within a chronological sequence of the first data segments.” *Id.*, 212.

60. On November 29, 2012, the applicant filed a Request for Continued Examination (“RCE”) along with an amendment, different from the previously proposed amendment, in which independent claims 1 and 9 were amended to positively recite that the first and second data segments were “content-related.” *Id.*, 193-196. With respect to the prior art, the applicant argued that the cited art did not

disclose an “sequentially outputting the content-related first data segments and the content-related second data segments . . . **on the basis of an assignment rule** for assigning each one of the content-related second data segments to one of the content-related first data segments,” as recited in amended claim 1. *Id.*, 203. The applicant added new independent claim 22. *Id.*, 196.

61. Subsequently, the Examiner issued another Office Action dated May 8, 2013, maintaining the § 101 rejection and withdrawing the prior art rejection. *Id.*, 154–162. In allowing claims 1-22 over the prior art, the Examiner stated “[n]o reason for allowance is needed as the record is clear in light of applicant’s arguments and amendment filed on 11/29/2012[.]” *Id.*, 160.

62. On June 28, 2013, the applicant filed an amendment to independent claims 1, 9, and 22, *e.g.*, amending claim 1 to recite that the outputting step was performed “by a device for synchronizing content-related data” to overcome the § 101 rejection. *Id.*, 142–149.

63. On August 1, 2013, the Examiner issued a Notice of Allowance. *Id.*, 30-41. In the Examiner’s statement of reason of allowance, the Examiner stated that “the closest prior art, [] Rosenau” disclosed “video data synchronized with the audio data” and another “closest prior art, [] Yamane” disclosed “frames are displaying in the chronological order” (*id.*, 37-40), but these disclosures are not equivalent to “sequentially outputting data segments on the basis of an assignment rule for

assigning [] each one of the content-related second data segment to one of the content-related first data segments” as recited in the allowed claims. *Id.*

C. Challenged Claims

64. The '794 patent has three independent claims, claims 1, 9 and 22, and nineteen dependent claims, claims 2-8 and 10-21. I have been asked to opine on all of claims 1-22, which are listed below:

1[pre] A method for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, the method comprising:
1[a][1] sequentially outputting, by a device for synchronizing content-related data, the content-related first data segments and the content-related second data segments according to their chronological sequence in such a way that
1[a][2] each of the content-related second data segments is output together with an associated one of the content-related first data segments on the basis of an assignment rule for assigning each one of the content-related second data segments to one of the content-related first data segments.
2 The method as claimed in claim 1, wherein the assignment rule is formed in such a way that each of the content-related second data segments are assigned to precisely one of the content-related first data segments and a further assignment is carried out after a number of sequentially succeeding content-related first data segments.
3 The method as claimed in claim 1, wherein the assignment rule is formed in such a way that, for each of the content-related second data segments, the content-related first data segments are assigned by specification of a number, the number representing a position of the content-related first data segments within the order of content-related first data segments.
4 The method as claimed in claim 1, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments and the content-related second data segments are synchronized on the basis of a combined assignment rule.
5 The method as claimed in claim 1, wherein media data are represented by at least one of the first data file and the second data file.

6 The method as claimed in claim 1, wherein at least one of the content-related first data segments and content-related second data segments are generated on the basis of at least one of the generic bit stream syntax description and the extensible stylesheet language transformation.
7 The method as claimed in claim 1, further comprising: determining, based upon a content-related marker, which content-related first data segments and content-related second data segments are to be output first in time.
8 The method as claimed in claim 7, wherein at least one of a scene change in an image sequence and a change of speaker in a speech sequence is identified by way of at least one of the first content-related markers and the second content-related markers.
9[pre] A device for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, comprising:
9[a][1] a synchronization device configured to output each of the content-related first data segments and the content-related second data segments, sequentially, according to their chronological sequence,
9[a][2] each of the content-related second data segments being output together with an associated one of the content-related first data segments based upon of an assignment rule for assigning each one of the content-related second data segments to each one of the content-related first data segment.
10 The method as claimed in claim 2, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments of the first data file and the content-related second data segments are synchronized on the basis of a combined assignment rule.
11 The method as claimed in claim 3, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments of the first data file and content-related second data segments are synchronized on the basis of a combined assignment rule.
12 The method as claimed in claim 5, wherein the media data include at least one of video and audio data.
13 The method as claimed in claim 1, wherein at least one of media data and metadata are represented by at least one of the first data file and the second data file.
14 The method as claimed in claim 13, wherein the metadata include metadata encoded according to the XML standard.

15 The method as claimed in claim 13, wherein the media data include at least one of video and audio data.
16 The method as claimed in claim 1, wherein the assignment rule is predefinable.
17 The device as claimed in claim 9, wherein the assignment rule is predefinable.
18 The method of claim 1, wherein the assignment rule is maintained in a file separate from the first data file and the second data file.
19 The device of claim 9, wherein the assignment rule is maintained in a file separate from the first data file and the second data file.
20 The method of claim 1, wherein the assignment rule is not based on a timestamp.
21 The device of claim 9, wherein the assignment rule is not based on a timestamp.
22[pre] A method for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, the method comprising:
22[a][1] determining an order for the content-related first data segments to be read out from the first data file and an order for the content-related second data segments to be read out from the second data file,
22[a][2] the content-related first data segments being ordered according to corresponding first content-related markers and the content-related second data segments being ordered according to corresponding second content-related markers;
22[b] assigning, according to an assignment rule, each of the ordered content-related second data segments that are read out from the second data file with a corresponding one of the ordered content-related first data segments that are read out from the first data file; and
22[c] sequentially outputting, by a device for synchronizing content-related data, in a chronological sequence, each of the assigned content-related first data segments and the assigned content-related second data segments.

D. Claim Construction (37 C.F.R. § 42.104(b)(3))

65. I have reviewed the entirety of the '794 patent and its prosecution history. For the purposes of my analysis, I have been asked to assume that no constructions are necessary, and that the ordinary and customary meanings of the

claim terms to a POSITA at the time of the alleged invention of the '794 patent apply.

66. I have been informed that there is a related district court proceeding in which the Patent Owner and other party proposed constructions as provided in the table below. *Starz Entm't, LLC v. VL Collective IP, LLC*, Amended Joint Claim Construction Chart, pages 4, 8-10 (Dkt. 80), Case No. 1:21-cv-01448 (D. Del.) ("*Starz*") (Ex. 1017). I understand the parties fully briefed the constructions of all disputed terms for the '794 patent. *Starz Entm't, LLC v. VL Collective IP, LLC*, Joint Claim Construction Brief filed by Plaintiffs, pages 1, 48-69 (Dkt. 78), Case No. 1:21-cv-01448 (D. Del.) (Ex. 1018).

Claim Term	Patent Owner's Proposed Construction	Opposing Party's Proposed Construction
"output" /"outputting" (Claims 1, 9)	"transmitted, stored together, or passed on to a processing unit" / "transmitting, storing together, or passing on to a processing unit"	Same as Patent Owner's Construction
Preambles (Claims 1, 9)	The preambles are construed to be limiting. In addition to the other limitations set forth in the preamble, the referenced first data file and second data file cannot be the same data file.	Same as Patent Owner's Construction
"assignment rule for assigning each one of	"assignment rule for assigning each one of the	Indefinite

<p>the content-related second data segments to one of the content-related first data segments” /</p> <p>“assignment rule for assigning each one of the content-related second data segments to each one of the content-related first data segment” /</p> <p>“assignment rule” /</p> <p>“the assignment rule is not based on a timestamp”</p> <p>(Collectively, the “assignment rule limitations”)</p> <p>(Claims 1, 3, 9, 16, 17, 20, 21)</p>	<p>content-related second data segments to one of the content-related first data segments, wherein the assignment is not performed by using exact timing information for the data segments (e.g., by using data segment timestamp information)”</p>	
<p>“content-related ... data segments”</p> <p>(Claims 1, 3, 7, 8, 9)</p>	<p>“data segments ordered in a file such that they will present the file’s contents in their intended presentation order when rendered sequentially”</p>	<p>“segments of content data that have a syntactical meaning within the respective data file”</p>

67. I have been informed that a claim construction order issued in the *Starz* Litigation on January 10, 2023, and the Court issued the below constructions for the

'794 patent. *Starz Entm't, LLC v. VL Collective IP, LLC*, Joint Claim Construction

Brief filed by Plaintiffs, (Dkt. 88), Case No. 1:21-cv-01448 (D. Del.) (Ex. 1019)

Claim Term	Court's Construction
<p>“output” / “outputting”</p> <p>(Claims 1, 9)</p>	<p>“transmitted, stored together, or passed on to a processing unit” / “transmitting, storing together, or passing on to a processing unit”</p>
<p>Preambles</p> <p>(Claims 1, 9)</p>	<p>The preambles are construed to be limiting. In addition to the other limitations set forth in the preamble, the referenced first data file and second data file cannot be the same data file.</p>
<p>“assignment rule for assigning each one of the content-related second data segments to one of the content-related first data segments” /</p> <p>“assignment rule for assigning each one of the content-related second data segments to each one of the content-related first data segment” /</p> <p>“assignment rule” /</p> <p>“the assignment rule is not based on a timestamp”</p> <p>(Collectively, the “assignment rule limitations”)</p>	<p>No construction necessary.</p>

(Claims 1, 3, 9, 16, 17, 20, 21)	
“content-related ... data segments” (Claims 1, 3, 7, 8, 9)	“segments of content data that have a syntactical meaning within the respective data file”

68. I also have been informed that Unified Patents, LLC (“Unified Patents”) filed an IPR on the ’794 patent that was instituted. Unified Patents did not offer specific construction for any claim term from the patent. *Unified Patents, LLC v. VL Collective IP, LLC*, IPR2022-01086, Paper 2 at 8-9 (PTAB Jun. 7, 2022). I understand that Patent Owner proposed constructions similar to the ones proposed in the district court proceeding, as provided in the table below. *Unified Patents, LLC v. VL Collective IP, LLC*, IPR2022-01086, Paper 7 at 16 (PTAB Jun. 7, 2022).

Claim Term	Patent Owner's Proposed Construction	Unified Patents' Proposed Construction
Preambles (Claims 1, 9 ¹⁰)	The preambles are construed to be limiting. The first data file and the second data file are distinct from each other.	Plain and ordinary meaning
“content-related ... data segments” (Claims 1, 9)	“data segments ordered in a file such that they will present the file’s contents in their intended presentation order when rendered sequentially”	Plain and ordinary meaning
“output” /“outputting” (Claims 1, 9) / (Claim 1)	“transmitted, stored together, or passed on to a processing unit” / “transmitting, storing together, or passing on to a processing unit”	Plain and ordinary meaning

69. In my opinion, the asserted prior art references teach all of the elements of the challenged claims under any of the above-discussed constructions, as set forth *infra* in Section X.

IX. PRIORITY DATE OF THE '794 PATENT

70. The '794 patent whose PCT application (PCT/EP2006/060808) was filed on March 16, 2006 claims priority to a German Patent Application, DE 10 2005

¹⁰ The proposed constructions apply for the identified claims and their dependent claims.

016 866, filed April 12, 2005. *See* '794 patent (Ex. 1001), field (22, 30, 86). Thus, the earliest possible priority date of any claim of the '794 patent is April 12, 2005.

71. For purposes of this Declaration only, I assume April 12, 2005 as the priority date for the challenged claims because it is unnecessary to determine specific priority dates of the challenged claims since the references relied upon for Grounds 1-4 in this Declaration predate the earliest possible priority date (April 12, 2005).

X. APPLICATION OF THE PRIOR ART TO CLAIMS 1-22

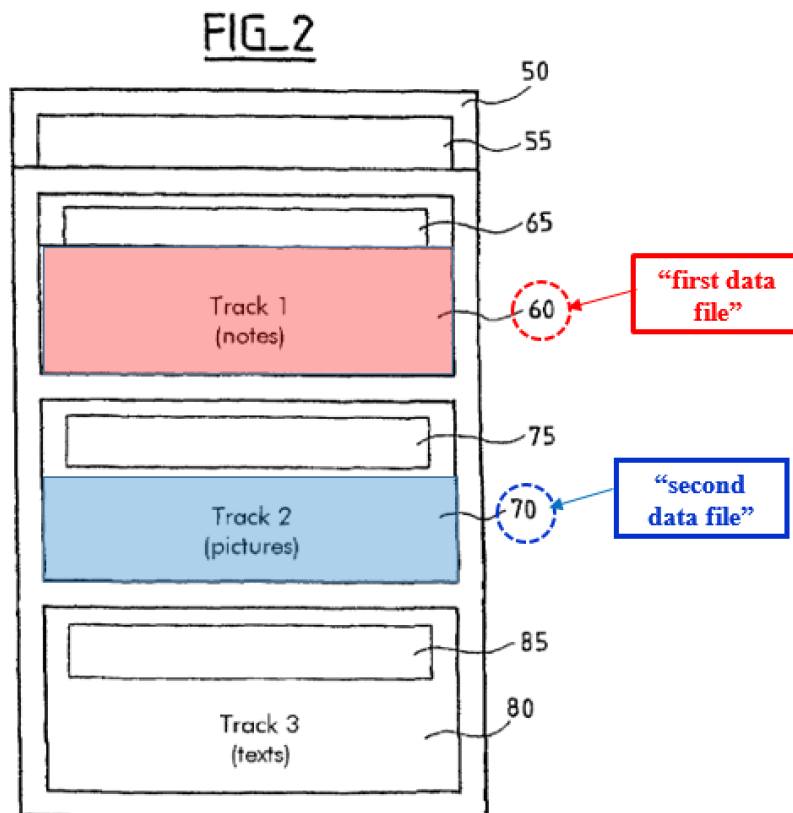
A. Ground 1: *Comps* Anticipates Claims 1-5, 9-13, 15-17, and 20-21 of the '794 Patent

72. *Comps* discloses each element of claim 1, as explained below.

1. Claim 1

- a. 1[pre]: “A method for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, the method comprising:”

73. *Comps* discloses claim 1's preamble. For example, *Comps* discloses “a method of synchronizing data in a multimedia document (50) comprising at least two separate data files (track1, track2) referred to as the first file, the second file, etc.” as shown below in Figure 2. *Comps*, 2:6-9. *Comps* further discloses “data of a first type, of a second type, etc. is stored in the first file, in the second file, etc., respectively[.]” *Id.*, 2:10-14. Accordingly, Track 1 60 in red is a “first data file” and Track 2 70 in blue is a “second data file.”



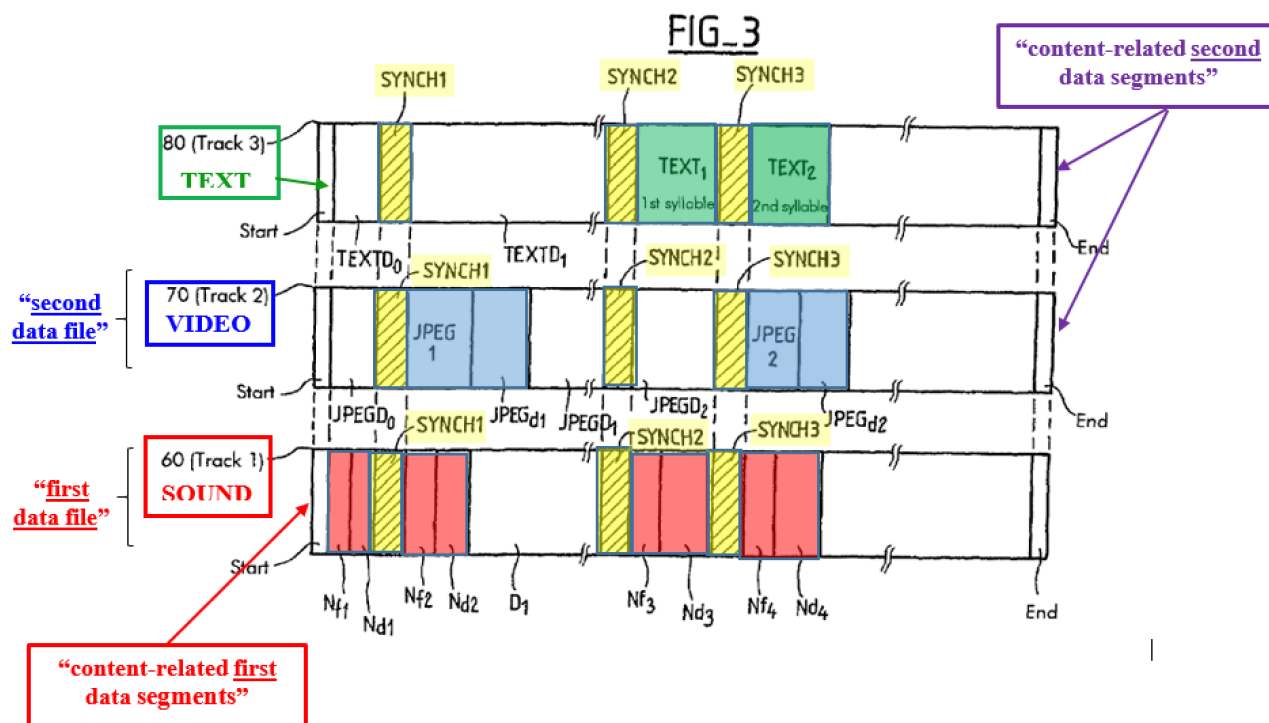
Comps, Fig. 3 (annotated).

74. As illustrated in Figure 3 below, each track can include different types of content. For example, Track 1 60 can be sound data, such as a sequence of musical notes or a Musical Instrument Digital Interface (“MIDI”) format track for sound, and Track 2 70 can be a sequence of pictures or video data. Track 3 80 can include sequences of texts. *Id.*, 2:54-64 (“Each track of the file 50 can contain a single type of data. For example, track 60 can be a MIDI (Musical Instrument Digital Interface) format track for sound, track 70 can contain a sequence of pictures, and track 80 can contain sequences of texts”), 3:8-19, 3:48-49 (“musical notes included in the track 1 data must not be interrupted.”), Fig. 2.

75. The content in each track is fragmented into data segments. For example, the sound content in Track 1 60 is fragmented into segments represented by fields Nf1:Nd1, Nf2:Nd2, Nf3:Nd3, Nf4:Nd4 annotated in red below. In Track 2 70, the picture content is fragmented into segments represented by fields JPEG1:JPEGd1, JPEG2:JPEGd2 as annotated in blue below. Finally, in Track 3 80, the sequence of texts is fragmented into segments represented by fields TEXT1, TEXT2 annotated in green below. *Id.*, 3:22-41.

76. As shown in Figure 3 of *Comps*, the sound segments in Track 1 60 and picture segments in Track 2 70 are “content-related first data segments” and “content-related second data segments” of claim 1, respectively. The sound segments in Track 1 60 and the text segments in Track 3 80 shown in Figure 3 of *Comps* could also correspond to “content-related first data segments” and “content-related second data segments” of claim 1, respectively.

77. It is my opinion that the sound segments in Track 1 60 shown in *Comps*, Figure 3 could also correspond to “content-related first data segments” of claim 1; and the pictures and text segments in Track 2 70, and Track 3 80 shown *Comps*, Figure 3 could collectively correspond to “content-related second data segments” of claim 1.



Comps, Fig. 3 (annotated).

78. As Figure 3 above shows, the respective data segments are synchronized based on the SYNCH command present in each track such that the file's contents will be presented or outputted in their intended presentation order from start to end. Moreover, the sound segments of the "first data file" and the picture segments of the "second data file" are different.

- b. 1[a][1]: "sequentially outputting, by a device for synchronizing content-related data, the content-related first data segments and the content-related second data segments according to their chronological sequence in such a way that"

79. *Comps* discloses this element. Regarding “outputting,” the ’794 patent’s specification states that “the term ‘output’ is understood to mean that first and second data segments are, for example, transmitted, stored together or passed on to a processing unit.”

80. As I explained in paragraph 66, the sound segments in Track 1 60 shown in *Comps*, Figure 3 are “content-related first data segments” of claim 1. The pictures and text segments in Tracks 70, 80 collectively correspond to “content-related second data segments” of claim 1.

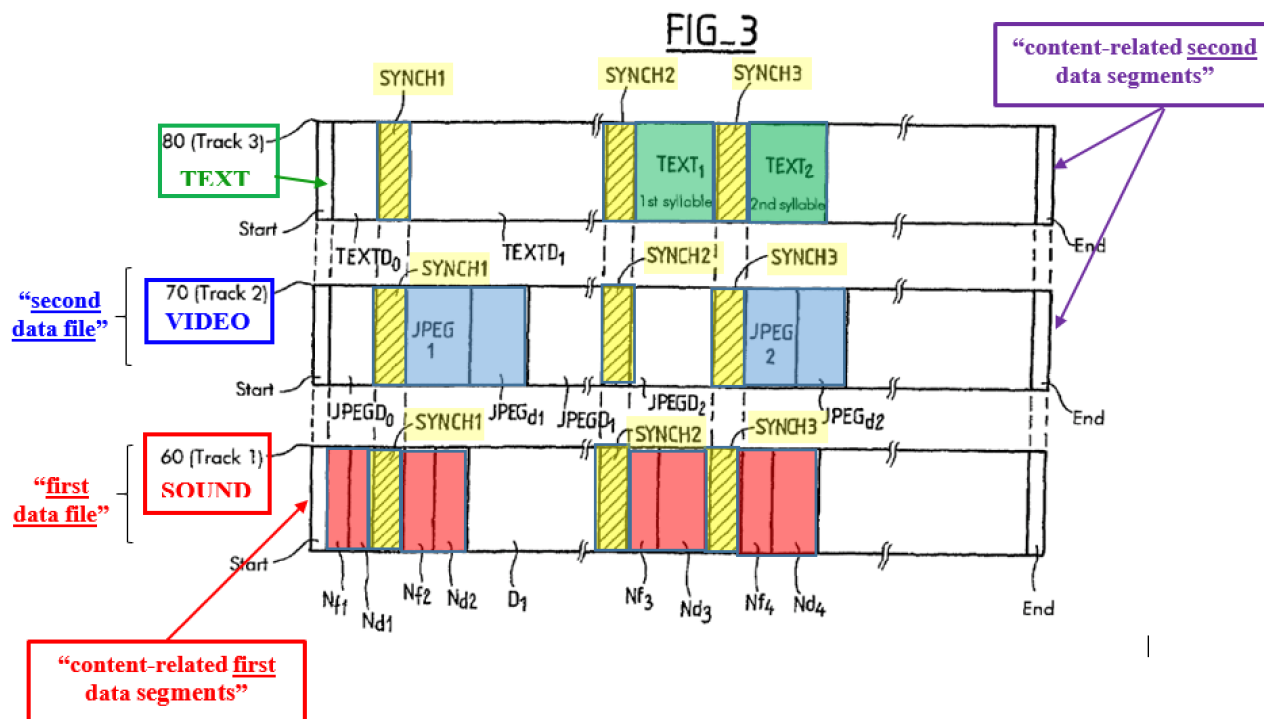
81. *Comps* discloses “a device for synchronizing data in a multimedia file containing at least one track in which [the] data is *stored* and at least one synchronization command in each track[.]” *Comps*, 2:24-37. The microprocessors, which are a “device” “sequentially” execute synchronization commands (*i.e.*, SYNCH1, SYNCH2, SYNCH3) to play or “output” sound segments in Track 1 60 and to display, or “output” picture segments and/or text segments stored (*i.e.*, “outputted”) in Tracks 70 and 80 altogether from start to end as illustrate in Figure 3 below, which is “according to their chronological sequence.” *Id.*, 3:5-7 (“Each track also contains data organized in the form of commands which are executed sequentially by the microprocessor (for example to display a picture or a text).”; *see also id.*, 3:20-21 (“Each track has a Start field for starting presentation to the user and an End field for ending presentation to the user[.]” 3:48-51 (“In the FIG.

3 example, musical notes included in the track 1 data must not be interrupted. Synchronizing the three tracks must not entail interrupting the music heard by the user.”), 4:3-17.

82. *Comps* “sequentially output[s] ... the content-related first data segments and the content-related second data segments according to their chronological sequence.” Figure 1 of the ’794 patent illustrates R1 and R2, which are the “chronological sequence” of how the data segments are to be output, are synchronized together so that content from R1 and R2 are sequentially output at the same time, and are thus output together. ’794*Pat*, 4:46-5:4, Fig. 1. Figure 3 of *Comps* similarly discloses Track 1 60 containing sound, Track 2 70 containing a sequence of pictures, and Track 3 80 containing a sequence of texts are “sequentially outputted” together as they are “scanned by microprocessors and presented simultaneously to a user.” *Comps*, 2:60-67 (“The different tracks may be intended to be scanned by microprocessors and presented simultaneously to a user. The different microprocessors therefore scan the tracks at the same time”); Fig. 3.

83. In the example in Figure 3 below, *Comps* further explains that the second note, represented as NF₂ and Nd₂, must be heard at the moment the first picture is displayed, corresponding to the field JPEG1 of track 2. After waiting for a period of time, indicated by field D1, the third note as indicated by Nf3 and Nd3,

“must be heard at the moment that the first syllable is displayed, corresponding to the field TEXT1 of track 1.” *Id.* 3:57-4:2; *see also* 3:48-51.



Comps, Fig. 3 (annotated).

84. Therefore, it is my opinion that *Comps* discloses this feature.

- c. **1[a][2]: “each of the content-related second data segments is output together with an associated one of the content-related first data segments on the basis of an assignment rule for assigning each one of the content-related second data segments to one of the content-related first data segments.”**

85. It is my opinion that *Comps* discloses this element. As discussed above in section X.A.1.b, “each of the content-related second data segments is output together with an associated one of the content-related first data segments.”

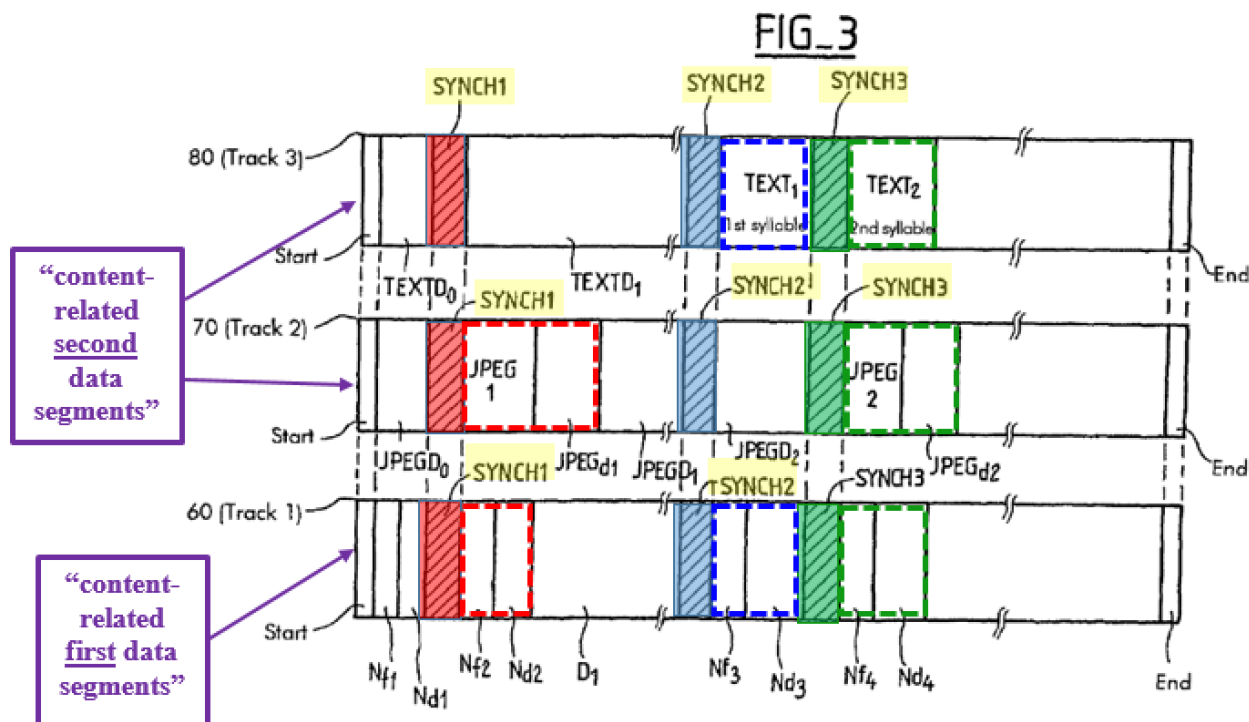
Microprocessors sequentially execute synchronization commands (the synchronization commands are shown as fields SYNCH1, SYNCH2, SYNCH3) to “output together” the picture segments and/or text segments stored in Track 2 70 and Track 3 80, which are the “content-related second data segments” with the “associated” sound segments in Track 1 60, *i.e.*, the “content-related first data segments”).

86. *Comps* discloses that the “different tracks may be intended to be scanned by microprocessors and presented simultaneously to a user. The different microprocessors, therefore, scan the tracks at the same time.” *Id.*, 2:64-67. *Comps* further discloses “[e]ach track also contains data organized in the form of [synchronization] commands which are executed sequentially by the microprocessor” to synchronously perform sound segments, display picture segments, and/or text segments stored in the tracks 60, 70, 80. *Id.*, 3:5-7 (emphasis added); *see also id.*, 3:20-21, 3:48-51, 4:3-17. As I illustrate below with grey, dotted boxes, the synchronization commands are fields SYNCH1, SYNCH1, SYNCH3. *Id.*, 4:3-17

87. Accordingly, the “assignment rule” which “assign[s] each one of the content-related second data segments to one of the content-related first data segments” is the synchronization commands that are executed by the microprocessors. Like the “assignment rule” described in the ’794 patent at column

6, lines 4-27, for example, which discloses content from data segment s2 is synchronized with content from data segment s1, tracks 60, 70, and 80 in *Comps* have inserted SYNCH commands that control which content from each track is synchronized with content from the other tracks, so that the end result is that the different content is “output together” or transmitted together simultaneously, as I explained in section X.A.1.b, and can be seen by the tracks vertical alignment illustrated below in Figure 3.

88. The execution of the synchronization commands by the microprocessors results in each of the picture segments such as JPEG₁, JPEG₂ to be “output together with an associated one of” the sound segments, Nf₂:Nd₂, Nf₄:Nd₄. For example, JPEG₁ is “output together with” the associated sound segment Nf₂:Nd₂, and JPEG₂ is “output together with” the associated sound segment Nf₄:Nd₄. *Id.*, Fig. 3, 4:5-17. As another example, based on the execution of the synchronization commands by the microprocessors, each of the text segments TEXT₁, TEXT₂ is “output together with an associated one of” the sound segments Nf₃:Nd₃, Nf₄:Nd₄. Thus, TEXT₁ is “output together with” the associated sound segment Nf₃:Nd₃, and TEXT₂ is “output together with” the associated sound segment Nf₄:Nd₄. *Id. Comps* in my opinion discloses this element.



Comps, Fig. 3 (annotated).

89. *Comps* discloses “[t]he synchronization commands are not temporal commands, as in the prior art, but are instead dependent on a specific event.” *Id.*, 3:44-47 (emphasis added); *see also id.*, 2:20-23 (“The method is advantageously characterized in that the important event corresponds to a command to display a text, a command to display a picture, or a command to reproduce a sound.”) (emphasis added). More specifically, *Comps* synchronization commands work by having a “master” track (such as the audio in Track 1) whose microprocessor sends a synchronization command to the microprocessors that control the other “slave” tracks. (“Synchronizing the three tracks must not entail interrupting the music heard by the user. The microprocessor dedicated to reading this track is considered to be

the master.”) *Id.* 3:49-52; *see also id.*, 4:22-28. *Comps* uses synchronization commands in this manner, as described, to avoid the use of timestamps. Accordingly, the synchronization commands in *Comps* being executed by the microprocessors, which is the “assignment rule” would not be performed by using exact timing information for the data segments for example, by using data segment timestamp information. It is thus my opinion that *Comps* satisfies element 1[a][2].

90. Additionally, the scope of “assignment rule” in claim 1 is broad such that the “assignment rule” does not need to be maintained in a file separate from the first data file and the second data file. Dependent Claims 18, 19 in *Comps* narrow the scope of “assignment rule” recited in independent claims 1 and 9 by requiring the “assignment rule” be “maintained in a file separate from the first data file and the second data file.” I have been informed that the doctrine of claim differentiation presumes that dependent claims are “of narrower scope than the independent claims from which they depend.” Thus, it is my opinion that the synchronization commands in Tracks 60, 70, and 80 that are executed by the microprocessors satisfy “the assignment rule” as recited in claim 1.

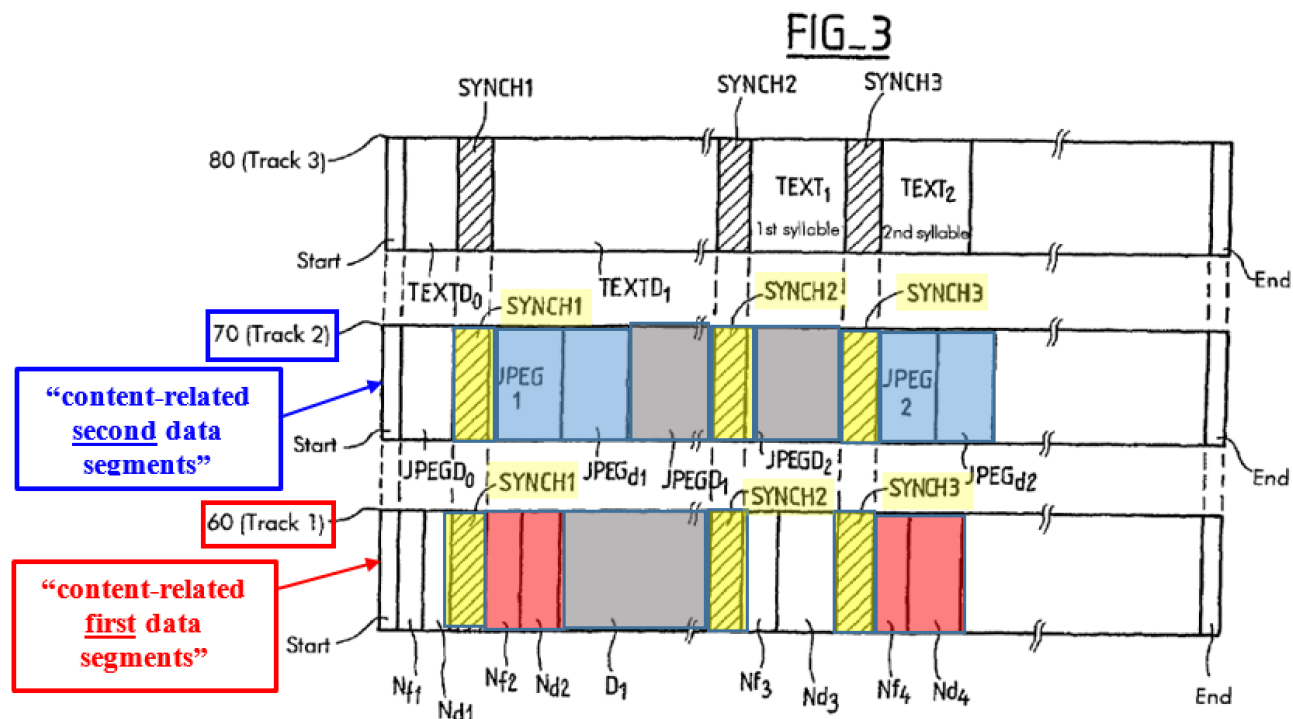
2. Claim 2

- a. **“The method as claimed in claim 1, wherein the assignment rule is formed in such a way that each of the content-related second data segments are assigned to precisely one of the content-related first data segments and a further assignment is carried out after a number**

of sequentially succeeding content-related first data segments.”

91. It is my opinion that *Comps* discloses this element. As I show in Figure 3 below, the synchronization commands executed by the microprocessors, which is “the assignment rule” are “formed in such a way that each” of the picture segments (*i.e.* “content-related second data segments”) “are assigned to precisely one” of the sound segments (*i.e.* “content-related first data segments”). For example, JPEG₁:JPEG_{d1} one of the picture segments shown below, the blue rectangle on the left is assigned precisely to Nf₂:Nd₂, one of the sound segments, which is in the red rectangle on the left. *See Comps*, 3:58-64. D1, JPEGD1 and JPEGD2 (in grey) represent waiting times before presenting subsequent notes and images. *Id.*, 3:25-26, 36-37. Similarly, JPEG₂:JPEG_{d2} is “assigned [] precisely” to Nf₄:Nd₄. *Id.*, Fig. 3, 3:65-4:2.

92. *Comps* also has “a further assignment [] carried out after a number of sequentially succeeding content-related first data segments.” For example, the assignment between JPEG₂:JPEG_{d2} and Nf₄:Nd₄ is performed after one “sequentially succeeding content-related first data segments” *i.e.*, Nf₃:Nd₃. *Comps*, in my opinion, discloses claim 2.



Comps, Fig. 3 (annotated).

3. Claim 3

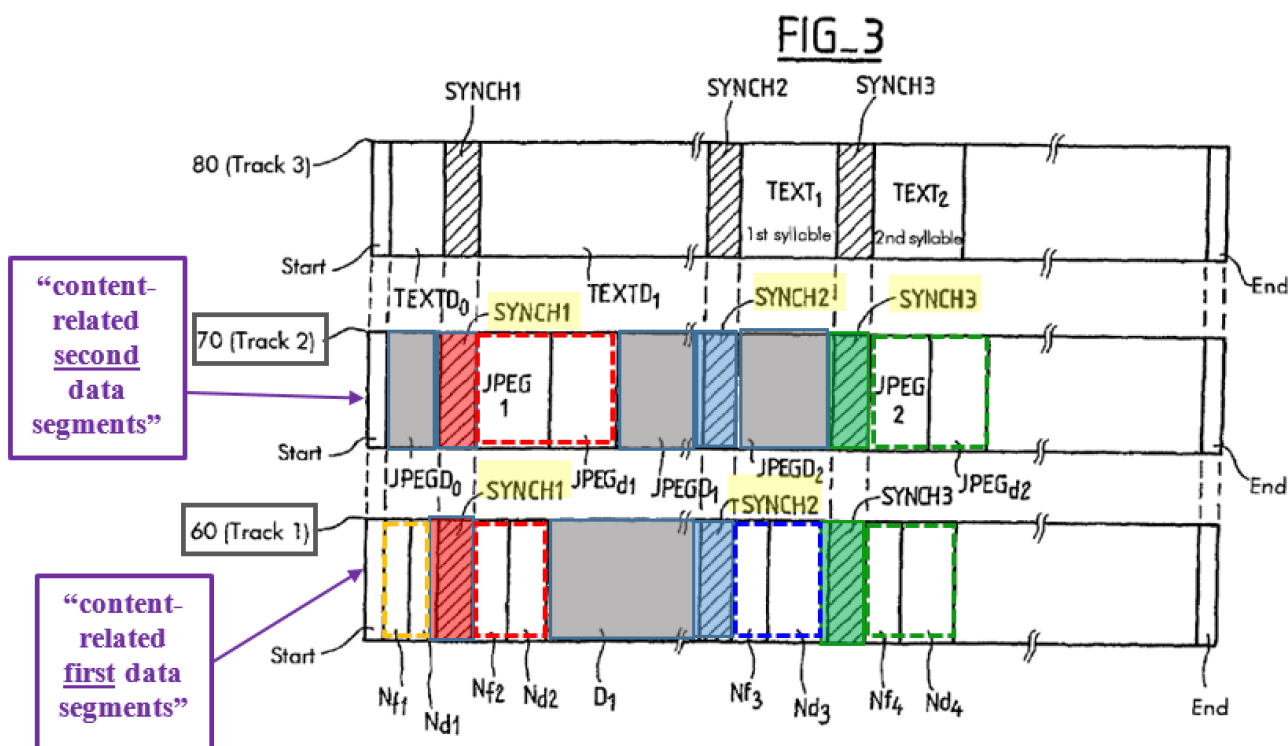
- a. "The method as claimed in claim 1, wherein the assignment rule is formed in such a way that, for each of the content-related second data segments, the content-related first data segments are assigned by specification of a number, the number representing a position of the content-related first data segments within the order of content-related first data segments."

93. *Comps* discloses this element. *Comps* discloses that "each synchronization command of each event to be executed simultaneously is *numbered identically from one file to another.*" *Comps*, Claim 6 (emphasis added). In Track 2 70, in Figure 3 below, two JPEG images are represented by the fields JPEG1 and

JPEG2. JPEG_{d1} and JPEG_{d2} represents the duration of how long the image must be presented to the user, respectively. *Id.*, 3:31-35. JPEGD₀, JPEGD₁, and JPEGD₂ “represent waiting times before or between images.” *Id.*, 36-37. Track 1 60, Nf₁ “represents the frequency of a first note, and Nd₁ represents its duration.” *Id.*, 3:22-24. Fields Nf₂ and Nd₂ indicate a second note, and D₁ “represents a waiting time before presenting the subsequent notes of the track.” *Id.*, 3:24-26; 4:3:18.

94. By using the numbered commands, *i.e.*, SYNCH1, SYNCH3, each of the picture segments JPEG₁:JPEG_{d1}, JPEG₂:JPEG_{d2} is assigned to an associated one of the sound segments, *i.e.*, Nf₂:Nd₂, Nf₄:Nd₄, respectively. *Id.*, Fig. 3. 3:57-4:18. The number (*i.e.*, 1, 2, 3) of the synchronization commands, *i.e.*, SYNCH1, SYNCH2, SYNCH3 that are executed by the microprocessors (*i.e.*, “the assignment rule”, “represent[s] the position of” the sound segments (*i.e.*, “content-related first data segments”) “within the order of” the sound segments, Nf₁:Nd₁, Nf₂:Nd₂, Nf₃:Nd₃, Nf₄:Nd₄. *Id.*, 3:57-4:2; Fig. 3.

95. SYNCH1 is positioned between the first and second sound segments Nf₁:Nd₁ and Nf₂:Nd₂; SYNCH2 is positioned between the second and third sound segments Nf₂:Nd₂ and Nf₃:Nd₃; and SYNCH3 is positioned between the third and fourth sound segments Nf₃:Nd₃ and Nf₄:Nd₄. *Id.*, 4:3-18; Fig. 3. This sequence is illustrated in Figure 3 of *Comps* below. Thus, it is my opinion that *Comps* discloses claim 3.



Comps, Fig. 3 (annotated).

4. Claim 4

- a. “The method as claimed in claim 1, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments and the content-related second data segments are synchronized on the basis of a combined assignment rule.”

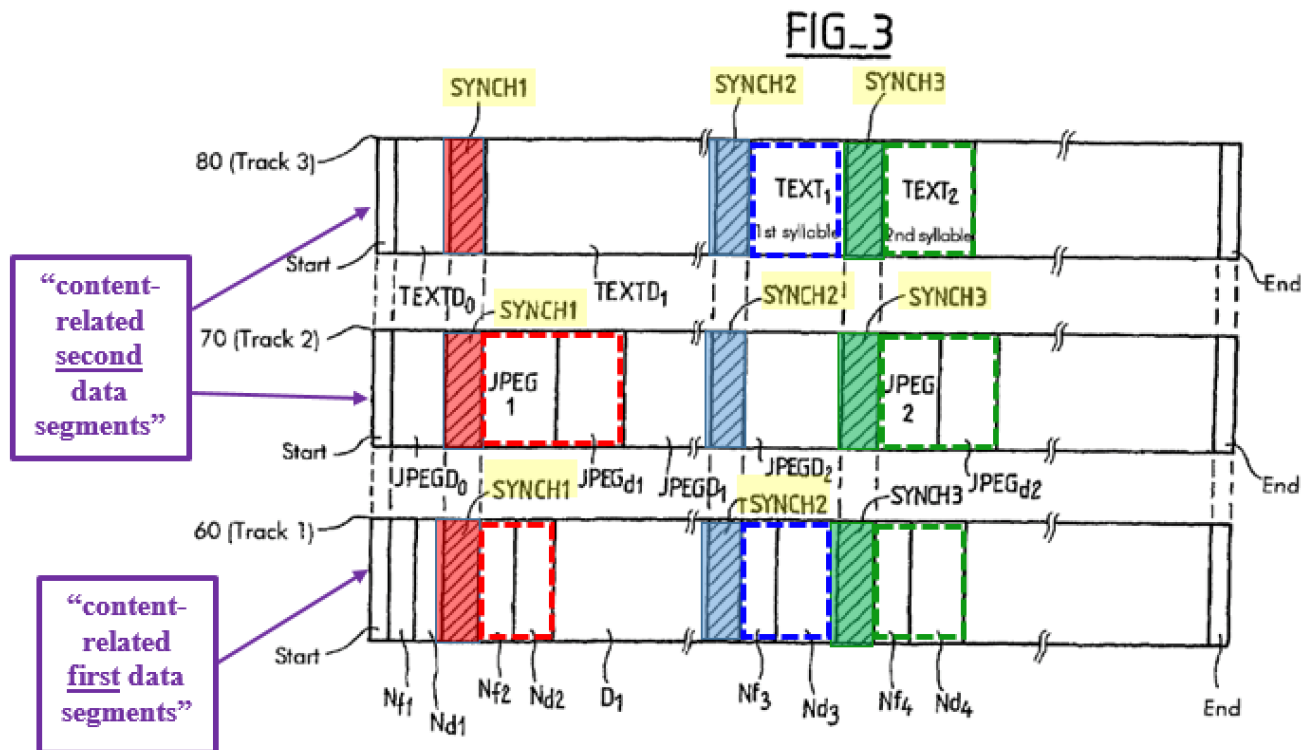
96. *Comps* discloses this element. The '794 patent explains “more than one single second data file D2 can be synchronized with the first data file D1. For example, there are two second data files D21, D22. For this purpose a combined predefinable assignment rule ZK is formed.” '794PAT, 7:5-9.

97. As I previously explained in section X.A.1.a, the sound segments in the track 60 are the “content-related first data segments” of claim 1, and the pictures and text segments in tracks 70, 80 would collectively be the “content-related second data segments” of claim 1. *Comps* also discloses “at least two separate data files (track1, track2) [are] referred to as the first file, the second file, etc.” *Id.*, 2:6-9. *Comps* discloses “data of a first type, of a second type, etc. is stored in the first file, in the second file, etc., respectively[,]” so each track, including track 70 and 80 are stored in at least two separate data files, which is a “plurality of second data files.” *Id.*, 2:10-14.

98. *Comps* discloses that the sound segments in track 60, which are “the content-related first data segments” and the pictures and text segments in the tracks 70, 80, which are “the content-related second data segments” are synchronized on the basis of “a combined assignment rule,” as depicted in Figure 3 below. As I previously explained in section X.A.1.c, the combined synchronization commands shown as SYNCH1, SYNCH2, SYNCH3 in Figure 3 that are executed sequentially by microprocessors are a “combined assignment rule,” so that the “musical notes included in the track 1 data must not be interrupted. Synchronizing the three tracks must not entail interrupting the music heard by the user.” *Id.*, 3:49-52. Accordingly, the master microprocessor for the track 1 sound data “forces the other microprocessors in charge of the other tracks, referred to as slaves, to synchronize

with it.” *Id.*, 3:53-54. In other words, one track is higher priority (e.g., track 1 containing audio). That track is controlled by the “master” microprocessor, which sends synchronization commands to the other “slave” microprocessors. They, then can wait or skip data in order to ensure that the track data they control (such as images or text) are being output at the same time as the corresponding data from the higher priority master track. *Id.* 4:18-28.

99. As an example, “the fourth note, corresponding to the fourth fields NF₄ and ND₄, must be heard at the moment at which are simultaneously displayed the second picture, corresponding to the field JPEG1 in track 2, and the second syllable, corresponding to the field TEXT2”). For these reasons, it is my opinion that *Comps* discloses claim 4.



Comps, Fig. 3 (annotated).

5. Claim 5

- a. **“The method as claimed in claim 1, wherein media data are represented by at least one of the first data file and the second data file.”**

100. *Comps* discloses this element. The '794 patent discloses “media data [are] video or audio data, and/or metadata[.]” ’794Pat, 3:4-8. *Comps* discloses “Track 60 or track 1 contains only *MIDI sound data* . . . Track 70 or track 2 contains only data corresponding to sequences of *JPEG images*. This data could equally be *video data* . . . Track 80 or track 3 contains only data corresponding to *text messages*.” *Comps*, 3:10-19 (emphasis added). *Comp* discloses “at least two separate data files (track1, track2) [are] referred to as the first file, the second file,

etc.” *Id.*, 2:6-9. *Comps* also discloses “data of a first type, of a second type, etc. is stored in the first file, in the second file, etc., respectively[.]” *Id.*, 2:10-14. Thus, *Comps* discloses sound data and the picture (or video) data, which are “media data,” are stored in “first data file” and the “second data file,” respectively. It is my opinion, therefore, that *Comps* discloses claim 5.

6. Claim 9

a. 9[pre]:“A device for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, comprising:”

101. This element is substantively identical to that in claim 1[pre].

102. Therefore, it is my opinion that *Comps* discloses this element for the same reasons I discussed in section X.A.1.a. With respect to “[a] device,” *Comps* teaches “a device for synchronizing data in a multimedia file containing at least one track in which [the] data is stored and at least one synchronization command in each track[.]” *Comps*, 2:24-31 (emphasis added). The microprocessor disclosed in *Comps* is a “device for synchronizing” because it “sequentially” executes synchronization commands SYNCH1, SYNCH2, SYNCH3 to play the sound segments and to display the picture segments and/or text segments stored in tracks 60, 70, 80, respectively, as illustrated in Figure 3. *Id.*, 3:5-7 (“Each track also contains data organized in the form of commands which are executed sequentially by the microprocessor (for example to display a picture or a text).”; *see also id.*,

2:20-24 (“The method is advantageously characterized in that the important event corresponds to a command to display a text, a command to display a picture, or a command to reproduce a sound.”); 3:48-51, 4:3-17.

- b. 9[a][1]: “a synchronization device configured to output each of the content-related first data segments and the content-related second data segments, sequentially, according to their chronological sequence,”

103. This element is substantively identical to that in claim 1[a][1].

104. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.1.b.

- c. 9[a][2] “each of the content-related second data segments being output together with an associated one of the content-related first data segments based upon of an assignment rule for assigning each one of the content-related second data segments to each one of the content-related first data segment.”

105. This element is substantively identical to that in claim 1[a][2].

106. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.1.c.

7. Claim 10

- a. “The method as claimed in claim 2, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments of the first data file and the content-related second data segments are synchronized on the basis of a combined assignment rule.”

107. This element is substantively identical to that in claims 2 and 4.

108. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in sections X.A.2 and X.A.4.

8. Claim 11

- a. **“The method as claimed in claim 3, wherein the second content-related data segments belong to a plurality of second data files, and the content-related first data segments of the first data file and content-related second data segments are synchronized on the basis of a combined assignment rule.”**

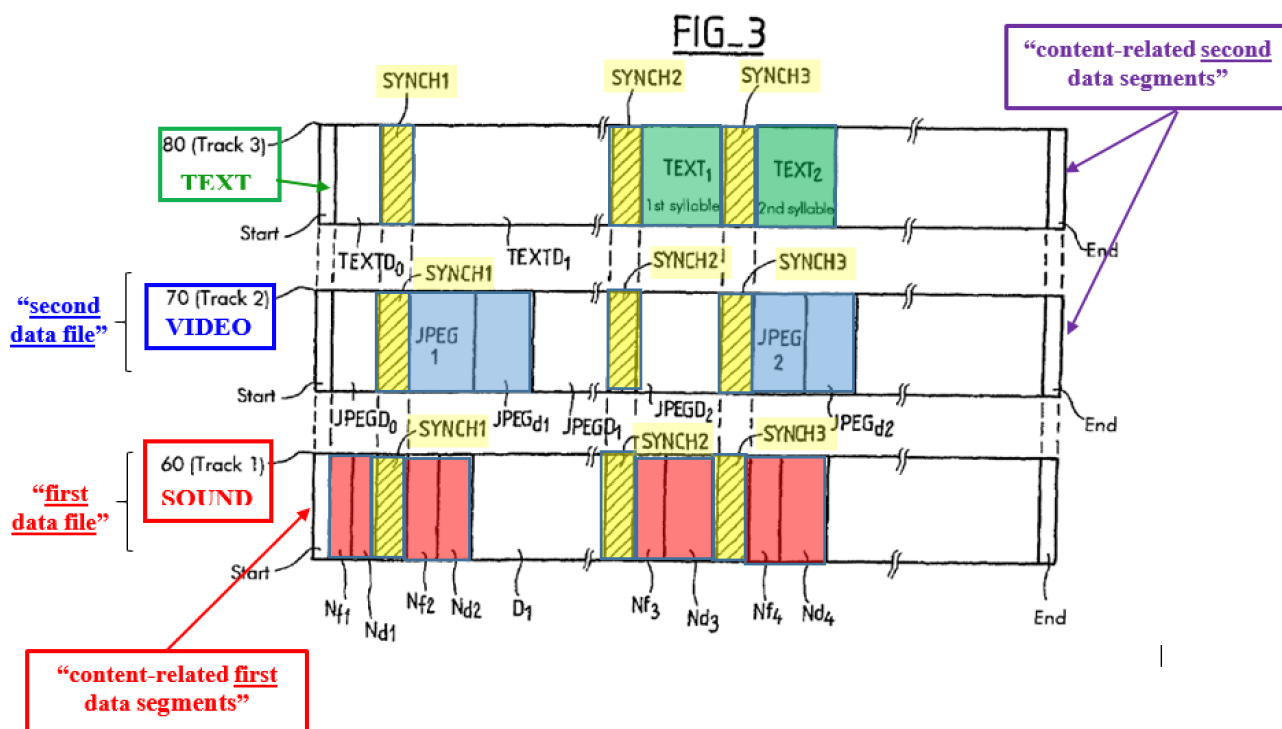
109. This element is substantively identical to that in claims 3 and 4.

110. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in sections X.A.3 and X.A.4.

9. Claim 12

- a. **“The method as claimed in claim 5, wherein the media data include at least one of video and audio data.”**

111. It is my opinion that *Comps* discloses this element. For example, as illustrated in Figure 3 below, *Comps* discloses, “Track 60 or track 1 contains only *MIDI sound data* . . . Track 70 or track 2 contains only data corresponding to sequences of *JPEG images*. This data could equally be *video data*.” *Comps*, 3:10-17 (emphasis added); *see also id.*, 3:48-49 (“In the FIG. 3 example, *musical notes* included in the track 1 data must not be interrupted.”) (emphasis added); Fig. 3. Accordingly, the sound data in track 60 is “audio data,” and the sequences of JPEG images in track 70 is “video” data.



Comps, Fig. 3 (annotated).

10. Claim 13

- a. "The method as claimed in claim 1, wherein at least one of media data and metadata are represented by at least one of the first data file and the second data file."

112. This element is substantively identical to that in claim 5.

113. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.5.

11. Claim 15

- a. "The method as claimed in claim 13, wherein the media data include at least one of video and audio data."

114. This element is substantively identical to that in claims 12 and 13.

115. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in sections X.A.9 and X.A.10.

12. Claim 16

a. **“The method as claimed in claim 1, wherein the assignment rule is predefinable.”**

116. *Comps* discloses this element.

117. The '794 patent discloses:

In an alternative embodiment, the *predefinable assignment rule* is formed in such a way that for each second data segment a first data segment is assigned by specification of a number, the number representing a position of the first data segment within the chronological sequence of the first data segments. By this, the assignment of first and second data segments is achieved in a particularly flexible manner. In this embodiment a plurality of second data segments can be assigned to a first data segment.

'794Pat, 2:55-63 (emphasis added); *see also id.*, 2:47-54; 5:35-6:42.

118. As I explain in section X.A.1.c, synchronization commands SYNCH1, SYNCH1, SYNCH3 executed sequentially by the microprocessors of *Comps* are the “assignment rule” in claim 1. *Comps*, 4:3-17; Fig. 3. *Comps* discloses “*at least one synchronization command is inserted in each file, characterized in that said synchronization command is inserted before each event-related command characterizing an important event.*” *Id.*, Abstract (emphasis added); *see also id.*, 3:5-7 (“*Each track also contains data organized in the form of [synchronization]*”).

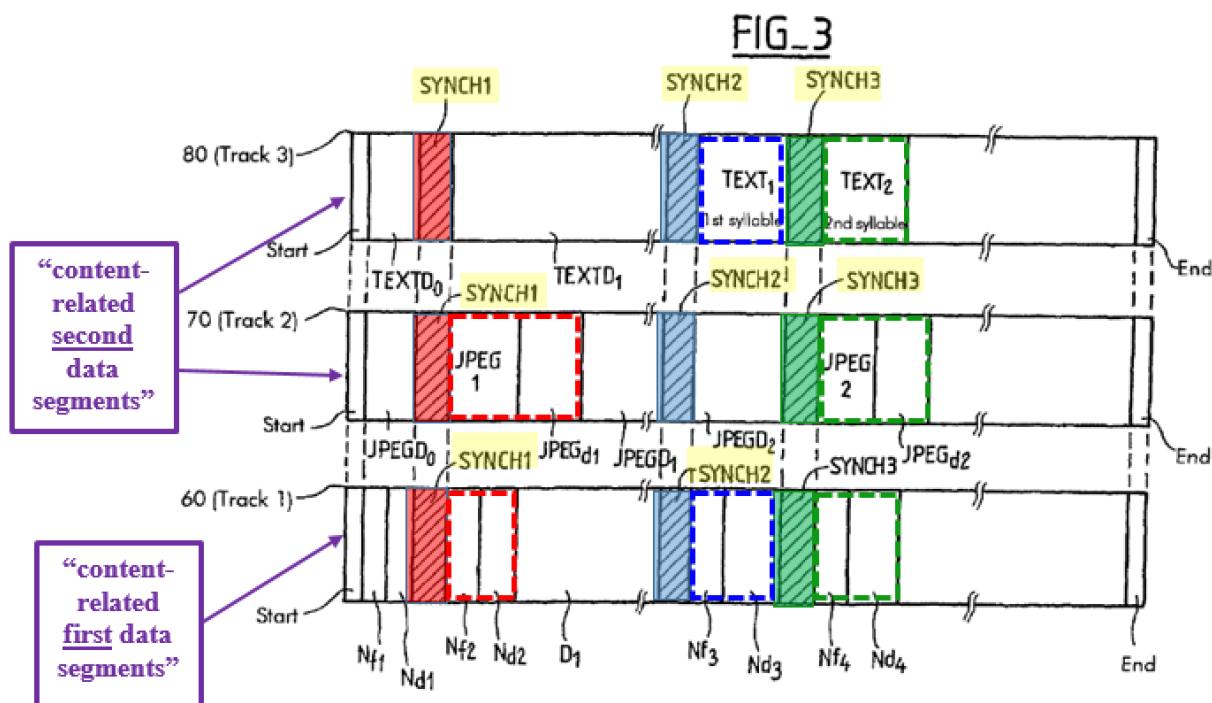
commands which are executed sequentially by the microprocessor (for example to display a picture or a text.”).

119. Therefore, it is my opinion, that the synchronization commands SYNCH1, SYNCH2, SYNCH3 executed by the microprocessors are an “assignment rule,” which is “predefinable.” To further elaborate, the synchronization commands are stored in each track 60, 70, 80. Once the microprocessor sequentially executes the predefined synchronization commands, the sound segments in track 60 are played and the picture segments and/or text segments stored in the tracks 70, 80 are displayed as depicted in Figure 3 below. *Id.*, 3:5-7; 3:48-51, 4:3-17.

120. As I described previously in section X.A.1.c, the synchronization commands are sent from the master microprocessor to slave microprocessors. *Comps* explains that “at the moment it receives data, the slave microprocessor dedicated to track I, which is lagging behind, has not yet encountered the field SYNCH1 in track I, it continues to scan its file without executing the commands encountered, in order to reach the field SYNCH1 as quickly as possible.” *Id.* 4:30-35. That shows that the synchronization commands were already in the track data so that the slave processor could scan ahead to locate it. Thus, the synchronization commands were predefined.

121. That is, as a result of the synchronization commands, for each second data segment, a first data segment is assigned by specification of a number,

information representing a position of the first data segment within the chronological sequence of the first data segments. For example, in Figure 3, after SYNCH 1 is executed, second data segment JPEG1 in track 70 which is assigned to the second note in audio track 60 $Nf_2:Nd_2$ (which is the first data segment) are displayed and played simultaneously, and JPEG2 in track 70, which is assigned to the fourth note in audio track 60, $Nf_4:Nd_4$ are displayed and played simultaneously as a result of the SYNCH commands. Accordingly, it is my opinion that the synchronization commands that are in each of tracks 60, 70, and 80 are an “assignment rule” that is “predefinable” because the synchronization commands are predefined, and because they assign the second data segments to the first data segments by specification of information that represents the position of the first data segment within the chronological sequence of the first data segments.



Comps, Fig. 3 (annotated).

13. Claim 17

- a. **“The device as claimed in claim 9, wherein the assignment rule is predefinable.”**

122. This element is substantively identical to that in claim 16.

123. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.13.

14. Claim 20

- a. **“The method of claim 1, wherein the assignment rule is not based on a timestamp.”**

124. *Comps* discloses this element. As I explained in section X.A.1.c, the synchronization commands SYNCH1, SYNCH1, SYNCH3, executed by the

microprocessors shown in *Comps*, Figure 3 are the claimed “assignment rule” of claim 1. *Comps*, 4:3-17. *Comps* discloses “[t]he synchronization commands are *not temporal commands*, as in the prior art, but are instead *dependent on a specific event*. Thus the fields SYNCH_i are not present in the tracks at regular time intervals.” *Id.* 3:46-47 (emphasis added); *see also id.*, 2:20-23 (“The method is advantageously characterized in that *the important event corresponds to a command* to display a text, a command to display a picture, or a command to reproduce a sound.”) (emphasis added).

125. More specifically, *Comps* synchronization commands work by having a “master” track (such as the audio in Track 1) whose microprocessor sends a synchronization command to the microprocessors that control the other “slave” tracks. (“Synchronizing the three tracks must not entail interrupting the music heard by the user. The microprocessor dedicated to reading this track is considered to be the master.”) *Id.* 3:49-52; *see also id.*, 4:22-28. *Comps* uses synchronization commands in this manner, as described, to avoid the use of timestamps.

126. Therefore, it is my opinion that the synchronization commands SYNCH1, SYNCH2, SYNCH3 that are executed by the microprocessors disclosed in *Comps* are “not based on a timestamp.”

15. Claim 21

- a. **“The device of claim 9, wherein the assignment rule is not based on a timestamp.”**

127. This element is substantively identical to that in claim 20.

128. Accordingly, *Comps* discloses this element for the same reasons discussed in section X.A.15.

B. Ground 2: *Comps* and *Wan* Render Obvious Claims 6-8, and 22

129. It is my opinion that *Comps* in combination with *Wan* renders obvious claims 6-8, and 22. Ground 2 is not duplicative of Ground 1, 3, and 4 because it is the only Ground based on *Comps* and *Wan*.

1. Claim 6¹¹

- a. **“The method as claimed in claim 1, wherein at least one of the content-related first data segments and content-related second data segments are generated on the basis of at least one of the generic bit stream syntax description and the extensible stylesheet language transformation.”**

130. It is my opinion that *Comps* in view of *Wan* disclose this element. As discussed in section X.A.1, which is incorporated by reference herein, *Comps* discloses all of the elements in claim 6 except that it does not expressly disclose that “the content-related first data segments” or “the content-related second data

¹¹ To the extent that this claim is not indefinite due to lack of antecedent basis of “the generic bit stream syntax description and the extensible stylesheet language transformation,” *Comps* in combination with *Wan* disclose this element.

segments” are “generated on the basis of at least one of the generic bit stream syntax description and the extensible stylesheet language transformation.”

131. But it was well known to a POSITA at the time of the '794 patent that content-related data segments are generated using an extensible stylesheet language transformation. For example, *Wan* describes that “the composer application 54 uses *an XSLT-based presentation template 60* to extract the required description fragments from a movie description 62 to generate a SMIL-like presentation description 64 (or presentation script). The <par> tag of SMIL specifies the start time and duration of a set of media objects that are to be presented in parallel.” *Wan*, [0059] (emphasis added); *see also id.*, [0057] (“An existing language such as *XSL Transformations (XSLT)* may be used for specifying the templates.”) (emphasis added).

132. *Wan* explains the advantages of an XML-based metadata stream with a separate content stream, and how that is more “efficient” and can work well with a “low bandwidth communication link.” It is in this context that it describes the use XSLT to support the creation of metadata templates that are separate from content. *Wan*, [0003]-[0004], [0038]-[0039]. A POSITA would have thus been motivated to use XSLT in the generation of separate content streams such as “the content-related first data segments and content-related second data segments.” Accordingly, it is my opinion that *Comps* in view of *Wan* renders claim 6 obvious.

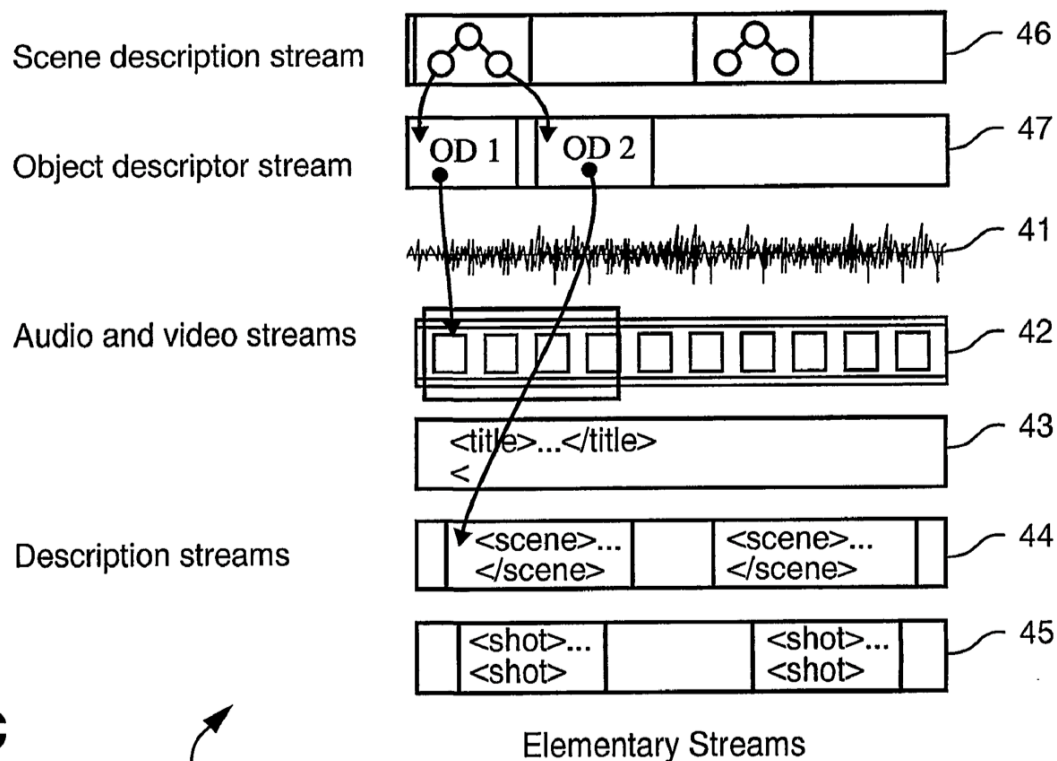
2. Claim 7

- a. **“The method as claimed in claim 1, further comprising: determining, based upon a content-related marker, which content-related first data segments and content-related second data segments are to be output first in time.”**

133. It is my opinion that *Comps* in view of *Wan* disclose this element. As I discuss in section X.A.1, *Comps* discloses all of the elements in claim 7 except determining, based upon a content-related marker, which content-related first data segments and content-related second data segments are to be output first in time. The specification of the '794 patent describes:

[I]t can be beneficial to determine the first and second data segments S1, S2 to be output first in time in each case within the first and second files D1, D2 on the basis of a marker M....In the example according to FIG.4, a first and a second marker M1, M2 are inserted, *the first marker M1 being identified by means of a text ‘Start 1’ and the second marker by way of a text ‘Start 2’*.

'794Pat, 6:50-63 (emphasis added). *Wan* similarly describes markers within the Scene description stream 46. See *Wan*, Figure 4C. “Also included is a high-level scene description stream 46 comprising (compound or primitive) nodes of media objects and having leaf nodes (which are primitive media objects) that point to object descriptors OD_n that make up an object descriptor stream 47.” *Wan*, [0053].

**Fig. 4C**

40

Wan, Fig. 4c.

134. *Wan* explains that the scene markers are used to determine temporal relationships between stream elements. “An example of this is a description of a particular scene in a movie, that provides for multiple camera angles to be viewed, thus permitting viewer access to multiple video streams for which only one video stream may, practically speaking, be viewed in real-time running of the movie. *This is to be contrasted with arbitrary descriptions which have no definable temporal relationship with the streamed content.*” *Wan*, [0054] (emphasis added). This disclosure teaches to a POSITA that the markers determine which content from

different segments are output first in time. Accordingly, *Comps* in view of *Wan* disclose this feature.

3. Claim 8

- a. **“The method as claimed in claim 7, wherein at least one of a scene change in an image sequence and a change of speaker in a speech sequence is identified by way of at least one of the first content-related markers and the second content-related markers.”**

135. *Comps* in view of *Wan* disclose this element. *Comps* discloses all of the elements in claim 8 except “wherein at least one of a scene change in an image sequence and a change of speaker in a speech sequence is identified by way of at least one of the first content-related markers and the second content-related markers.” As discussed in section X.B.2, a POSITA would have understood, for example, that *Wan* discloses “content-related markers” with this scene description stream 46.¹² See *Wan*, Figure 4C, [0053].

¹² To the extent that this claim is not indefinite due to lack of antecedent basis of “the first content-related markers” or “the second content-related markers,” *Comps* in combination with *Wan* disclose this element. While claim 8 is dependent from claim 7 that is dependent from claim 1, neither claim 1 nor claim 7 recites “first content-related markers” or “second content-related markers.”

136. *Wan* discloses, a multimedia stream 40 includes an audio stream 41, a video stream 42, and a scene description stream 46, as shown in Figure 4C below. *Id.*, [0053]. *Wan* explains the advantages of an XML-based metadata stream with a separate content stream, and how that is more “efficient” and can work well with a “low bandwidth communication link.” *Wan*, [0003]-[0004], [0038]-[0039]. It is in this context that *Wan* describes the use of a scene description stream 46 that is separate from content that can be used for transmitting *updates to scene graphs*, which are “content-related markers.” *Id.*, [0047] (emphasis added).

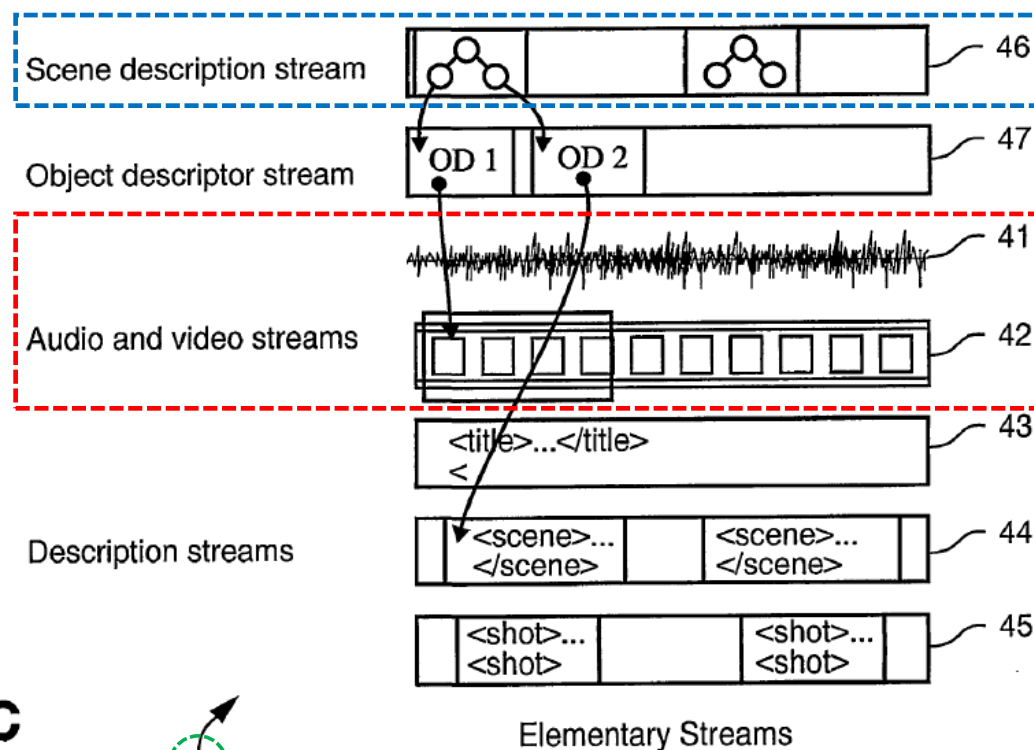


Fig. 4C



Wan, Fig. 4C (annotated).

137. *Comps*’ teaches that a display of the pictures represented by fields JPEG1 and JPEG2 would be identified by way of fields JPEG1:JPEG_{d1} and JPEG2:JPEG_{d2}, respectively. See *Comps*, 3:31-37. This disclosure of media segments in a stream corresponds to *Wan*’s teaching of media streams 41 and 42. *Wan*, Figure 4C, [0053]. Thus, *Comps* in view of *Wan*’s teaching of use of the scene description stream 46 for transmitting updates to scene graphs, discloses “a scene change in an image sequence is identified by way of ... content-related markers” as recited in claim 8.

4. Claim 22

138. *Comps* in view of *Wan* teaches all the elements of Claim 22.

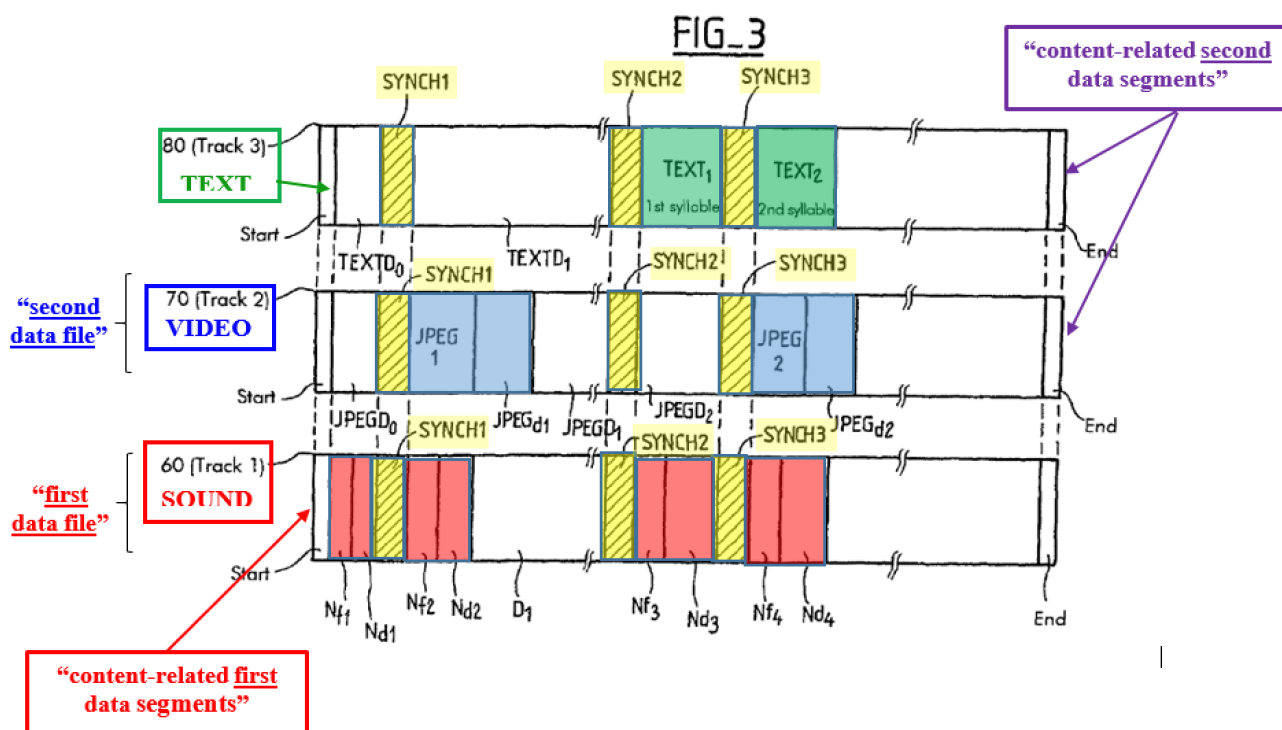
a. **22[pre]: “A method for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, the method comprising:”**

139. This element is substantively identical to that in claim 1[pre]. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.1.a.

b. **22[a][1]: “determining an order for the content-related first data segments to be read out from the first data file and an order for the content-related second data segments to be read out from the second data file,”**

140. *Comps* discloses this element. *Comps* discloses “*the first data file and the second data file are read simultaneously and successive event-related [synchronization] commands in the first data file and the second data file are*

executed.” Comps, Claim 4 (emphasis added). The microprocessors “sequentially” execute the synchronization commands SYNCH1, SYNCH2, SYNCH3 to play the sound segments and to display the picture segments and/or text segments in the pre-determined order their respective segments are stored in the tracks 60, 70, 80, as depicted below in Figure 3, respectively. *Id.*, 3:5-7; *see also id.*, 3:8-43, 4:3-17. For example, the sound segments are played in the order of fields $Nf_1:Nd_1$, $Nf_2:Nd_2$, $Nf_3:Nd_3$, $Nf_4:Nd_4$, as illustrated in red in Figure 3 below, and picture segments are displayed in the order of fields JPEG1 and JPEG2, in blue below,. *Comps* expressly discloses this element.



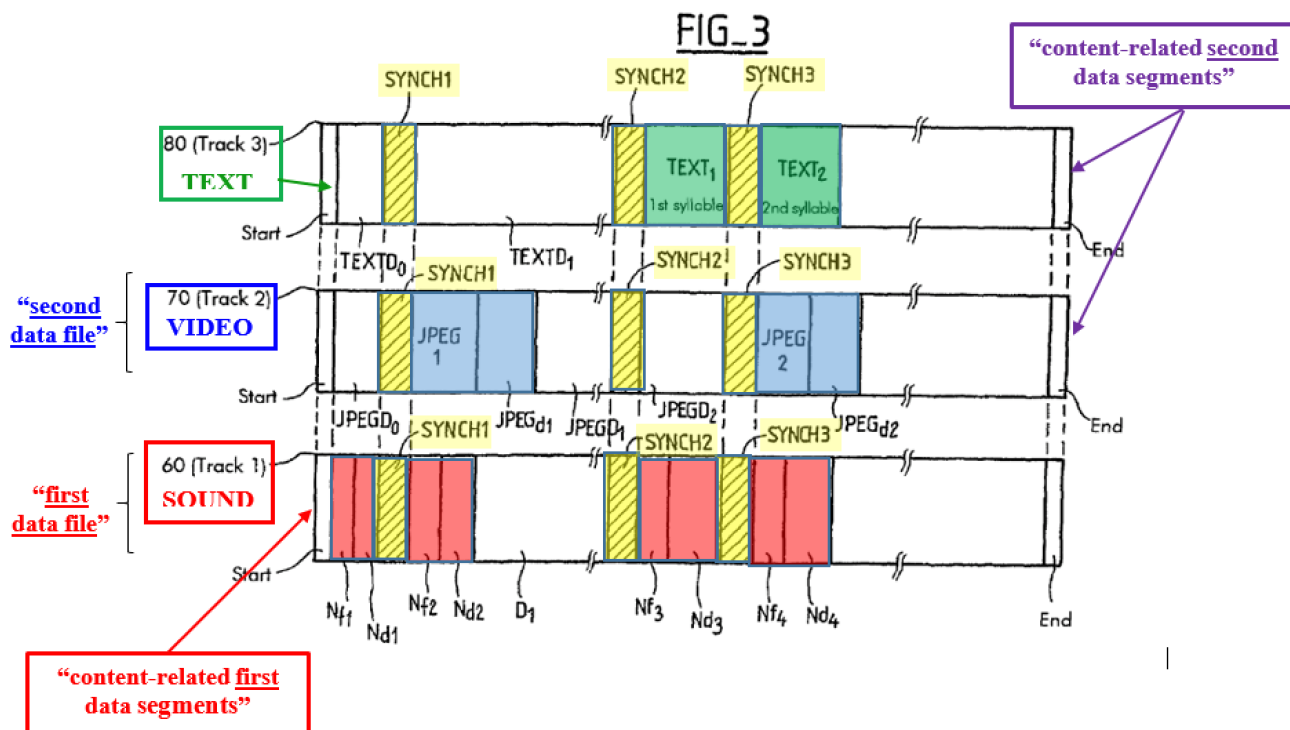
Comps, Fig. 3 (annotated).

- c. **22[a][2]: “the content-related first data segments being ordered according to corresponding first content-related markers and the content-related second data segments being ordered according to corresponding second content-related markers;”**

141. As discussed in section X.B.2, the Scene description stream 46 of *Wan* discloses “content-related markers.” *Wan*, Figure 4C, [0053] (“Also included is a high-level scene description stream 46 comprising (compound or primitive) nodes of media objects and having leaf nodes (which are primitive media objects) that point to object descriptors OD_n that make up an object descriptor stream 47.”). *Wan* also describes that those markers as part of the scene description stream define a temporal relationship among the streamed content. *Id.*, [0054] (“This is to be contrasted with arbitrary descriptions which have no definable temporal relationship with the stream content.”).

142. *Comps* describes that each track 60, 70, 80 contains corresponding data segments, *e.g.*, sound segments played after the Start field in the order Nf₁:Nd₁, Nf₂:Nd₂, Nf₃:Nd₃, Nf₄:Nd₄ for the sound track 60, picture segments are displayed after the Start field in the order JPEG₁:JPEG_{d1}, JPEG₂:JPEG_{d2} for the picture track 70, and text segments are displayed after the Start field in the order of TEXT₁, TEXT₂ for the text track 80. *Id.*, 3:22-41. The sound fields Nf₁:Nd₁, Nf₂:Nd₂, Nf₃:Nd₃, Nf₄:Nd₄ are “first content-related markers” and the picture fields JPEG₁:JPEG_{d1}, JPEG₂:JPEG_{d2}, and text fields TEXT₁, TEXT₂ are “second content-

related markers.” *Id.*, 3:22-37. *Comps* in view of *Wan*, therefore, discloses this element.



Comps, Fig. 3 (annotated).

- d. 22[b]: “assigning, according to an assignment rule, each of the ordered content-related second data segments that are read out from the second data file with a corresponding one of the ordered content-related first data segments that are read out from the first data file; and”

143. This element is substantively identical to that in claim 1[a][2].

144. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.1.c.

- e. 22[c]: “sequentially outputting, by a device for synchronizing content-related data, in a chronological sequence, each of the assigned content-related first data segments and the assigned content-related second data segments.”

145. This element is substantively identical to that in claim 1[a][1].

146. Accordingly, it is my opinion that *Comps* discloses this element for the same reasons discussed in section X.A.1.b.

2. Obviousness Based on *Comps* and *Wan*

147. There are compelling motivations to combine the relevant multimedia content synchronizing features from *Comps* and *Wan*. For example, a POSITA would have found it obvious to combine *Comps*’s teachings of a method of synchronizing different types of data segments with *Wan*’s teachings of use of XSLT-based presentation template and use of the scene description stream for transmitting updates to scene graphs.

148. First, a POSITA would have looked to *Wan* because it is analogous art to *Comps*. *Comps*, 2:6-9 (the invention provides a method of synchronizing multimedia data including “at least two separate data files (track1, track2) referred to as the first file, the second file, etc.”); *Wan*, [0043-0045] (disclosing that descriptions 33 are synchronized with associated audio files 31 and video files 32 using an internal time graph constructed by an encoder 34). It is my opinion that both *Comps* and *Wan* are directed to a method of synchronizing multimedia data.

See, e.g., *Comps*, Title, Abstract, 1:4-5, 2:6-19; *Wan*, [0045, 0052, 0063, 0068], Claim 13.

149. Second, *Comps* itself provides motivation to look to *Wan*. *Comps* recognizes the issue of resource constraints for processing multimedia files, specifically disclosing “the size of multimedia files and the computation power needed to process them constitute a problem.” *Comps*, 1:5-10, 1:56-58 (“The available memory is limited for reasons of overall size and battery life [of mobile terminals].”). As a solution to this issue, *Wan* discloses, for example, that “[t]he description does not have to be repeated as frequently as the AV content because the description changes much less frequently and, at the same time, *consumes significantly fewer computing resources* at the decoder end.”) *Wan*, [0073]. *Wan* also explains that separating the metadata from the content so that the XML metadata document can be parsed as soon as a sufficient portion of it is received will be “most useful in the case of a low bandwidth communication link and/or a device with very limited resources.” *Wan*, [0038]-[0039]. Accordingly, it is my opinion that a POSITA would have been motivated to look to *Wan* to solve the above-discussed issue recognized by *Comps*.

150. Third, synchronizing multimedia data was a well-known concept. ’794*Pat*, 2:4-6 (admitting that a Bitstream Binding Language (BBL) method, by which a synchronization can be performed, is known from prior art); *Comps*, 1:4-5

(disclosing a method of synchronizing different types of data in a multimedia file); *Wan*, [0063] (disclosing a method of synchronizing AV object and fragment of descriptions); *Ahn*, [0003] (disclosing a method of synchronizing metadata with multimedia contents); *Kim*, [0016] (disclosing a method of synchronizing lyric and user input data with audio data); *Kristjansson*, 2:19-25 (disclosing a method of synchronizing video and audio data in a unified manner); *Thong*, Title (disclosing a content-based synchronization method for data streams); *Abbott*, Abstract (disclosing a synchronization method to minimize a time offset between audio and video data); SMIL (Synchronized Multimedia Integration Language) has been a W3C standard since 1998 when version 1.0 was defined (<https://www.w3.org/TR/SMIL1/>). The Timing and Synchronization Module (<https://www.w3.org/TR/2005/REC-SMIL2-20051213/smil-timing.html>) describes how SMIL solves “fundamental media synchronization problems and defined a powerful way of choreographing multimedia content” (10.2 Introduction).

151. Especially given the strong background knowledge about synchronization of multimedia content that a POSITA would be familiar with, and the use of well-known technologies such as XML and streaming content, the specific combination of *Comps* with *Wan* would have been technically straightforward and there would be a reasonable expectation of success. In particular, *Comp* discloses multiple streaming tracks of content in different files, and *Wan* also discloses

multiple streaming tracks of content in different files as well as an additional scene description stream (*see Wan*, Figure 3, [0043]). A combination of *Comp* and *Wan* would have simply added the scene description stream of *Wan* to the multiple tracks of *Comp* in a way that was already disclosed by *Wan*. The result would have been predictable since it would use the technology described in *Wan* along with the multiple tracks, files, and synchronization commands described in *Comps*.

C. Ground 3: *Comps* and *Ahn* Render Obvious Claim 14 of the '794 Patent

152. *Comps* in combination with *Ahn* renders obvious claim 14. Ground 3 is not duplicative of Grounds 1, 2, and 4 because it is the only Ground based on *Comps* and *Ahn*.

1. Claim 14

a. “The method as claimed in claim 13, wherein the metadata include metadata encoded according to the XML standard.”

153. As discussed in section X.A.1 of Ground 1, which is incorporated by reference herein, *Comps* discloses “content-related first data segments of a first data file” and “content-related second data segments of a second data file” as recited in claim 1. *Comps* does not disclose that the content-related data segments are “metadata encoded according to the XML standard.”

154. But the SYNCH commands in *Comps* would be understood by a POSITA to be metadata since it described aspects of the data – namely, how the

different tracks related to each other. Furthermore, it was well-known to a POSITA at the time of the '794 patent that content-related data may be metadata. *Wan*, [0002] (“multimedia may be conveniently referred to as being formed of content and descriptions. The description is typically formed by metadata”); *Ahn*, [0003] (“The present invention relates to an apparatus and method for synchronizing metadata with multimedia contents, and transmitting them.”); *Seong*, [0003] (“The present invention relates to [] a transcoding system and method that can effectively synchronize segmentation metadata with AV contents”).

155. Specifically, *Ahn* discloses “prior art only disclose metadata description methods and do not include synchronization and transmission methods of *the multimedia contents and related metadata*” *Ahn*, [0005]), thus “the present invention [] provide[s] a method for *synchronizing metadata with multimedia contents*, and transmitting them.” *Id.*; [0006] (emphasis added). *Ahn* discloses that “[t]he metadata synchronization format converter 420 *converts the XML-language metadata transmitted from the metadata authoring unit 300 into binary codes, and converts them into a synchronization format.*” *Id.*, [0030] (emphasis). Thus, it is my opinion that *Comps* and *Ahn* render obvious claim 14.

2. Obviousness Based on *Comps* and *Ahn*

156. There are compelling motivations to combine the relevant multimedia content synchronizing features from *Comps* and *Ahn*. For example, a POSITA

would have found it obvious to combine *Comps*' teachings of a method of synchronizing different types of data with *Ahn*'s teachings of use of XML-language metadata.

157. First, a POSITA would have looked to *Ahn* because it is analogous art to *Comps*. *Comps*, 2:6-9 (the invention provides a method of synchronizing multimedia data including "at least two separate data files (track1, track2) referred to as the first file, the second file, etc."); *Ahn*, [0033] (disclosing a method of converting the metadata into a synchronization format and synchronize it with multimedia data). Both *Comps* and *Ahn* are directed to a method of synchronizing multimedia data. See *Comps*, Title, Abstract, 1:4-5, 2:6-19; *Ahn*, [0003, 0006, 0033], Title, Abstract.

158. Second, *Comps* itself provides motivation to look to *Ahn*. *Comps* recognizes the issue of synchronizing the different types of data. For example, *Comps* describes that "[t]he problem is therefore to match the text to the music and the pictures, i.e. to *synchronize the different types of data contained in the same multimedia file*." *Comps*, 1:26-28 (emphasis added). As a solution to this issue, *Ahn* discloses, for example, a "method for synchronizing metadata with multimedia contents [*i.e.*, different types of data]." *Ahn*, [0003]. Thus, a POSITA would have been motivated to look to solve the above-discussed issue recognized by *Comps*.

159. Third, synchronizing multimedia data was a well-known concept. '794*Pat*, 2:4-6 (admitting that a Bitstream Binding Language (BBL) method, by which a synchronization can be performed, is known from prior art); *Comps*, 1:4-5 (disclosing a method of synchronizing different types of data in a multimedia file); *Wan*, [0063] (disclosing a method of synchronizing AV object and fragment of descriptions); *Ahn*, [0003] (disclosing a method of synchronizing metadata with multimedia contents); *Kim*, [0016] (disclosing a method of synchronizing lyric and user input data with audio data); *Kristjansson*, 2:19-25 (disclosing a method of synchronizing video and audio data in a unified manner); *Thong*, Title (disclosing a content-based synchronization method for data streams); *Abbott*, Abstract (disclosing a synchronization method to minimize a time offset between audio and video data).

160. Finally, a POSITA would have had a reasonable expectation of success in combining *Comps* with *Ahn* since use of XML was such a standard and well-known technology. A POSITA would know that content metadata would need to be in some structured format, and so would look to standard formats that existing tools and parsers supported. XML is one of those well-known formats and would be one of a small number of formats to use to store metadata for content such as that disclosed in *Comps*.

D. Ground 4: *Comps* and *Kim* Render Obvious Claims 18 and 19 of the '794 Patent

161. *Comps* in combination with *Kim* renders obvious claims 18 and 19.

Ground 4 is not duplicative of Grounds 1, 2, and 3 because it is the only Ground based on *Comps* and *Kim*.

1. Claim 18

a. “The method of claim 1, wherein the assignment rule is maintained in a file separate from the first data file and the second data file.”

162. It is my opinion that *Comps* in view of *Kim*, disclose this element.

163. As discussed in section X.A.1 of Ground 1, which is incorporated by reference herein, *Comps* discloses “an assignment rule,” “a first data file,” and “a second data file” of claim 1. *Comps* does not disclose that the “assignment rule is maintained in a file separate from the first data file and the second data file.”

164. *Kim* describes that lyric or user data related with a song recorded on a rewritable disk such as DVD-RW is “segmentized into several units. Each segmentized unit is *linked* with each corresponding part of the recorded song through segment synchronizing information for the purpose of minute synchronization of song and additional data.” *Kim*, [0030] (emphasis added). The segmentized lyric or user data may be recorded “*in a file separated from the management information file storing management information for controlling reproduction of all of recorded*

audio data.” Id., Claim 7 (emphasis added); *see also* Claim 6; [0032, 0034, 0040-0045].

165. A POSITA would look to *Kim* and learn that any kind of information regarding management of information files could be stored in a file that was separate from the information files. This would include the SYNCH commands of *Comps*, where is information regarding management of content information. Thus, the combination of *Comps* and *Kim* would store the SYNC ids of *Comps* in a file separate from its content files, thus rendering claim 18 obvious.

166. It is my opinion that this combination would have a reasonable expectation of success since deciding how to structure data in a computer system, such as storing data in the same file or separate files was a well-known technique and one of just two possibilities.

2. Claim 19

a. “The device of claim 9, wherein the assignment rule is maintained in a file separate from the first data file and the second data file.”

167. This element is substantively identical to that in claim 18. Accordingly, *Comps* in view of *Kim* disclose this element for the same reasons discussed in section X.D.1.

3. Obviousness Based on *Comps* and *Kim*

168. It is my opinion that there are compelling motivations to combine the relevant multimedia content synchronizing features from *Comps* and *Kim*. For

example, a POSITA would have found it obvious to combine *Comps*' teachings of a method of synchronizing different types of data with *Kim*'s teachings of use of a different file for the assignment rule.

169. First, a POSITA would have looked to *Kim* because it is analogous art to *Comps*. *Comps*, 2:6-9 (the invention provides a method of synchronizing multimedia data including "at least two separate data files (track1, track2) referred to as the first file, the second file, etc."); *Kim*, [0016] (disclosing a method of synchronizing additional data such as lyric data and user input data with audio data). Both *Comps* and *Kim* are directed to a method of synchronizing multimedia data. *See Comps*, Title, Abstract, 1:4-5, 2:6-19; *Kim*, [0002, 0016, 0030], Title, Abstract.

170. Second, *Comps* itself provides motivation to look to *Kim*. *Comps* recognizes the issue of synchronizing the different types of data. For example, *Comps* describes that "[t]he problem is therefore to match the text to the music and the pictures, i.e. to *synchronize the different types of data contained in the same multimedia file.*" *Comps*, 1:26-28 (emphasis added). As a solution to this issue, *Kim* discloses, for example, "a synchronizing method that records additional data such as lyric data and user input data to be synchronized minutely with audio data [*i.e., different types of data*] on a rewritable recording medium." *Kim*, [0016]. Thus, it is my opinion that a POSITA would have been motivated to look to *Kim* to solve the above-discussed issue recognized by *Comps*.

171. Third, synchronizing multimedia data was a well-known concept. '794*Pat*, 2:4-6 (admitting that a Bitstream Binding Language (BBL) method, by which a synchronization can be performed, is known from prior art); *Comps*, 1:4-5 (disclosing a method of synchronizing different types of data in a multimedia file); *Wan*, [0063] (disclosing a method of synchronizing AV object and fragment of descriptions); *Ahn*, [0003] (disclosing a method of synchronizing metadata with multimedia contents); *Kim*, [0016] (disclosing a method of synchronizing lyric and user input data with audio data); *Kristjansson*, 2:19-25 (disclosing a method of synchronizing video and audio data in a unified manner); *Thong*, Title (disclosing a content-based synchronization method for data streams); *Abbott*, Abstract (disclosing a synchronization method to minimize a time offset between audio and video data).

XI. SECONDARY CONSIDERATIONS

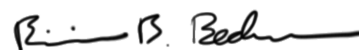
172. Based on my expertise and experience of the industry and review of the asserted prior art and other references cited in my declaration, I am not aware of any evidence of secondary considerations that would support a finding of non-obviousness. The asserted prior art demonstrates that the elements recited in the challenged claims of the '794 patent were readily available in the prior art, and there was no evidence of failure by others.

XII. CONCLUSION

173. Based on my review and above analysis of the prior art, it is my opinion that claims 1-5, 9-13, 15-17, and 20-22 of the '794 patent are anticipated by *Comps*; claims 6-8 of the '794 patent would have been obvious to a POSITA in light of *Comps* and *Wan*; claim 14 of the '794 patent would have been obvious to a POSITA in light of *Comps* and *Ahn*; and claims 18 and 19 of the '794 patent would have been obvious to a POSITA in light of *Comps* and *Kim*.

174. I declare under penalty of perjury that the foregoing is true and correct. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this 5 day of May, 2023, in Windsor, Massachusetts.



Benjamin B. Bederson, Ph.D.

Dr. Benjamin B. Bederson

University of Maryland, College Park
Professor Emeritus, Computer Science

bederson@umd.edu
www.cs.umd.edu/~bederson

I have read the following and certify that it is a current and accurate statement of my professional record as of May 4, 2023.



1. PERSONAL INFORMATION

EDUCATION

9/89 – 6/92	Ph.D., New York University, Computer Science Dept.	New York, NY
9/87 – 8/89	M.S., New York University, Computer Science Dept.	New York, NY
9/82 – 6/86	B.S., Rensselaer Polytechnic Institute, Computer Science Dept. Minor, Electrical Engineering	Troy, NY

EMPLOYMENT

3/22 – Present	J.S. Bryant School, Inc. Therapeutic high school for LGBTQ+ students Co-founder, Board member, Treasurer and Clerk	Cummington, MA
1/19 – Present	Bederson Expert Services, Inc. Technical consulting	Windsor, MA
1/98 – Present	University of Maryland, Computer Science Dept./UMIACS 7/19 – Present: Professor Emeritus, Computer Science Department. 4/14 – 6/18: Associate Provost of Learning Initiatives, and Executive Director of Teaching and Learning Transformation Center 4/13 – 4/14: Special Advisor to the Provost on Technology and Educational Transformation 8/11 – 6/19: Professor 9/04 – 7/11: Associate Professor 6/00 – 9/06: Director of the Human-Computer Interaction Lab (HCIL) 1/98 – 8/04: Assistant Professor	College Park, MD
1/06 – Present	International Children's Digital Library Foundation Co-founder and Technical Director for this library of exemplary free online children's books from around the world. 7/17 – present: President, secretary and treasurer	Windsor, MA
10/14 – 4/23	Hazel Analytics, Inc. Co-founder & Board member: regulatory data analytics 10/14 – 12/19: Chief Technology Officer	Seattle, WA
9/06 – 10/14	Zumobi, Inc. Co-founder of VC-funded startup for mobile apps and advertising. 9/06–1/08: VP Client Technologies 1/08–10/14: Chief Scientist	Seattle, WA
6/03 – 11/08	Windsor Interfaces, Inc. Founder: Small company to commercialize HCIL products.	Chevy Chase, MD
1/95 – 8/97	University of New Mexico, Computer Science Department Assistant Professor in the Computer Science Department.	Albuquerque, NM
9/92 – 8/94	Bell Communications Research (Bellcore) Research Scientist in Computer Graphics and Interactive Media group.	Morristown, NJ

6/93 – 7/94	New York University, Media Research Laboratory Visiting Research Scientist.	New York, NY
9/90 – 8/92	Vision Applications, Inc. Research Scientist.	New York, NY
9/88 – 6/90	New York University, Computer Science Department Teaching Assistant.	New York, NY
1/87 – 5/87	North Slope Borough/North Slope Higher Education Center Instructor. Taught Computers & Society to rural Eskimo population.	Barrow, AK

2. RESEARCH, SCHOLARLY, AND CREATIVE ACTIVITIES

Note about publications:

- In all references, my name is in bold.
- Authors are displayed as published. Authors are usually listed in order of decreasing contribution – except that sometimes the last author is the lead advising faculty member. I put an asterisk next to my name when I served that role.
- Students that I supervised are underlined.

A. BOOKS

i. Authored Books

1. Herrnson, P.S., Niemi, R.G., Hanmer, M.J., **Bederson, B. B.**, Conrad, F.G., Traugott, M., (2008) “Voting Technology and the Not-So-Simple Act of Casting a Ballot,” *Brookings Institute Press*.

ii. Edited Collections

1. Teevan, J., Jones, W., **Bederson, B. B.** (Eds.) (2006) *Communications of the ACM (CACM)*, Editors of special issue on “Personal Information Management”, 49 (1), ACM Press.
2. **Bederson, B. B.**, Shneiderman, B. (Eds.) (2003). *The Craft of Information Visualization: Readings and Reflections*. San Francisco: Morgan Kaufmann.

iii. Book Chapters

1. Fails, J., Druin, A., **Bederson, B. B.**, Weeks, A., & Rose, A. (2009) A Child’s Mobile Digital Library: Collaboration, Community and Change. Druin, A. (Ed.) *Mobile Technology for Children: Designing for Interaction and Learning*, Morgan Kaufmann Press.
2. Herrnson, P.S., Niemi, R.G., Hanmer, M.J., **Bederson, B. B.**, Conrad, F.G., Traugott, M.W. (2008) The Current State of Electronic Voting in the United States. Chen, H., Brandt, L., Dawes, S., Gregg, V., Hovy, E., Macintosh, A., Traunmüller, R., Larson, C. (Eds.) *Digital Government: Advanced Research and Case Studies*, Springer, 157-180.
3. Karlson, A., **Bederson, B. B.**, & Contreras-Vidal, Jose L. (2007) Understanding One Handed Use of Mobile Devices. Lumsden, Jo (Ed.), *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, Idea Group Reference, 86-101.
4. Hutchinson, H., Druin, A., **Bederson, B. B.** (2007) Designing Searching and Browsing Software for Elementary-Age Children. Lazar, J. (Ed.), *Universal Usability*. John Wiley. 1618-1630.
5. Druin, A., **Bederson, B. B.**, Boltman, A., Muira, A., Knotts-Callahan, D., & Platt, M. (1999). Children As Our Technology Design Partners. A. Druin (Ed.), *The Design of Children's Technology* (51-72). San Francisco: Morgan Kaufmann.
6. Hollan, J. D., **Bederson, B. B.**, & Helfman, J. (1998). Information Visualization. Helander, M., Landauer, T. K., & Prabhu, P. V. (eds.), *The Handbook of Human Computer Interaction* (Chap. 2, 33-48). Amsterdam: Elsevier Press.

7. **Bederson, B. B.**, Hollan, J. D., Stewart, J., Rogers, D., Druin, A., Vick, D., Ring, L., Grose, E., & Forsythe, C. (1997). A Zooming Web Browser. C. Forsythe, J. Ratner, & E. Grose (eds.), *Human Factors and Web Development* (Chap. 19, 255-266). New Jersey: Lawrence Erlbaum.
8. **Bederson, B. B.**, & Druin, A. (1995). Computer Augmented Environments: Physical Spaces to Enrich Our Lives. Jakob Nielsen (eds.), *Advances in Human-Computer Interaction* (Vol. 5, Chap. 2, 37-66). New Jersey: Ablex Press.
9. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1993). A Miniaturized Space-Variant Active Vision System: Cortex-I. Richard Mammone (eds.), *Artificial Neural Networks for Speech and Vision*. Chapman-Hall Publishers.
10. Wallace, R. S., Ong, P.-W., **Bederson, B. B.**, & Schwartz, E. L. (1993). Connective Graphs in Space-Variant Active Vision. George A. Bekey & Ken Goldberg (eds.), *Neural Networks in Robotics*. Kluwer Academic Publishers.

B. REFEREED ARTICLES

i. Journals

1. **Bederson, B. B.**, Jin, G. Z., Leslie, P., Quinn, A. J., Zou, B. (2017) "Incomplete Disclosure: Evidence of Signaling and Countersignaling", in *American Economic Journal: Microeconomics*, (in press).
2. Anderson, L. B., Gardner, E. E., Wolvin, A., Kirby-Straker, R., Yalcin, A., & **Bederson, B. B.*** (2016), "Incorporating Learning Analytics into Basic Course Administration: How to Embrace the Opportunity to Identify Inconsistencies and Inform Responses", in *Journal of the Association for Communication Administration*, 35 (1), Winter-Spring 2016, pp 2-14.
3. Lee, T. Y., **Bederson, B. B.*** (2016) "Give the people what they want: studying end-user needs for enhancing the web", *PeerJ Computer Science*, Nov, 2016, 2:e91 <https://doi.org/10.7717/peerj-cs.91>
4. Yalcin, M. A., Elmqvist, N. & **Bederson, B. B.***. (2015) "AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations". in *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 688–697, 2015.
5. Hu, C., Resnik, P., & **Bederson, B. B.*** (2014) "Crowdsourced Monolingual Translation", in *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21 (4), Aug 2014, pp. 22:1–22:35.
6. Lee, T.Y., Mauriello, M., Ahn, J. & **Bederson, B. B.*** (2014) "CTArcade: Computational Thinking with Games in School Age Children", in *International Journal of Child-Computer Interaction*, 2 (1), Jan 2014, 26–33.
7. Bonsignore, E., Quinn, A. J., Druin, A. & **Bederson, B. B.*** (2013) "Sharing Stories 'In The Wild': A Mobile Storytelling Case Study using StoryKit", in *Transactions on Computer-Human Interaction (TOCHI)*, ACM, 20 (3), July 2013, Article 18.
8. Resnik, P., Buzek, O., Kronrod, Y., Hu, C., Quinn, A.J. & **Bederson, B. B.** (2013) "Using Targeted Paraphrasing and Monolingual Crowdsourcing to Improve Translation", in *Transactions on Intelligent Systems and Technology*, ACM, 4 (3), July 2013, Article 38, 21 pages.
9. **Bederson, B. B.** (2011) "The Promise of Zoomable User Interfaces", in *Behaviour & Information Technology*, Taylor & Francis, Nov-Dec 2011, 30 (6), 853-866.
10. Druin, A., **Bederson, B. B.**, Rose, A., Weeks, A. (2009) "From New Zealand to Mongolia: Co-Designing and Deploying a Digital Library for the World's Children", in *Children, Youth and Environments*, University of Colorado, 19 (1), 34-57.
11. Conrad, F.G., **Bederson, B. B.**, Lewis, B., Peytcheva, E., Traugott, M.W., Hanmer, M.J., Herrnson, P.S., & Niemi, R.G. (2009) "Electronic voting eliminates hanging chads but introduces new usability challenges", in *International Journal of Human-Computer Studies (IJHCS)*, Elsevier Press, 67 (1), 111-124.
12. Hanmer, M.J., Park, W-H., Traugott, M.W., Niemi, R.G., Herrnson, P.S., **Bederson, B. B.**, and Conrad, F.C. (2008) "Losing Fewer Votes: The Impact of Changing Voting Systems on Residual Votes", in *Political Research Quarterly*, Sage Publications, September, 2008. doi:10.1177/1065912908324201.

-
13. **Bederson, B. B.**, Clamage, A., Plaisant, C. (2008) Enhancing In-Car Navigation Systems with Personal Experience, in *Transportation Research Record (TRR)*, The National Academies, 2064 (2008) 33-42.
 14. Herrnson, P.S., Niemi, R.G., Hanmer, M.J., Francia, P.L., **Bederson, B. B.**, Conrad, F.G., Traugott, M. (2008) Voter's Evaluations of Electronic Voting Systems: Results from a Usability Field Study, *American Politics Research*, Sage Journals, 36 (4), July 2008, 580-611.
 15. Cockburn, A., Karlson, A., & **Bederson, B. B.*** (2008) A Review of Overview+Detail, Zooming, and Focus+Context Interfaces, *ACM Computing Surveys*, ACM Press, December 2008, 41 (1).
 16. Kang, H., **Bederson, B. B.**, & Suh, B. (2008) Capture, Annotate, Browse, Find, Share: Novel Interfaces for Personal Photo Management, *International Journal of Human-Computer Interaction (IJHCI)*, 23 (3), 1-23.
 17. Parr, C.S., Lee, B., **Bederson, B. B.*** (2007) EcoLens: Integration and Interactive Visualization of Ecological Datasets, *Journal of Ecological Informatics*, Elsevier, 2 (1), 61-69.
 18. Suh, B., and **Bederson, B. B.*** (2007) Semi-Automatic Photo Annotation Strategies Using Event Based Clustering and Clothing Based Person Recognition, *Interacting With Computers*, Elsevier, 19 (4), 524-544.
 19. Hutchinson, H., **Bederson, B. B.**, Druin, A., (2007) Supporting Elementary-Age Children's Searching and Browsing: Design and Evaluation Using the International Children's Digital Library, *Journal of the American Society for Information Science and Technology*, John Wiley & Sons, 58 (11), 1618-1630.
 20. Kang, H., Plaisant, C., Lee, B., **Bederson, B. B.***, (2007) NetLens: Iterative Exploration of Content-Actor Network Data, *Information Visualization*, Palgrave Macmillan, 6 (1), 18-31.
 21. Lee, B., Parr, C.S., Plaisant, C., **Bederson, B. B.**, Veksler, V.D., Gray, W.D., Kotfila, C. (2006) TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts, *IEEE Transactions on Vision and Computer Graphics (TVCG)*, IEEE Press, 12 (6), 1414-1426.
 22. Plaisant, C., **Bederson, B. B.**, Clamage, A., Hutchinson, H.B., Druin, A. (2006) Shared Family Calendars: Promoting Symmetry and Accessibility, *Transactions on Computer-Human Interaction*, New York: ACM, 13 (3), 313-346.
 23. Shneiderman, B., **Bederson, B. B.**, and Drucker, S., (2006) Find that photo! Interface strategies to annotate, browse, and share, *Communications of the ACM*, 49 (4), 69-71.
 24. Hutchinson, H.B., Rose, A., **Bederson, B. B.**, Weeks, A.C., Druin, A. (2005) The International Children's Digital Library: A Case Study in Designing for a Multi-Lingual, Multi-Cultural, Multi-Generational Audience, *Information Technology and Libraries*, 24 (1), 4-12.
 25. Herrnson, P.S., **Bederson, B. B.**, Lee, B., Francia, P. L., Sherman, R.M., Conrad, F.G., Traugott, M., Niemi, R.G. (2005) Early Appraisals of Electronic Voting, *Social Science Computing Review*, 23 (3), 274-292.
 26. Hourcade, J. P., **Bederson, B. B.**, Druin, A., & Guimbretière, F. (2004) Differences in pointing task performance between preschool children and adults using mice, *Transactions on Computer-Human Interaction*, New York: ACM, 11 (4), 357-386.
 27. Parr, C. S., Lee, B., & **Bederson, B. B.*** (2004). Visualizations for Taxonomic and Phylogenetic Trees. *Bioinformatics*, Oxford University Press, 20 (17), 2997-3004.
 28. **Bederson, B. B.**, Grosjean, J., Meyer, J. (2004) Toolkit Design for Interactive Structured Graphics, *Transactions on Software Engineering*, New York: IEEE, 30 (8), 535-546.
 29. Chipman, L. E., **Bederson, B. B.**, Golbeck, J. A. (2004) SlideBar: Analysis of a Linear Input Device. *Journal of Behaviour and Information Technology*, London, Taylor & Francis, 23 (1), 1-9.
 30. **Bederson, B. B.**, Clamage, A., Czerwinski, M. P., & Robertson, G. G. (2004) DateLens: A Fisheye Calendar Interface for PDAs, *Transactions on Computer-Human Interaction*, New York: ACM, 11 (1), 90-119.
 31. Hourcade, J. P., **Bederson, B. B.**, & Druin, A. (2004) Building KidPad: An Application for Children's Collaborative Storytelling, *Software: Practice & Experience*, New Jersey: John Wiley & Sons, Ltd, 34 (9), 895-914.

-
32. Druin, A., **Bederson, B. B.**, Weeks, A., Farber, A., Grosjean, J., Guha, M. L., Hourcade, J. P., Lee, J., Liao, S., Reuter, K., Rose, A., Takayama, Y., & Zhang, L. (2003). The International Children's Digital Library: Description and Analysis of First Use. *First Monday*, 8(5).
 33. Druin, A., Revelle, G., **Bederson, B. B.**, Hourcade, J. P., Farber, A., Lee, J., & Campbell, D. (2003) A Collaborative Digital Library for Children: A Descriptive Study of Children's Collaborative Behavior and Dialogue. *The Journal of Computer-Assisted Learning*, 19 (2), 239-248.
 34. Hourcade, J.P., **Bederson, B. B.**, Druin, A., Rose, A., Farber, A., Takayama, Y. (2003). The International Children's Digital Library: Viewing Digital Books Online. *Interacting with Computers*, Elsevier Press, 15 (3), 151-167.
 35. Hornbæk, K., **Bederson, B. B.**, & Plaisant, C. (2002) Navigation Patterns and Usability of Zoomable User Interfaces With and Without an Overview, *ACM Transactions on Computer-Human Interaction*, 9 (4), 362-389.
 36. **Bederson, B. B.**, Shneiderman, B., & Wattenberg, M. (2002) Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, *ACM Transactions on Graphics*, 21 (4), 833-854, ACM Press. – **Won 2021 IEEE Vis Test of Time Award in InfoVis**, <http://ieevis.org/year/2021/info/awards/test-of-time-awards>.
 37. Good, L., & **Bederson, B. B.*** (2002) Zoomable User Interfaces as a Medium for Slide Show Presentations, *Information Visualization*, Palgrave Macmillan, 35-49.
 38. Revelle, G., Druin, A., Platner, M., **Bederson, B. B.**, Hourcade, J.P., & Sherman, L. (2002) A Visual Search Tool for Early Elementary Science Students, *Journal of Science Education and Technology*, 11(1), 49-57.
 39. **Bederson, B. B.**, & Meyer, J. (1998). Implementing a Zooming User Interface: Experience Building Pad++. *Software: Practice and Experience*, 28(10), 1101-1135.
 40. **Bederson, B. B.**, Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, 3-31.
 41. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1995). A Miniaturized Space-Variant Active Vision System: Cortex-I. *Machine Vision and Applications*, 8 (2), 101-109.
 42. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1994). A Miniature Pan-Tilt Actuator: the Spherical Pointing Motor. *IEEE Transactions on Robotics and Automation*, 10(3), 298-308.
 43. Wallace, R. S., Ong, P.-W., **Bederson, B. B.**, & Schwartz, E. L. (1994). Space-Variant Image Processing. *International Journal of Computer Vision*, 13(1), 71-90.

ii. Peer-Reviewed Published Full-Length Conference Papers

1. Yalçın, M. A., Elmqvist, N., & **Bederson, B. B.*** (2017) Raising the Bars: Evaluating Treemaps vs. Wrapped Bars for Dense Visualization of Sorted Numeric Data, *Proceedings of Graphics Interface (GI 2017)*, 41-49. DOI: <https://doi.org/10.20380/GI2017.06>
2. Quinn, A., & **Bederson, B. B.*** (2014) AskSheet: Efficient Use of Web Workers to Support Decision Making, *Proceedings of ACM CSCW (CSCW 2014)*, ACM Press, pp. 1456–1466.
3. Yeh, T., Chang, T-H., Xie, B., Walsh, G., Watkins, I., Wongsuphasawat, K., Davis, L.S., & **Bederson, B. B.*** (2011) Creating Contextual Help for GUIs Using Screenshots, *Proceedings of ACM UIST (UIST 2011)*, ACM Press, 145-154.
4. Quinn, A., & **Bederson, B. B.*** (2011) Human Computation: A Survey and Taxonomy of a Growing Field, *Proceedings of ACM CHI (CHI 2011)*, ACM Press, 1403-1412.
5. Resnik, P., Buzek, O., Hu, C., Kronrod, Y., Quinn, A.J., & **Bederson, B. B.*** (2010) Improving Translation via Targeted Paraphrasing, *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, 127-137.
6. Hu, C., **Bederson, B. B.**, & Resnik, P. (2010) Translation by Interactive Collaboration between Monolingual Users, *Proceedings of Graphics Interface (GI 2010)*, 39-46.

7. Dearman, D., Karlson, A., Meyers, B., & **Bederson, B. B.** (2010) Multi-Modal Text Entry and Selection on Mobile Devices, *Proceedings of Graphics Interface (GI 2010)*, 19-26.
8. Quinn, A., Hu, C., Arisaka, T., & **Bederson, B. B.*** (2008) Readability of Scanned Books in Digital Libraries, *Proceedings of ACM CHI (CHI 2008)*, ACM Press, 705-714. [22% acceptance rate]
9. Karlson, A., & **Bederson, B. B.*** (2008) One-Handed Touchscreen Input for Legacy Applications, *Proceedings of ACM CHI (CHI 2008)*, ACM Press 1399-1408. [22% acceptance rate]
10. **Bederson, B. B.**, Clamage, A., Plaisant, C., (2008) Enhancing In-Car Navigation Systems with Personal Experience, in *Proc. of the Transportation Research Board 87th annual meeting*, The National Academies, Washington, DC (2008) 1-11.
11. Karlson, A., & **Bederson, B. B.*** (2007) ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices, *Proceedings of INTERACT 2007*, 324-338 – **Winner of Brian Shackel Award for making an “outstanding contribution with international impact in the field of HCI”.**
12. Druin, A., Weeks, A., Massey, S., & **Bederson, B. B.** (2007) Children’s Interests and Concerns When Using the International Children’s Digital Library: A four country case study, *Proceedings of the Joint Conference on Digital Libraries (JCDL 2007)*, 167-176. – **Nominated for Best Paper Award, came in #2.** [36% acceptance rate].
13. Parhi, P., Karlson, A., & **Bederson, B. B.*** (2006). “Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices”, *Proceedings of MobileHCI 2006*, ACM Press, 203-210. – **Winner of Best Historical Paper Award, September 2016.**
14. Hutchinson, H. B., **Bederson, B. B.**, & Druin, A. (2006). “The Evolution of the International Children’s Digital Library Searching and Browsing Interface”, *Proceedings of 5th International Conference for Interaction Design and Children (IDC 2006)*, ACM Press, 105-112. [38% acceptance rate]
15. Hutchinson, H., Druin, A., **Bederson, B. B.**, Reuter, K., Rose, A., Weeks, A. (2005) How do I Find Blue Books About Dogs? The Errors and Frustrations of Young Digital Library Users, *Proceedings of HCI 2005*, Las Vegas, NV (CD-ROM).
16. Shneiderman, B. & **Bederson, B. B.** (2005). “Maintaining Concentration to Achieve Task Completion”, *Proceedings of Designing User Experience (DUX 2005)*.
17. Karlson, A., **Bederson, B. B.**, & SanGiovanni, J. (2005). AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices, *CHI Letters*, 7(1), 201-210. [25% acceptance rate] – **Nominated for Best Paper Award.**
18. Khella, A., **Bederson, B. B.*** (2004). Pocket PhotoMesa: A Zoomable Image Browser for PDAs. *In Proceedings of Mobile and Ubiquitous Multimedia (MUM 2004)* ACM Press, 19-24.
19. Lee, B., Parr, C. S., Campbell, D., & **Bederson, B. B.*** (2004). How Users Interact With Biodiversity Information Using TaxonTree. *In Proceedings of Advanced Visual Interfaces (AVI 2004)* ACM Press, 320-327. [25% acceptance rate]
20. Suh, B., Ling, H., **Bederson, B. B.**, & Jacobs, J. W. (2003). Automatic Thumbnail Cropping and Its Effectiveness. *UIST 2003, ACM Symposium on User Interface Software and Technology*, *CHI Letters*, 5(2), 95-104. [22% acceptance rate] – **Winner of Best Student Paper Award.**
21. **Bederson, B. B.**, Lee, B., Sherman, R., Herrnson, P. S., Niemi, R. G. (2003). Electronic Voting System Usability Issues. *CHI 2003, ACM Conference on Human Factors in Computing Systems*, *CHI Letters*, 5(1), 145-152. [16% acceptance rate]
22. Hutchinson, H., Mackay, W., Westerlund, B., **Bederson, B. B.**, Druin, A., Plaisant, C., Evans, H., Hansen, H., Conversy, S., Beaudouin-Lafon, M., Roussel, N., Lacomme, L., Eiderbäck, B., Lindquist, S., Sundblad, Y. (2003). Technology Probes: Inspiring Design for and with Families. *CHI 2003, ACM Conference on Human Factors in Computing Systems*, *CHI Letters*, 5(1), 17-24. [16% acceptance rate]
23. Knudtson, K., Druin, A., Kaplan, N., Summers, K., Chisik, Y., Kulkarni, R., Moulthrop, S., Weeks, H., & **Bederson, B.** Starting an Intergenerational Technology Design Team: A Case Study. *In Proceedings of Interaction Design and Children (IDC 2003)* ACM Press, 51-58.

24. Baudisch, P., Cutrell, P., Robbins, D., Czerwinski, M., Tandler, P., **Bederson, B. B.**, & Zierlinger, A. (2003) Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. *Proceedings of Interact 2003*, Austria: International Federation for Information Processing, 57-64. [33% acceptance rate]
25. Plaisant, C., Grosjean, J., & **Bederson, B. B.*** (2002) SpaceTree: Design Evolution of a Node Link Tree Browser. *Proceedings of Information Visualization Symposium (InfoVis 2002)* New York: IEEE, 57-64. [27% acceptance rate] – **Won 2022 IEEE Vis Test of Time Award in InfoVis**, <https://ieevis.org/year/2022/info/awards/test-of-time-awards>.
26. Suh, B., & **Bederson, B. B.*** (2002) OZONE: A Zoomable Interface for Navigating Ontology Information. *Proceedings of International Conference on Advanced Visual Interfaces (AVI 2002)*, ACM Press, Trento, Italy, 139-143. [30% acceptance rate]
27. Boltman, A., Druin, A., **Bederson, B.**, Hourcade, J.P., Stanton, D., O'Malley, C., Cobb, S., Benford, S., Fast, C., Kjellin, M., & Sundblad, Y. (2002). The nature of children's storytelling with and without technology. *Proceedings of American Educational Research Association (AERA) Conference*, New Orleans.
28. **Bederson, B. B.** (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. *UIST 2001, ACM Symposium on User Interface Software and Technology*, CHI Letters, 3(2), 71-80. [19% acceptance rate]
29. Druin, A., **Bederson, B. B.**, Hourcade, J.P., Sherman, L., Revelle, G., Platner, M., Weng, S. (2001) Designing a Digital Library for Young Children. In *Proceedings of Joint Conference on Digital Libraries (JCDL 2001)* ACM Press, 398-405. [30% acceptance rate]
30. Benford, S., **Bederson, B. B.**, Åkesson, K., Bayon, V., Druin, A., Hansson, P., Hourcade, J. P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K., Stanton, D., Sundblad, Y., & Taxén, G. (2000). Designing Storytelling Technologies to Encourage Collaboration Between Young Children. *CHI 2000, ACM Conference on Human Factors in Computing Systems*, CHI Letters, 2(1), 556-563. [19% acceptance rate]
31. **Bederson, B. B.**, Meyer, J., & Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000, ACM Symposium on User Interface Software and Technology*, CHI Letters, 2(2), 171-180. [26% acceptance rate]
32. **Bederson, B. B.** (2000). Fisheye Menus. *UIST 2000, ACM Symposium on User Interface Software and Technology*, CHI Letters, 2(2), 217-225. [26% acceptance rate]
33. Gandhi, R., Kumar, G., **Bederson, B. B.**, & Shneiderman, B. (2000). Domain Name Based Visualization of Web Histories in a Zoomable User Interface. In *Proceedings of Eleventh International Workshop on Database and Expert Systems Applications – Second International Workshop on Web-Based Information Visualization (WebVis 2000)*, IEEE Computer Society, Los Alamitos, CA (2000), 591-598.
34. **Bederson, B. B.**, & Boltman, A. (1999). Does Animation Help Users Build Mental Maps of Spatial Information? In *Proceedings of Information Visualization Symposium (InfoVis 99)* New York: IEEE, 28-35. [23% acceptance rate]
35. Combs, T. T. A., & **Bederson, B. B.*** (1999). Does Zooming Improve Image Browsing? In *Proceedings of Digital Library (DL 99)* New York: ACM, 130-137. [21% acceptance rate]
36. Stewart, J., **Bederson, B. B.**, & Druin, A. (1999). Single Display Groupware: A Model for Co-Present Collaboration. In *Proceedings of Human Factors in Computing Systems (CHI 99)* ACM Press, 286-293. [25% acceptance rate]
37. Hightower, R. R., Ring, L., Helfman, J., **Bederson, B. B.**, & Hollan, J. D. (1998). Graphical Multiscale Web Histories: A Study of PadPrints. In *Proceedings of ACM Conference on Hypertext (Hypertext 98)* ACM Press, 58-65.
38. Druin, A., Stewart, J., Proft, D., **Bederson, B. B.**, & Hollan, J. D. (1997). KidPad: A Design Collaboration Between Children, Technologists, and Educators. In *Proceedings of Human Factors in Computing Systems (CHI 97)* ACM Press, 463-470. [22% acceptance rate]

-
39. **Bederson, B. B.**, Hollan, J. D., Stewart, J., Rogers, D., Druin, A., & Vick, D. (1996). A Zooming Web Browser. In *Proceedings of SPIE Multimedia Computing and Networking* New York: IEEE, 260-271. <https://www.spiedigitallibrary.org/conference-proceedings-of-spice/2667/0000/Zooming-Web-browser/10.1117/12.235898.short>
 40. **Bederson, B. B.**, & Hollan, J. D. (1995). Advances in the Pad++ Zoomable Graphics Widget. In *Proceedings of USENIX Tcl/Tk '95 Workshop (Tcl/Tk 95)* New York: USENIX, 3.
 41. Furnas, G. W., & **Bederson, B. B.** (1995). Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proc. of Human Factors in Computing Systems (CHI 95)* ACM Press, 234-241. [29% acceptance rate]
 42. **Bederson, B. B.**, & Hollan, J. D. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proc. of User Interface and Software Technology (UIST 94)* ACM Press, 17-26.
 43. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1993). Control & Design of the Spherical Pointing Motor. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 93)* New York: IEEE, 2, 630-636.
 44. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1992). A Miniaturized Active Vision System. In *Proceedings of Eleventh International Conference on Pattern Recognition*, IEEE Press, 58-61.
 45. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (1992). Two Miniature Pan-Tilt Devices. In *Proceedings of IEEE International Conference on Robotics and Automation*, New York: IEEE Press, (1), 658-663.
 46. Wallace, R. S., **Bederson, B. B.**, & Schwartz, E. L. (1992). Voice-Bandwidth Visual Communication Through Logmaps: the Telecortex. In *Proceedings of IEEE Workshop on Applications of Computer Vision* New York: IEEE Press, 4-10.

iii. Other Publications

1. Yalcin, M. A., Elmqvist, N., & **Bederson, B. B.*** (2017). Piled Bars: Dense Visualization of Numeric Data. Poster presented at Graphics Interfaces, 2017.
2. **Bederson, B. B.**, Barker, K., Owens, L., Kim, Y. J. & Palmer, K. (2017) Designing Organizations to Support Tech on Campus. Panel at SXSW Edu, Austin, TX, March 6, 2017.
3. Yalcin, M. A., Elmqvist, N., & **Bederson, B. B.*** (2016). Keshif: Out-of-the-Box Visual and Interactive Data Exploration Environment. In *Proceedings of IEEE VIS 2016 Workshop on Visualization in Practice: Open Source Visualization and Visual Analytics Software*.
4. Yalcin, M. A., Elmqvist, N., & **Bederson, B. B.*** (2016). Cognitive Stages in Visual Data Exploration. In *Proceedings of Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV 2016)* at IEEE VIS 2016.
5. Roberts, S., Swan, L., & **Bederson, B. B.** (2016). "Fishing for Faculty: The Bait and Hooks", Workshop in 2016 Professional and Organizational Development (POD) Conference, Louisville, KY, November, 2016.
6. Anderson, L. B., Gardner, E. E., Wolvin, A., Kirby-Straker, R., Yalcin, A., & **Bederson, B. B.*** (2015), "Incorporating Learning Analytics into Basic Course Administration: How to Embrace the Opportunity to Identify Inconsistencies and Inform Responses". In *NCA 2015 - 101th Annual meeting of the National Communication Association*. **Top Paper in Basic Course Division**.
7. Anderson, L. B., Gardner, E. E., Wolvin, A., Kirby-Straker, R., Yalcin, A., & **Bederson, B. B.*** (2015), "Utilizing learning analytics to improve the basic course: A qualitative analysis of the instructor feedback captured in Revealr". *Paper presented at the annual meeting of the Eastern Communication Association, Philadelphia, PA, June 2015*.
8. Hara, K., Hajiaghayi, M., & **Bederson, B. B.** (2015). FluTCHA: Using Fluency to Distinguish Humans from Computers, short paper and poster, WWW2015, Florence, Italy, May 18-22, 2015.
9. Teasley, S., & **Bederson, B. B.** (2015). Learning Analytics and iSchools: What, Why & Worries, at *iConference*, Session for Interaction and Engagement, March 24-27, 2015, Newport Beach, CA.

-
10. Xu, Z., Goldwasser, D., **Bederson, B. B.**, Lin, J. (2014) Visual Analytics of MOOCs at Maryland, *Poster at Learning @ Scale Conference (L@S '14)*.
 11. **Bederson, B. B.**, Jin, G., Leslie, P., Quinn, A.J., & Zhou, B. (alphabetical order), (2014) "Incomplete Disclosure: Insights from Maricopa", International Industrial Organization Conference, April 11-13, 2014, Chicago.
 12. Lee, T.Y., Mauriello, M.L., Ingraham, J., Sopan, A., Ahn, J., & **Bederson, B. B.*** (2012) "CTArcade: Learning Computational Thinking while Training Virtual Characters Through Game Play", WiP at CHI 2012, ACM Press, 2309-2314.
 13. Cranor, L., **Bederson, B. B.**, & Quesenberry, W. (2012) "Occupy CHI! Engaging U.S. Policymakers", Panel in CHI 2012, ACM Press, 1139-1142.
 14. **Bederson, B. B.**, Quinn, A. J., & Rose, A. (2012) SearchParty: Learning to Search in a Web-based Classroom, Workshop paper at CHI 2012.
 15. Hu, C., Resnik, P., Kronrod, Y., & **Bederson, B. B.*** (2012) Deploying MonoTrans Widgets in the Wild. Note at CHI 2012, ACM Press, 2935-2938.
 16. Hu, C., Resnik, P., Kronrod, Y., Eidelman, V., Buzek, O., & **Bederson, B. B.*** (2011) The Value of Monolingual Crowdsourcing in a Real-World Translation Scenario: Simulation using Haitian Creole Emergency SMS Messages. In *Proceedings of EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, 399-404.
 17. Kronrad, Y., Hu, C., Buzek, O., & Quinn, A.J. (2011) Using Monolingual Crowd to Improve Translation. Student poster at AAAS 2011. **Winner of poster in Math, Technology and Engineering category.**
 18. **Bederson, B. B.**, & Quinn, A.J. (2011) Web Workers Unite! Addressing Challenges of Online Laborers. Alt.CHI paper at CHI 2011, ACM Press, 97-106.
 19. Hu, C., **Bederson, B. B.**, & Resnik, P. (2011) MonoTrans2: A New Human Computation System to Support Monolingual Translation. Short paper at CHI 2011, ACM Press, 1133-1136.
 20. Kronrod, Y., Buzek, O., Hu, C., Quinn, A., Resnik, P., & **Bederson, B. B.*** (2010) Using Crowdsourcing to Translate Without Bilinguals. Poster at CrowdConf 2010.
 21. Hu, C., **Bederson, B. B.**, and Resnik, P. (2010) Translation by Iterative Collaboration Between Monolingual Users. Abstract and demo at the Human Computation Workshop (HCOMP 2010), 54-55.
 22. Buzek, O., Resnik, P., & **Bederson, B. B.*** (2010) Error Driven Paraphrase Annotation using Mechanical Turk, In *Creating Speech and Language Data With Amazon's Mechanical Turk*, NAACL 2010 Workshop, 2010.
 23. Chen, R., Rose, A., & **Bederson, B. B.*** (2009) How People Read Books Online: Mining and Visualizing Web Logs for Use Information. In *Proceedings of European Conference on Digital Libraries (ECDL 2009)*, Short Paper, 364-369.
 24. **Bederson, B. B.**, Rutledge, P., Quinn, A. (2009) Generalizing the International Children's Digital Library. In *Proceedings of the Digital Humanities Conference (DH 09)*, Poster.
 25. **Bederson, B. B.**, Quinn, A., Druin, A. (2009) Designing the Reading Experience for Scanned Multi-lingual Picture Books on Mobile Phones. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2009)*, Short Paper, ACM Press, New York, NY, 305-308.
 26. Hu, C., Rose, A., **Bederson, B. B.*** (2009) Locating Text in Scanned Books. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2009)*, Poster.
 27. Druin, A., Cavallo, D., Fabian, C., **Bederson, B. B.**, Revelle, G., Rogers, Y. (2009) Mobile Technologies for the World's Children. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2009)*, ACM Press, Panel.
 28. **Bederson, B. B.** (2008) "Experiencing the International Children's Digital Library", *Interactions Magazine*, ACM Press, 15 (6), 50-54.
 29. Lazar, J., Hochheiser, H., Johnson, J., Karat, C-M., **Bederson, B. B.** (2008) "CHI Policy Issues Around the World", Panel in CHI 2008, ACM Press, 2277-2280.

-
30. Teevan, J., Jones, W., **Bederson, B. B.**, (2006) "Introduction: Personal Information Management," Communications of the ACM, 49 (1), 40-43. Ranked #3 "Year Rank" in annual downloads among ACM Magazines and Computing Surveys articles (CACM, 50 (12), 92-93).
 31. Lee, B., Parr, C.S., Plaisant, C., **Bederson, B. B.**, (2006) "TreePlus: Visualizing Graphs as Trees," Video in InfoVis 2006.
 32. Herrnson, P.S., Niemi, R.G., Hanmer, M.J., **Bederson, B. B.**, Conrad, F.G., Traugott, M., (2006) "The Importance of Usability Testing of Voting Systems," in Usenix/Accurate Electronic Voting Workshop.
 33. Herrnson, P., Niemi, R., Hanmer, M., **Bederson, B. B.**, Conrad, F., Traugott, M., (2006) "Voter Errors in Electronic voting: Voting Systems, Ballot Type, and Voter Traits," In Midwest Political Science Association annual conference.
 34. Herrnson, P., Niemi, R., Hanmer, M., **Bederson, B. B.**, Conrad, F., Traugott, M., (2006) "Voters' Abilities to Cast Write-In Votes Using Electronic Voting Systems," In Midwest Political Science Association annual conference.
 35. Conrad, F., Peytcheva, E., Traugott, M., Lewis, B., Herrnson, P., **Bederson, B. B.** (2006) "Usability of Electronic Voting Systems: Field and Laboratory Experiments," In Midwest Political Science Association annual conference.
 36. Lee, B., Parr, C. S., Plaisant, C., & **Bederson, B. B.*** (2005) "Visualizing Graphs as Trees: Plant a Seed and Watch it Grow", Proceedings of Graph Drawing 2005, 516-518.
 37. Klein, C. & **Bederson, B. B.*** (2005). "Benefits of Animated Scrolling", Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2005) ACM Press, Short Paper, 1965-1968.
 38. Lee, B., Czerwinski, M., Robertson, G. G. & **Bederson, B. B.** (2005). "Understanding Research Trends in Conferences using PaperLens", Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2005) ACM Press, Short Paper, 1969-1972.
 39. Traugott, M.W., Hanmer, M.J., Park, W.H., Herrnson, P.S., Niemi, R.G., **Bederson, B. B.**, Conrad, F.G., (2005) "The Impact of Voting Systems on Residual Votes, Incomplete Ballots, and Other Measures of Voting Behavior," In American Political Science Association meeting (APSA 2005).
 40. Lee, B., Czerwinski, M., Robertson, G. G. & **Bederson, B. B.** (2004). "Understanding Eight Years of InfoVis Conferences using PaperLens", InfoVis 2004 Symposium Compendium. **Winner of InfoVis Contest.**
 41. **Bederson, B. B.** (2004) Interfaces for Staying in the Flow. In Ubiquity, ACM Press, (5) 27.
 42. Hourcade, J. P., **Bederson, B. B.**, Druin, A. (2004) Preschool Children's Use of Mouse Buttons. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2004) ACM Press, Short Paper, 1411-1412.
 43. Selker, T., Fischer, E. A., **Bederson, B. B.**, McCormack, C., Nass, C. (2003) Voting: User Experience, Technology and Practice. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2003) ACM Press, Panel, 700-701.
 44. **Bederson, B. B.**, Clamage, A., Czerwinski, M. P., Robertson, G. G. (2003) A Fisheye Calendar Interface for PDAs: Providing Overviews in Small Displays. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2003) ACM Press, Demonstration, 618-619.
 45. Hourcade, J.P., **Bederson, B. B.**, Druin, A., Rose, A., Farber, A., Takayama, Y. (2002). The International Children's Digital Library: Viewing Digital Books Online. Interaction Design and Children International Workshop, August 28-29, 2002, Eindhoven, The Netherlands 15 (3), 151-167.
 46. Herrnson, P., **Bederson, B. B.** (2002) Debugging Maryland Balloting, Washington Post, Outlook Section, p. B08, Sunday May 12th, 2002.
 47. Good, L., **Bederson, B. B.**, Stefik, M., Baudisch, P. (2002) Automatic Text Reduction For Changing Size Constraints. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2002) ACM Press, Short Paper, 798-799.

-
48. Hourcade, J.P., **Bederson, B. B.**, Druin, A., Taxén, G. (2002) KidPad: A Collaborative Storytelling Tool for Children. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2002) ACM Press, Referred Demo, 500-501.
 49. Hourcade, J.P., **Bederson, B. B.**, Druin, A. (2002) SearchKids: A Digital Library Interface for Young Children. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2002) ACM Press, Referred Demo, 512-513.
 50. **Bederson, B. B.** (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. UIST 2001, ACM Symposium on User Interface Software and Technology. Refereed demo.
 51. Browne, H., **Bederson, B. B.**, Druin, A., Sherman, L., & Westerman, W. (September 2000) Designing a Collaborative Finger Painting Application for Children, HCIL-2000-17, CS-TR-4184, UMIACS-TR-2000-66
 52. Hourcade, J., **Bederson, B. B.** (May 1999) Architecture and Implementation of a Java Package for Multiple Input Devices (MID) HCIL-99-08, CS-TR-4018, UMIACS-TR-99-26
 53. Hightower, R.R., Ring, L.T., Helfman, J.I., **Bederson, B. B.**, Hollan, J.D. (1998) PadPrints: Graphical Multiscale Web Histories. UIST 1998, ACM Symposium on User Interface Software and Technology. Tech Note, 121-122.
 54. Stewart, J., Raybourn, E., **Bederson, B. B.**, & Druin, A. (1998). When Two Hands Are Better Than One: Enhancing Collaboration Using Single Display Groupware. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 98) ACM Press, Short Paper, 287-288.
 55. **Bederson, B. B.**, Hollan, J. D., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local Tools: An Alternative to Tool Palettes. In Proceedings of User Interface and Software Technology (UIST 96) ACM Press, Short Paper, 169-170.
 56. **Bederson, B. B.**, Hollan, J. D. (1995) Pad++: A Zoomable Graphical Interface System. In Proceedings of Conference on Human Factors in Computing Systems Conference Companion (CHI 95) ACM Press, Short Paper, 23-24.
 57. **Bederson, B. B.**, & Davenport, G., & Druin, A., & Maes, P. (1994). Immersive Environments: A Physical Approach to the Multimedia Experience, In Proceedings of IEEE International Conference on Multimedia Computing and Systems, Boston, Massachusetts, May 17-19, 1994, (Panel).
 58. **Bederson, B. B.** (1995). Audio Augmented Reality: A Prototype Automated Tour Guide. In Proceedings of Human Factors in Computing Systems (CHI 95) New York: ACM Press, (Short Paper), 210-211.
 59. Stone, L. M., Erickson, T., **Bederson, B. B.**, Rothman, P., Muzzy, R. (1994) Visualizing Data: Is Virtual Reality the Key? IEEE Visualization, IEEE Press, Short Paper, 410-413.
 60. **Bederson, B. B.**, Stead, L., & Hollan, J. D. (1994). Pad++: Advances in Multiscale Interfaces. In Proceedings of Human Factors in Computing Systems (CHI 94) New York: ACM Press, Short Paper, 315-316.
 61. **Bederson, B. B.** (1992). A Miniature Space-Variant Active Vision System: Cortex-I. Doctoral dissertation, New York University, New York, NY.

E. TALKS, ABSTRACTS, AND OTHER PROFESSIONAL PAPERS PRESENTED

i. Keynote Talks

1. April 2017 **CHIuXiD - The International Conference in Human-Computer Interaction & User Experience**, Jakarta, Indonesia
“From Human-Computer Interaction to Strategic Leadership for Change”
2. March 2017 **Maryland College Learning Center Association**, Shady Grove, MD
“Fishing for Faculty: Teaching & Learning Engagement at UMD”
3. April 2014 **Innovations in Teaching and Learning Conference**, College Park, MD
“An Introduction to the Teaching and Learning Transformation Center”

-
4. October, 2012 **South African Institute for Computer Scientists and Information Technologists**
Johannesburg, South Africa
“Social Participation: How Collective Activity Can Make Change”
 5. September, 2010 **Visual Information Communication International Symposium**, Beijing, China
“The Promise of Zoomable User Interfaces”
 6. May, 2010 **Human-Computer Interaction Lab’s 27th Annual Symposium**, College Park, MD
“The Promise of Zoomable User Interfaces”
 7. November, 2009 Raja Roy Singh Keynote Lecture at the **UNESCO Asia-Pacific Programme of Educational Innovation for Development**, Hangzhou, China
“Educational Technology for Creativity, Collaboration and Community”
 8. February, 2009 **Microsoft Internal “UX Canvas Camp”**, Redmond, WA
“A Short History of Zoomable User Interfaces”
 9. September, 2002 **IEEE Symposium on Human Centric Computing Languages and Environments (HCC 02)**, Arlington, VA
“Interfaces for Staying in the Flow”

ii. Invited Talks (doesn’t include Keynote talks, which are listed above)

1. October 3, 2018, **Rutgers University**, Newark, NJ
“Transforming Teaching at the University of Maryland”
2. September 28, 2018, **EPAM Executive Roundtable**, Minsk, Belarus
“Workforce of the Future”
on Panel with Eugene Roman (Canada Tire), Sjoerd Gehring (Johnson & Johnson) & Sandra Loughlin (UMD)
3. September 27, 2018, **EPAM Software Engineering Conference (SEC 2018)**, Minsk, Belarus
“Open Company, Open Community, Open World”
on Panel with Elaina Shekhter (EPAM), Sjoerd Gehring (Johnson & Johnson), Scott Nissenbaum (Ben Franklin Technology Partners) & Sandra Loughlin (UMD)
4. December 4, 2017, **Cornell Tech**, New York, NY
“Academic Technologies in Higher Ed”
5. November 10, 2017, **Online Programs and MOOCs – Opportunities and Challenges**, in The 7th Annual MIS Academic Leadership Conference, RHTSmith School of Business, University of Maryland
6. Sept 2, 2017, **Google**, New York, NY
“Academic Technologies in Higher Ed”
7. June 11, 2015 **Designing a Learning Environment**, in “Open Textbooks: Challenges for Policy” workshop, University of Haifa, Haifa, Israel
8. May 13, 2015 **Strategic Integration of MOOCs into Graduate and Professional STEM Programs**, Champaign, Illinois
Raineri, D., Breslow, L., **Bederson, B. B.**
“Future of MOOCs in Research Universities”
9. April 13, 2015 **#EduAnalyticsDC Workshop**, Washington, DC
“Understanding Consistency in a Large Multi-Section Course with Many Instructors”
10. March 3, 2015 **Coursera Partners Conference**, Newport Beach, CA
Verstelle, M., Bennett, A., **Bederson, B. B.**, & Jeffries, Pam
“Designing a MOOC Portfolio: Content Strategy for Online Education”
11. March 1, 2015 **Coursera Partners Conference**, Newport Beach, CA
Buitendijk, S., **Bederson, B. B.**, Kerrison, M., & Price, V.
“Value to Partners: Case Studies From Partner Institutions”

-
12. January 29, 2015 **University of Wisconsin, Madison**, Madison, WI
“Reflections on 20 Years Doing HCI Systems Research”
Department of Computer Science, Distinguished Lecture Series
 13. Sheil, M., Rock, E., O’Brien, L., **Bederson, B. B.**, & Xiaoming, L. (2014) “Future Trajectories of MOOCs in Higher Ed”, Panel at Coursera Partners’ Conference.
 14. February 12, 2014 **Patuxent Partnership**, Lexington Park, MD
“UMD MOOCs and What the Future Holds For Them”
 15. January 29, 2014 **Carnegie Mellon University**, Pittsburgh, PA
“Power to the People: Approaches to Crowdsourcing”
 16. January 14, 2014 **Microsoft**, Friendship Heights, MD
“Cracking the Code: Improving Computer Science Education Outcomes in the State of Maryland”
Panel participant on Microsoft public series “Conversations on Education”
 17. November 20, 2013 **University of Maryland**, College Park, MD
“Mobile, Online, Flipped, Active: New Uses of Educational Technologies”
College of Education
 18. October 11, 2013 **Huawei**, Hong Kong, China
“When Crowds Come Together”
 19. October 8, 2013 **National Institute of Information and Communications Technology**, Osaka, Japan
“Crowdsourced Monolingual Translation”
in 2nd Workshop on Future Directions in Translation Research
 20. February 1, 2013 **Google**, Mountain View, CA
“Social + MOOCs + Hcomp = SMOOCH: Computing Comes to Education”
 21. January 29, 2013 **Google**, Mountain View, CA
“The Promise of Zoomable User Interfaces: Reflections on Design of Simple UX”
 22. December 6, 2012 **Sunlight Foundation**, Washington, DC
“Digital Disclosure of a Nationally Standardized Database of Restaurant Food Safety Inspections” (with Ginger Jin)
 23. November 19, 2012 **University of Maryland**, College Park, MD
“Innovation, Iteration & Creation: A Discussion about the Development of New Ideas in the Modern World”
 24. November 6, 2012 **World Usability Day**, Minsk, Belarus
“The Promise of Zoomable User Interfaces”
 25. November 2, 2012 **Central & Eastern European Software Engineering Conference in Russia** Moscow, Russia
“Learning to Code in an Online World”
 26. November 2, 2012 **Central & Eastern European Software Engineering Conference in Russia** Moscow, Russia
Panelist in “Peopleware: what skills and qualifications we need to build a dream team of software engineers”
 27. October 4, 2012 **UNISA – University of South Africa**, Johannesburg, South Africa
“Innovation, Iteration & Creation: A Discussion about the Development of New Ideas in the Modern World”
 28. June 20, 2012 **University of Haifa**, Haifa, Israel
“Books that are Blue: What Children Can Teach Us About Creating New Technologies” (with Allison Druin)
 29. June 19, 2012 **Interdisciplinary Center**, Herzilya, Israel
“Books that are Blue: What Children Can Teach Us About Creating New Technologies” (with Allison Druin)

-
30. Nov 17, 2011 **Future of Information Alliance**, College Park, MD
“Anonymity and Privacy While Learning”
 31. Sept 21, 2011 **U.S. Senate and Congress Briefings**, Washington, DC
Panel participant in “Deconstructing the iPad: How Federally Supported Research Leads to Game-Changing Innovation” run by the Task Force on American Innovation and the CRA
 32. Aug 24, 2011 **Summer Social Webshop** College Park, MD
“Translation by Collaboration Among Monolingual Users”
 33. May, 2011 **National Science Foundation**, Arlington, VA
Distinguished Lecture Series
“Translation by Collaboration among Monolingual Users” (with Philip Resnik)
 34. September, 2010 **Microsoft Research Asia**, Beijing, China
“Collaborative Monolingual Translation”
 35. September, 2010 **Tsinghua University**, Beijing, China
“Collaborative Monolingual Translation”
 36. April, 2010 **CHI 2010**, Atlanta, GA
Social Impact Award talk
 37. October, 2009 **Google**, Mountain View, CA
Panel participant in “Breakthrough Learning in a Digital Age” invitation only conference
 38. September, 2009 **Google Tech Talk**, Mountain View, CA
“Translation by Iterative Collaboration between Monolingual Users”
 39. April, 2009 **University of Illinois (CS Dept.)**, Champaign-Urbana, IL
“Zoomable User Interfaces from Sun Sparcstations to iPhones”
 40. April, 2009 **University of Illinois (iSchool)**, Champaign-Urbana, IL
“From Ulaan Baatar to iPhone: A Digital Library for the World’s Children”
 41. February, 2009 **Tufts University**, Medford, MA
“From iPhone to Ulaan Baatar: A Digital Library for the World’s Children”
 42. February, 2009 **Microsoft Internal “UX Canvas Camp”**, Redmond, WA
“A Short History of Zoomable User Interfaces”
 43. September, 2008, **Carnegie Mellon University**, Pittsburgh, PA
“Black Ears to Blonde Cats: Paths for Designing Change”
 44. May, 2008, **Manchester Public Library** Manchester by the Sea, MA
“No Road, Drive: The ICDL Travels to Mongolia”
 45. April, 2008 **Web 2.0 Expo**, San Francisco, CA
“Next Generation Mobile UI” (with John SanGiovanni)
 46. February, 2008 **USACM Executive Council**, Washington, DC
“Voting Technology: The Not-So-Simple Act of Casting a Ballot”
 47. October, 2007 **Widget Summit**, San Francisco, CA
“Mobile Widget Design”
 48. September, 2007 **Knight Center for Specialized Journalism**, College Park, MD
“Mobile Technology Trends”
 49. June, 2007 **Mobile Marketing Association**, New York, NY
“Designing for Usability”
 50. May, 2007 **MIT Enterprise Forum**, Arlington, VA
“Co-present Mobile Collaboration”
 51. December, 2006 **Human Factors and Ergonomics Society Potomac Chapter**, Bethesda, MD
“A Library for the World’s Children”

-
52. June, 2006 **Open Society Forum**, Ulaan Baatar, Mongolia
"A Library for the World's Children"
 53. February, 2006 **Dingman Center for Entrepreneurship, R.H. Smith School of Business, UMD**
"Web 2.0 UI"
 54. December, 2005 **World Bank**, Washington, DC
"The International Children's Digital Library"
 55. December, 2005 **IBM T.J. Watson Research Center**, Hawthorne, NY
"Making Visualization Work"
 56. November, 2005 **NSA**, Laurel, MD
"Making Visualization Work"
 57. October, 2005 **IBM Research**, Cambridge, MA
"Making Visualization Work"
 58. October, 2005 **Johns Hopkins University – Computer Science Dept.**, Baltimore, MD
"Making Visualization Work"
 59. September, 2005 **InfoToday Conference on 'Innovations in Search'**, New York, NY
"Zooming Into Visual Search"
 60. March, 2005 **NFAIS**, Philadelphia, PA
"Making Sense of Search Results"
 61. February, 2005 **American Association for the Advancement of Science (AAAS)** Washington, DC
Panel in annual conference,
"Designing Interfaces for Voting Machines"
 62. February, 2005 **National Association of State Election Directors (NASED)** Washington, DC
Invited talk at annual conference.
"Designing Interfaces for Voting Machines"
 63. January, 2005 **Microsoft**, Seattle, WA
"Interfaces for Staying in the Flow"
 64. January, 2005 **Google**, Mountain View, CA
"The International Children's Digital Library"
 65. December, 2004 **National Library of New Zealand**, Wellington, New Zealand
"The International Children's Digital Library"
 66. December, 2004 **University of Waikato**, Hamilton, New Zealand
"Interfaces for Staying in the Flow"
 67. December, 2004 **Victoria University Wellington**, Wellington, New Zealand
"Interfaces for Staying in the Flow"
 68. December, 2004 **University of Canterbury**, Christchurch, New Zealand
"Interfaces for Staying in the Flow"
 69. November, 2004 **National Archives and Records Administration (NARA)**, College Park, MD
in Conference "Partnerships in Innovation: Serving a Networked Nation"
"International Children's Digital Library"
 70. April, 2004 **Johns Hopkins University Applied Physics Lab**, Laurel, MD
"Interfaces for Staying in the Flow"
 71. October, 2003 **University of Maryland, Robert H. Smith School of Business**, College Park, MD
on InForum 2003 panel on "Does IT Matter?"
 72. September, 2003 **University of Maryland**, College Park, MD
"Making Visualization Work"
 73. September, 2003 **University of Victoria**, Victoria, Canada
"Interfaces for Staying in the Flow"

-
74. March, 2003 **National Archives and Records Administration**, College Park, MD
"Visualizations for Archivists"
 75. March, 2003 **Microsoft Research**, Redmond, WA
"DateLens: A Scalable Fisheye Calendar Interface"
 76. January, 2003 **Sun Microsystems**, Menlo Park, CA
"Zoomable User Interfaces for Searching and Browsing"
 77. September, 2002 **Microsoft Research Limited**, Cambridge, England
"Information Visualization for Mobile Devices"
 78. July, 2002 **Microsoft Research**, Redmond, WA
Demo of DateLens with Bill Gates at Microsoft Faculty Summit
 79. July, 2002 **Microsoft Research**, Redmond, WA
"Scalable Interfaces for Mobile Devices"
 80. June, 2002, **WebShop**, Dept. of Sociology, UMD, College Park, MD
"Access and Usability: The Importance of Internet Applications"
 81. January, 2002 **Workshop on Election Standards and Technology**, Washington, DC
"Visualization Techniques for Electronic Voting"
 82. October, 2001 **Microsoft Research**, Redmond, WA
"Zoomable User Interfaces for Searching and Browsing: SearchKids & PhotoMesa"
 83. October, 2001 **Xerox PARC**, Palo Alto, CA
"Zoomable User Interfaces for Searching and Browsing: SearchKids & PhotoMesa"
 84. November, 2000 **Census Bureau, Dept. of Commerce**, Suitland, MD
"Zoomable User Interfaces"
 85. May, 2000 **Bowie State University**, Bowie, MD
"Zoomable User Interfaces and Single Display Groupware"
 86. April, 2000 **University of Delaware**, DE
"Zoomable User Interfaces and Single Display Groupware"
 87. August, 1999 **Xerox PARC**, Palo Alto, CA
"Zoomable User Interfaces and Single Display Groupware"
 88. August, 1999 **University of California, Berkeley**, Berkeley, CA
"Zoomable User Interfaces and Single Display Groupware"
 89. August, 1999 **Sun Microsystems, Inc. (Java Group)**, Palo Alto, CA
"Zoomable User Interfaces and Single Display Groupware"
 90. July, 1999 **Nokia**, Linköping Sweden
"Zoomable Information Retrieval Systems and Single Display Groupware"
 91. June, 1998 **IBM**, Yorktown, NY
"Zoomable User Interfaces"
 92. May, 1998 **Washington DC Linux User's Group**, Alexandria, VA
"Zoomable User Interfaces"
 93. April, 1998 **National Institute of Standards and Technology (NIST)**, Gaithersburg, MD
"Zoomable User Interfaces"
 94. March, 1998 **Microsoft Research**, Redmond, WA
"Zoomable User Interfaces"
 95. October, 1997 **Netherlands Design Institute**, Amsterdam, Holland
"Pad++: Zoomable User Interfaces"
 96. October, 1997 **Philips Labs**, Eindhoven, Holland
"Pad++: Zoomable User Interfaces"

-
97. October, 1997 **CID, Royal Institute of Technology (KTY)**, Stockholm, Sweden
"Pad++: Zoomable User Interfaces"
 98. September, 1997 **Swedish Institute of Computer Science (SICS)**, Stockholm, Sweden
"Pad++: Zoomable User Interfaces"
 99. October, 1996 **ART3000, A French society for the development of Art & Technology**, Paris, France
"Pad++: A Zoomable Graphical Interface"
 100. October, 1996 **University of Michigan, Department of Information Sciences**, Ann Arbor, MI
"Pad++: A Zoomable Graphical Interface"
 101. July, 1996 **AT&T Bell Labs**, NJ
"Pad++: A Zoomable Graphical Interface"
 102. January, 1996 **Stanford University, Cognitive Psychology Group**, CA
"Pad++: A Zoomable Graphical Interface"
 103. October, 1995 **Cornell University, Computer Science Department**, Ithaca, NY
"Pad++: A Zoomable Graphical Interface"
 104. June, 1995 **Sandia National Labs**, Los Alamos, NM
"Pad++: A Zoomable Graphical Interface"
 105. April, 1995 **Los Alamos National Labs**, Los Alamos, NM
"Pad++: A Zoomable Graphical Interface"
 106. November, 1993 **Digital Equipment Corporation, Stanford Research Center**, CA
"Pad++: A Multiscale Hierarchical Sketchpad"
 107. October, 1993 **Brown University, Computer Science Department**, Providence, RI
"Pad++: A Multiscale Hierarchical Sketchpad"
 108. April, 1993 **University of Chicago, Computer Science Department**, |
"PA3D: A Multiscale Hierarchical Sketchpad"
 109. October, 1992 **Rensselaer Polytechnic Institute, Computer Science Department**, Troy, NY
"Cortex-I: A Miniature Active Vision System"

H. ORIGINAL DESIGNS, PLANS, INVENTIONS, AND PATENTS

i. Patents

1. SanGiovanni, J., & **Bederson, B. B.** (Inventors). (2022) *Systems, methods, and computer program products displaying interactive elements on a canvas*. US Patent #11,243,673
2. **Bederson, B. B.**, & SanGiovanni, J. (Inventors). (2021) *Techniques to modify content and view content on a mobile device*. US Patent #11,061,524.
3. SanGiovanni, J., & **Bederson, B. B.**, and Painter, B. (Inventors). (2018) *Systems and methods for adaptive third party content*. US Patent #10,019,963.
4. SanGiovanni, J., & **Bederson, B. B.** (Inventors). (2018) *Systems, methods, and computer program products displaying interactive elements on a canvas*. US Patent #9,959,020.
5. **Bederson, B. B.**, & SanGiovanni, J. (Inventors). (2017) *Techniques to modify content and view content on mobile devices*. US Patent #9,778,810.
6. SanGiovanni, J., & **Bederson, B. B.** (Inventors). (2016) *Systems, Methods, and Computer Program Products Providing an Integrated User Interface for Reading Content*. US Patent #9,361,130.
7. SanGiovanni, J., & **Bederson, B. B.** (Inventors). (2014) *Systems, Methods, and Computer Program Products Displaying Interactive Elements on a Canvas*. US Patent #8,819,570.
8. Pahud, M., Murillo, O. E., Karlson, A. K., & **Bederson, B. B.** (Inventors). (2012). *Monitoring Pointer Trajectory and Modifying Display Interface*. US Patent #8,261,211.

-
9. Good, L.E., **Bederson, B. B.**, & Stefik, M.J. (Inventors). (2010). *Methods and Systems for Supporting Presentation Tools Using Zoomable User Interfaces*. US Patent #7,707,503.
 10. **Bederson, B. B.**, Good, L. E., & Stefik, M.J. (Inventors). (2009). *Methods and Systems for Incrementally Changing Text Representation*. US Patent #7,650,562 B2.
 11. **Bederson, B. B.**, Good, L. E., & Stefik, M. J. (Inventors). (2009). *Methods and Systems for Incrementally Changing Text Representation*. US Patent #7,549,114 B2.
 12. Wallace, R. S., **Bederson, B. B.**, & Schwartz, E. L. (Inventors). (1997). *TV Picture Compression and Expansion*. US Patent #5,642,167.
 13. **Bederson, B. B.**, Wallace, R. S., & Schwartz, E. L. (Inventors). (1993). *Two-Dimensional Pointing Motor*. US Patent #5,204,573.
 14. Wallace, R. S., **Bederson, B. B.**, & Schwartz, E. L. (Inventors). (1992). *Telephone Line Picture Transmission*. US Patent #5,175,617.

ii. University Disclosures

1. **Bederson, B. B.** & Lin, J. (Inventors) (2009), CrowdFlow: A Computer-Human Hybrid Cloud Computing Model.
2. **Bederson, B. B.** (Inventor) (2002), Fisheye Calendar: A Calendar interface for small to large devices, IS-2002-015.
3. **Bederson, B. B.** (Inventor) (2001), PhotoMesa: A Zoomable Image Browser, IS-2001-053
4. **Bederson, B. B.**, & Chipman, L. G. (Inventors) (2000), *Physical Linear Slider Device to Control Computer Interface*, IS-2000-018.
5. **Bederson, B. B.** (Inventor) *Fisheye Menus For Selecting an Item From a Long List*, IS-2000-025.
6. **Bederson, B. B.**, Druin, A., & Pablo, J. P. (Inventors) (2000), *KidPad: A Collaborative Storytelling Environment for Children*, IS-2000-019.

I. CONTRACTS AND GRANTS

Total Funding: \$18,036,144

UMD Portion: \$10,329,299

1. **Ithaka SR, July 2016, co-PI**
Elementary Statistics with an adaptive learning system
\$113,500
2. **Maryland Innovation Initiative, March 2014, PI**
A Nationwide Database of Food Safety Inspections for Retail Establishments
\$100,000
3. **Huawei, July 2013, PI**
XParty: Synchronous Online Language Learning
\$54,000
4. **Google, March 2013, PI**
XParty: Synchronous Peer Learning in MOOCs (extension)
\$48,872
5. **Google, December 2011, PI**
Search Party: Learning to Search in a Web-Based Classroom
\$60,000
6. **Alfred P. Sloan Foundation, November 2011, co-PI**
Digital Disclosure of a Nationally Standardized Database of Restaurant Food Safety Inspections
\$465,272

7. **Google, June 2011, PI**
Search Party: Web-Based Classroom Learning to Search
\$15,000
8. **Google, June 2011, co-PI**
Translate the World: A Unified Framework for Crowdsourcing Translation
\$150,000
9. **Nokia Research, July 2010, PI**
Mobile Learning
\$36,000
10. **NSF, CDI Type I: Translation as a Collaborative Process, Sep 2009, PI**
\$630,000
11. **Google, July 2009, PI**
Translation as a Collaborative Process
\$70,000
12. **Sesame Workshop, January 2009, co-PI**
Literacy 360 Alliance
\$22,500
13. **Eliassen Foundation, Nov 2008, co-PI**
ICDL Elias Project
\$50,000
14. **Intel, Nov 2008, co-PI**
ICDL ClassmatePC Project
\$50,000
15. **NSF, Designing and Understanding Intergenerational Mobile Learning Communities, Sep 2008, co-PI**
\$100,000
16. **Government of Mongolia Ministry of Education, Culture and Science – Nov2008, PI**
International Children’s Digital Library Phase 2
\$230,000
\$190,000 (for UMD)
17. **NSF, Usability Study of Independent Voter Verification Systems – July 2006, co-PI**
PI – Paul Herrnsen (Gov’t & Politics)
\$68,000
18. **Government of Mongolia Ministry of Education, Culture and Science – May 2006, PI**
Creating Digital Libraries in Mongolia
\$75,000
19. **IBM Faculty Award – December 2005, PI**
Award to support my research on the open source Piccolo graphics toolkit.
\$30,000
20. **Maryland State Board of Elections, September 2005 – January 2006, co-PI**
PI – Paul Herrnsen (Gov’t & Politics)
\$70,000
21. **(Agency not disclosed), April 2005 – March 2006, co-PI**
Joint Institute for Knowledge Discovery
\$224,000 (my part of larger \$2,000,000 effort led by VS Subrahmanian)
22. **Booz Allen (subcontract from ARDA), March 2005 – April 2006**
Human-Information Interaction Research
\$325,000

-
23. **Toshiba, July 2004 – December 2006, PI**
Automatic Thumbnail Cropping
\$20,000
 24. **Microsoft, January 2004 – December 2006, PI**
Interaction Design and Visualization
\$1,000,000
 25. **DARPA, February, 2003 – January 2006, PI**
Zoomable User Interfaces for the Semantic Web
\$230,000
 26. **NSF, September, 2003 – August 2006, co-PI**
PI – Paul Herrnson (Gov't & Politics), co-PI, Richard Niemi (Univ. of Rochester), Michael Traugott (Univ. of Michigan)
An Assessment of Voting Technology and Ballot Design
\$900,000 – approx \$540K for UMD research
 27. **Chevron, Oct 2002 – Oct 2003, co-PI**
PI – Catherine Plaisant (UMIACS)
Information Visualization for the Oilfield of the Future
\$75,000
 28. **IBM, September, 2002, PI**
T221 high-resolution display
\$9,000 in kind
 29. **IMLS, September, 2002 – August, 2005, co-PI**
PI – Allison Druin (CLIS)
The Children's International Digital Library
\$397,162
 30. **NSF, September, 2002 – August 2005, PI**
Search Interfaces for Biodiversity Informatics
\$380,000
 31. **NSF, September 2002 – August 2007, co-PI**
PI – Allison Druin (CLIS), co-PI, Anne Weeks
Developing a Children's International Digital Library
\$3,000,000 (\$2,125,000 to UMD)
 32. **NSF, September 2002 – August 2005, co-PI**
co-PI – Allison Druin (CLIS), PI - Nancy Kaplan (Univ. of Baltimore), Stuart Malthrop (U. of Balt.)
An HCI Partnership Serving Under represented Groups
\$420,000 (\$133,000 to UMD)
 33. **Microsoft Research, May 2002 – April 2001, PI**
Mobile Interfaces and .NET
\$100,000
 34. **Microsoft Research, September 2001 – August 2002, PI**
Mobile Interfaces
\$50,000
 35. **DARPA, May 2001 – August 2003, PI**
Zoomable User Interfaces for the Semantic Web
\$320,000
 36. **European Union – Esprit, The Disappearing Computer, Jan 2001 – December 2003, co-PI**
co-PI – Allison Druin (CLIS), Several PIs in Europe
With Royal Institute of Technology (KTH – Sweden), Univ. of Paris Sud, France.
InterLiving: Designing Interactive, Intergenerational Interfaces for Living Together
~\$1,500,000 - approx \$400K for UMD research

-
37. **Chevron, Nov 2000 – May, 2002, PI**
co-PI – Catherine Plaisant (UMIACS)
Information Visualization
\$141,000
 38. **DARPA, May 1998 – June 2002, PI**
Zoomable User Interfaces for InImEx and Command Post of the Future (CPOF)
\$1,000,000
 39. **Lockheed Martin Corp. sub-contract from DARPA, June 1999 - June 2000, PI**
Unified Zoomable Environment (UZE) for Command Post of the Future
\$40,000
 40. **European Union – Esprit, Intelligent Information Interfaces, Sept. 1998 – August 2001, co-PI**
co-PI – Allison Druin (CLIS), Several PIs in Europe
With Swedish Institute of Computer Science, Royal Institute of Technology (KTH – Sweden)
University of Nottingham, England.
KidStory
~\$1,500,000 - approx \$300K for UMD research
 41. **Sun Microsystems, May 1999, PI**
Zoomable User Interfaces
\$30,000 in equipment
 42. **UNM sub-contract from DARPA, February 1998 – March 1999, PI**
Pad++: Zoomable User Interfaces
\$200,000
 43. **Microsoft, April, 1998, PI**
Productivity and development software
\$5,000 in software and tech support
 44. **Trivista, PI January 1998 - January 1999, PI**
Graphical User Interfaces based on Zooming
\$50,000

UNIVERSITY OF NEW MEXICO

45. **Sony Research Corporation, PI**
co-PI – Jim Hollan (Univ. of New Mexico), co-PI – Ken Perlin (New York University)
Licensing of Pad++ software from UNM and NYU
\$500,000 plus royalties
46. **Intel Donation, January, 1997, co-PI**
PI – Jim Hollan (Univ. of New Mexico)
200 MHz PC Computer
\$4,000 in-kind
47. **DARPA, July'94 - June'97, Worked on grant with Jim Hollan**
Beyond Imitation: A Strategy For Building A New Generation of HCI Design Environments
\$3,000,000 over 3 years with subcontracts to New York University and the Univ. of Michigan
48. **Sandia LDRD (Sandia internal research project) Oct'95 - Oct'96, PI**
Data Zooming - A New Physics for Information Navigation
\$64,000
49. **Sandia SURP (Sandia University Research Program) Oct'95 - Oct'96, PI**
Pad++: A Zoomable Interface to the World-Wide Web
\$40,000
50. **Sony Research Corporation, Jan'96, PI**
co-PI – Allison Druin (University of New Mexico)
Help plan Sony's efforts to develop a Zooming User Interface system
\$3,674

-
51. **Sandia LDRD (Sandia internal research project) Oct'94 - Oct'95, Worked on grant with Jim Hollan**
\$40,000
 52. **Bellcore Donation, July'93, PI**
SGI Indigo Workstation computer
\$30,000 in-kind

J. FELLOWSHIPS, PRIZES, AWARDS

1. **2022 – IEEE VIS Test of Time Award.** Joint with Catherine Plaisant and Jesse Grosjean – for my 2002 paper: SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 2002, pp. 57-64, doi: 10.1109/INFVIS.2002.1173148. <https://ieevis.org/year/2022/info/awards/test-of-time-awards>
2. **2021 – IEEE VIS Test of Time Award.** Joint with Ben Shneiderman and Martin Wattenberg – for my 2001 and 2002 papers: Ordered and Quantum Treemaps: Making effective use of 2D space to display hierarchies. Benjamin B. Bederson, Ben Shneiderman, Martin Wattenberg. *ACM TOG* 21(4):833-854 200
DOI: 10.1145/571647.571649. <https://ieevis.org/year/2021/info/awards/test-of-time-awards>
3. **2016 – Winner of Best Historical Paper Award.** Joint with Pekka Parhi and Amy K. Karlson
MobileHCI 2016 – 18th International Conference on Human-Computer Interaction With Mobile Devices and Services for “Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices”, MobileHCI 2006.
4. **2012 – Elected to the CHI Academy**
<http://www.sigchi.org/about/awards/>

The CHI Academy is an honorary group of individuals who have made substantial contributions to the field of human-computer interaction. These are the principal leaders of the field, whose efforts have shaped the disciplines and/or industry, and led the research and/or innovation in human-computer interaction. The criteria for election to the CHI Academy are:

- Cumulative contributions to the field.
- Impact on the field through development of new research directions and/or innovations.
- Influence on the work of others.

5. **2011 – Recognized as an ACM Distinguished Scientist**
http://awards.acm.org/distinguished_member/year.cfm

The Distinguished Member Grade recognizes those ACM members with at least 15 years of professional experience and 5 years of continuous Professional Membership who have achieved significant accomplishments or have made a significant impact on the computing field.

6. **2010 – ICDL received the American Library Association President’s Award for International Library Innovation.** Joint with Allison Druin and Ann Weeks.
7. **2010 – Winner of SIGCHI Social Impact Award**
<http://www.sigchi.org/about/awards/2010-sigchi-awards>
8. **2009 – Winner of Digital Education Achievement Award from the Center for Digital Education and Converge magazine**
For joint work on the International Children’s Digital Library with Allison Druin and Ann Weeks.
9. **2007 – Winner of Brian Shackel Award for “outstanding contribution with international impact in the field of HCI”**
Karlson, A., & Bederson, B. B. (2007) ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices, *Proceedings of INTERACT 2007*, 324-338.
10. **2005 – IBM Faculty Award**
\$30,000 gift to support my work on open source Piccolo graphics toolkit

-
11. **2005 – The International Children’s Digital Library** (www.icdlbooks.org) website was selected for inclusion by the National Endowment for the Humanities on their EDSITEment website (<http://edsitement.neh.gov>) as one of the best online resources for education in the humanities.
 12. **2004 – Winner of InfoVis Contest**
Lee, B., Czerwinski, M., Robertson, G. G. &, Bederson, B. B. (2004). “Understanding Eight Years of InfoVis Conferences using PaperLens”, InfoVis 2004, IEEE, 216-217.
 13. **2003 – Winner of Best Student Paper Award**
Suh, B., Ling, H., Bederson, B. B., & Jacobs, J. W. (2003). “Automatic Thumbnail Cropping and Its Effectiveness”. UIST 2003, ACM Symposium on User Interface Software and Technology, CHI Letters, 5(2), 95-104.
 14. **1992 – Janet Fabri Memorial Award for Outstanding Doctoral Dissertation**,
Department of Computer Science, Courant Institute, New York University

K. REVIEWING ACTIVITIES

ACM Conference on Graphics (SIGGRAPH)
ACM Conference on Human Factors in Computing Systems (CHI)
ACM Conference on Information Retrieval (IR)
ACM Conference on User Interfaces and Software Technology (UIST)
ACM Transactions on Computer-Human Interaction (TOCHI)
Communications of the ACM (CACM)
Computer Graphics and Applications
Graphics Interface Conference, Canada (GI)
International Conference on Information Visualization
International Journal of Human-Computer Interaction
IEEE Symposium on Information Visualization (InfoVis)
IEEE Symposium on Visual Analytics Science and Technology (VAST)
IEEE Transactions on Software Engineering (TSE)
IEEE Transactions on Visualization and Computer Graphics (TVCG)
Journal of Experimental Algorithmics
Journal of Information Visualization
Journal of Software Practice and Engineering (SPE)
NSERC (Canadian scientific funding agency)
National Science Foundation

L. OTHER

i. Software

StoryKit	iPhone app for children and their elders to create new books. Put in iPhone App Store August, 2009. Typical usage of 100,000 hours of story writing per month.
ICDL for iPhone	iPhone app for reading books from the International Children's Digital Library. Put in iPhone App Store November, 2008.
International Children's Digital Library	A library of children's books. Put on the Web Nov, 2002. http://www.childrensbooks.org <ul style="list-style-type: none">• Unique accessors total: > 7,000,000• Unique accessors monthly: ~130,000
DateLens	A calendar program for PDAs. Put on the Web July, 2002
SpaceTree	A general purpose tree browser. Put on the Web Dec, 2002. http://www.cs.umd.edu/hcil/spacetree
PhotoMesa	An end-user zoomable photo browser. Put on the Web May, 2001. http://www.cs.umd.edu/hcil/photomesa
CounterPoint	A zoomable presentation tool that works as a plug-in for Microsoft PowerPoint. Put on the Web in late 2001. http://www.cs.umd.edu/hcil/counterpoint
Piccolo	A library to support structured 2D graphics, the successor to jazz. Put on the Web in July 2002. http://www.cs.umd.edu/hcil/piccolo
Jazz	A general-purpose Java toolkit for building Zoomable User Interfaces. Put on the Web as Open Source software July 1999. http://www.cs.umd.edu/hcil/jazz
KidPad	An application that teaches young children communication skills through collaborative storytelling. Built using MID and KidPad. Put on the Web for non-commercial use March 2000. Access through March 2001 follow. http://www.cs.umd.edu/hcil/kidpad
Fisheye Menus	A new user interface widget for selecting one item for a long list using graphical distortion. Put on the Web for non-commercial use June 2000. Access through March 2001 follow. http://www.cs.umd.edu/hcil/fisheyemenu
MID	A general-purpose Java toolkit for supporting multiple input devices. It currently supports multiple USB mice on Windows 98. Put on the Web as Open Source software July 1999. It is used by the small but growing community of researchers investigating Single Display Groupware http://www.cs.umd.edu/hcil/mid
Pad++	Pad++ is described in detail in the papers referenced above. http://www.cs.umd.edu/hcil/pad++
Togl	An extension to Tcl/Tk that allows the creation of OpenGL graphics written in C to appear in a window managed by Tk. In addition, Tcl event bindings can be attached to the window which can then communicate with the C code that did the OpenGL rendering. Put on the net for public access. http://togl.sourceforge.net/

ii. Media

1. February 2, 2017 **The Diamondback** by Carly Taylor
"McKeldin Mall's Newest Academic Building Will Be Completed in March After Nearly 3 Years"
<http://www.dbknews.com/2017/02/02/edward-st-john-construction-umd/>

2. August 29, 2016 **The Diamondback** by Andrew Dunn
“University System of Maryland to offer an online course for free”
<http://www.dbknews.com/2016/08/30/mooc-umb-umuc-online-usm/>
3. February 15, 2016 **The Diamondback** by Samantha Reilly
“SGA passes resolution to revive 2012 syllabus bill”
<http://www.dbknews.com/2016/02/16/umc-sga-passes-syllabus-resolution-to-revive-university-senate-bill/>
4. September 24, 2015 **The Diamondback** by Taylor Swaak
“Active-Learning Courses Gain Traction at the University of Maryland”
http://www.diamondbackonline.com/news/active-learning-courses-gain-traction-at-the-university-of-maryland/article_435098ca-6247-11e5-a8e2-df59d97181a5.html
5. September 14, 2015 **Washington Post Express** by Beth Luberecki
“Massive open online courses (MOOCs) can help you achieve the next step in your career”
<https://www.washingtonpost.com/express/wp/2015/09/14/massive-open-online-courses-moocs-can-help-you-achieve-the-next-step-in-your-career/>
6. April 30, 2015 **The Diamondback** by Talia Richman
“USM Center for Academic Innovation to be renamed to honor Brit Kirwan”
http://www.diamondbackonline.com/news/article_51cbc02c-eeb7-11e4-a838-6b9857ba8e37.html
7. April 13, 2015 **The Diamondback** by Rokia Hassanein
“UMD Team Creates Nation’s Largest Food Safety Inspection Database”
http://www.diamondbackonline.com/news/article_553da506-e181-11e4-b6c9-0f23d720b7cc.html#.VS1VV_HbNqc.twitter
8. April 6, 2015 **Quality Assurance & Food Safety Magazine**
“Database of Food Safety Inspection Information Created”
<http://www.qualityassurancemag.com/food-safety-inspection-information.aspx>
9. February 23, 2015 **Diamondback, UMD Student Newspaper** by Taylor Swaak
“New UMD website offers alternative ways to save money on textbooks”
http://www.diamondbackonline.com/news/article_0d910a34-baf5-11e4-826a-dfbcb9e8b5f8.html
10. January 13, 2015 **WAMU Kojo Nnamdi Show**
“The Rise of Mobile Apps”
http://thekojoannamdishow.org/shows/2015-01-13/tt_the_rise_of_mobile_apps
11. October 1, 2014 **The Chronicle of Higher Education**
“Optimism About MOOCs Fades in Campus IT Offices”
<http://chronicle.com/blogs/wiredcampus/optimism-about-moocs-fades-in-campus-it-offices-survey-finds/54705>
12. September 8, 2014 **WAMU Kojo Nnamdi Show**
“Beefing Up Photo, Cloud and Online Security”
<http://thekojoannamdishow.org/shows/2014-09-09/beefing-photo-cloud-and-online-security>
13. April 29, 2014 **Diamondback, UMD Student Newspaper** by James Levin
“Learning to ask for help”
http://www.diamondbackonline.com/news/article_632cbbb8-cf5b-11e3-b8b4-0017a43b2370.html
14. April 18, 2014 **CNN.com** by Emanuella Grinberg
“Removing ‘barriers’ to education through free college textbooks”
<http://www.cnn.com/2014/04/18/living/open-textbooks-online-education-resources/>
15. January 27, 2014 **DiamondBack, UMD Student Newspaper** by Ellie Silverman
“University expanding online class sequences through Vanderbilt partnership”
http://www.diamondbackonline.com/news/campus/article_dff739f4-86ed-11e3-9a18-001a4bcf6878.html
16. September 27, 2013 **Baltimore Business Journal** by Sarah Gantz
“Maryland Schools use Online Model to Tout Brand, Attract Students”
<http://www.bizjournals.com/baltimore/print-edition/2013/09/27/maryland-schools-use-online-model-to.html?page=all>

17. July 16, 2013 **HuffingtonPost.com**
“Blended Learning: College Classrooms of the Future”
http://www.huffingtonpost.com/uloop/blended-learning-college-_b_3598718.html
18. December 5, 2012 **Diamondback, UMD Student Newspaper** by Allison Gray
“University members hoping to see ‘massive open online courses’ integrated”
http://www.diamondbackonline.com/news/article_4873330c-3ef1-11e2-9a16-0019bb30f31a.html
19. September 21, 2012 **SmartMoney.com** by Quentin Fottrell
In Apple’s Maps War vs. Google, Consumers Lose
http://blogs.smartmoney.com/advice/2012/09/21/in-apples-maps-war-vs-google-consumers-lose/?link=SM_hp_ls4e
20. September 18, 2012 **WAMU Kojo Nnamdi Show**
“Creativity and the Patent Wars” (guest on one-hour show)
<http://thekojonnamdishow.org/shows/2012-09-18/creativity-and-patent-wars>
21. September 7, 2012 **New York Magazine** by Matthew Shaer
“The Digital Elite: Smartphones are getting huge. Can the average-size thumb keep up?” (discusses my study on thumb size and mobile touch screens.
22. Aug 13, 2012 **Fact Witness in Apple v. Samsung trial covered broadly**
<http://blogs.wsj.com/digits/2012/08/13/samsungs-first-witnesses-push-prior-art-after-minor-victory/>
http://news.cnet.com/8301-13579_3-57492486-37/samsung-goes-on-offense-aims-to-bust-apple-patents/
<http://arstechnica.com/tech-policy/2012/08/apple-rests-patent-case-as-judge-tosses-three-samsung-phones-from-trial/>
<http://www.forbes.com/sites/connieguglielmo/2012/08/13/apple-says-it-complained-about-samsung-copying-in-2010-live-blog/>
<http://allthingsd.com/20120813/judge-refuses-to-toss-most-of-apples-suit-against-samsung/>
<http://www.theverge.com/2012/8/13/3240686/samsung-apple-patents-launchtile-diamonddtouch-table>
<http://www.eweek.com/c/a/Mobile-and-Wireless/Samsung-Begins-Presenting-Case-in-Apple-Patent-Infringement-Trial-162894/>
23. May 7, 2012 **WAMU Kojo Nnamdi Show**
“Free Online Courses - Higher Ed of the Future?” (guest on hour show)
<http://thekojonnamdishow.org/shows/2012-05-07/free-online-courses-higher-ed-future>
24. January 24, 2012 **Washington Post** by Bonnie Berkowitz and Alberto Cuadra
“Touchscreens: How they work”, I am the primary source
http://www.washingtonpost.com/national/health-science/touchscreens-how-they-work/2012/01/23/gIQAZHlAMQ_graphic.html
25. June 28, 2011 **New Scientist** by Jim Giles
“The man-machine: Harnessing humans in a hive mind”, discussed MonoTrans.
<http://www.newscientist.com/article/mg21028181.900-the-manmachine-harnessing-humans-in-a-hive-mind.html>
26. June 14, 2011 **WAMU Kojo Nnamdi Show**
“A Shift in Cloud Computing” (guest on hour show)
<http://thekojonnamdishow.org/shows/2011-06-14/shift-cloud-computing>
27. May 13, 2011 **Discovery News** by Alyssa Danigelis
“Tech Turns Air into a Multi-touch screen”, quoted.
<http://news.discovery.com/tech/touch-screen-technology-zero-touch-110513.html>
28. April 28, 2011 **Converge Magazine** by Tanya Roscorla
“Music Apps Take Top Prizes in U of Maryland Contest”, discusses mobile app from the University of Maryland mobile app contest that I ran.

29. April 26, 2011 **Voice of America** by Do Noghte Sefr (in Persian)
“Mobility Initiative ابتکار تحرک”
<http://www.youtube.com/donoghteseifr>
30. April 24, 2011 **Washington Post**
“Business Rx: A jukebox democracy in search of voters”, discusses mobile app from the University of Maryland mobile app contest that I ran.
31. February 25, 2011 **New Scientist** by Jim Giles and Jacob Aron
“Crowdsourced translations get the word out from Libya”, discusses my MonoTrans project
<http://www.newscientist.com/article/dn20174-crowdsourced-translations-get-the-word-out-from-libya.html>
32. February 9, 2011 **DiamondBack, UMD Student Newspaper**
“Creativity at her fingertips” by Erin Egan
http://www.diamondbackonline.com/news/article_f7143841-131c-5695-a310-42596336345d.html
33. February 8, 2011 **WAMU Kojo Nnamdi Show**
“A Revolution in Tech Jobs? The Murky World of Online Labor” (hour show with me as only guest)
<http://thekojoonnamdishow.org/shows/2011-02-08/revolution-tech-jobs-murky-world-online-labor>
34. November 23, 2010 **WAMU Kojo Nnamdi Show**
“The End of Email and other Tech Prognostications”
(hour show with me and one other guest)
<http://thekojoonnamdishow.org/shows/2010-11-23/end-email-and-other-tech-prognostications>
35. November 9, 2010 **MIT Technology Review** by Erica Naone
“Gesturing at Your TV Isn’t Ready for Prime Time” quotes me discussing Microsoft Kinect
<http://www.technologyreview.com/computing/26691/>
36. October 27, 2010 **MIT News** by Larry Hardesty
“Programming crowds” quotes me in analyzing Rob Miller’s work
<http://web.mit.edu/newsoffice/2010/programming-crowds-1027.html>
37. October 2, 2010 **News 8 TV Austin**
“App Wrap: eReading for Kids” discussing StoryKit app for iPhone
<http://www.news8austin.com/content/headlines/274426/app-wrap--ereading-for-kids>
38. August 24, 2010 **WAMU Kojo Nnamdi Show**
“Automating Language Translation: The Future of a Unified Internet?” discussing my research and the general topic of crowdsourcing and translation.
<http://thekojoonnamdishow.org/shows/2010-08-24/automating-language-translation-future-unified-internet>
39. May 4, 2010 **Technology Review**
“Redesigning the Web for Touch Screens”
http://www.technologyreview.com/printer_friendly_article.aspx?id=25236&channel=computing&section=
40. April 2010 **FutureGov Magazine** by Kelly Ng, Singapore
Description of ICDL for iPad application
41. April 17, 2010 **ABC 7 News, WLS-TV Chicago**
“Chicago Students Face Off in Battle of Books”
<http://abclocal.go.com/wls/story?section=news/local&id=7392055>
42. April 8, 2010 **Publisher’s Weekly**
“The iPad Meets the Children’s Book”
http://www.publishersweekly.com/article/455914-The_iPad_Meets_the_Children_s_Book.php?nid=2788&source=link&rid=16811925
43. April 3, 2010 **The Independent**, London
“Children’s book iPad apps offer multilingual tales, long-distance bedtime stories”
<http://www.independent.co.uk/arts-entertainment/books/childrens-book-ipad-apps-offer-multilingual-tales-longdistance-bedtime-stories-1934847.html>

44. January 28, 2010 **Baltimore Sun**, Baltimore, MD
“Apple unveils tablet computer” by Gus G. Sentementes
<http://www.baltimoresun.com/business/technology/bal-bz.ipad28jan28,0,5120135.story>
45. December 2009 **FutureGov Magazine** by Kelly Ng, Singapore
“Inside the Teacher’s Lounge” covered Ben’s keynote UNESCO talk in Hangzhou, China
46. November 20, 2009 **World Bank Blog**
<http://blogs.worldbank.org/edutech/icdl-1>
47. November 2, 2009 **Technology Review.com**
“First Test for Election Cryptography”
<http://www.technologyreview.com/web/23836/>
48. December 23, 2008 **WAMU Kojo Nnamdi Show**, Washington, DC
“Computer Guy”
49. September 18, 2008 **America.gov**
“Kids Help Design International Children’s Digital Library” by Jeffrey Thomas
<http://www.america.gov/st/educ-english/2008/September/200809181459141CJsamohT0.2946283.html>
Also published in **gulf-times.com** in Qatar
http://www.gulf-times.com/site/topics/article.asp?cu_no=2&item_no=245520&version=1&template_id=46&parent_id=26
50. April 25, 2008 **C|Net News.com**
“The Six Secrets of Mobile Computing Success” by Tom Krazit
51. February 8, 2008 **WYPR Maryland Morning**, Baltimore, MD
10 minute radio interview about usability of voting systems
52. January 30, 2008 **KCSN News** by Jessica Caldera
Short radio interview about usability of voting systems
53. January 29, 2008 **Technology Review.com** by Erica Naone
“Voting with (Little) Confidence”
<http://www.technologyreview.com/Infotech/20122/>
54. January 28, 2008 **ScienceDaily.com**
“Touch Screen Voting A Hit; Critics Miss Mark on Security, Study Says”
<http://www.sciencedaily.com/releases/2008/01/080123170954.htm>
55. January 26, 2008 **Newsweek** by Barrett Sheridan
“Like a Super Hero: Humans weren’t made for scrolling and searching. We were made for zooming.”
<http://www.newsweek.com/id/105532/output/print>
56. January 25, 2008 **Government Executive.com** by Gautham Nagesh
“Voters confused by e-voting machines, study finds”
<http://www.govexec.com/dailyfed/0108/012508n1.htm>
57. January 22, 2008 **WAMU Kojo Nnamdi Show**, Washington, DC
Thirty minute segment on “Technology User Frustrations – Spotlight on Cellphones”
<http://wamu.org/programs/kn/08/01/22.php#19083>
58. January 16, 2008 **KUOW, The Conversation**, Seattle, WA
The guest on an hour-long call-in talk show on the future of cellphones.
<http://www.kuow.org/defaultProgram.asp?ID=14159>
59. October 30, 2007 **WAMU Kojo Nnamdi Show**, Washington, DC
“Computer Guy”, talking about software reliability
60. August 7, 2007 **WAMU Kojo Nnamdi Show**, Washington, DC
“Computer Guy”, current tech events, iPhone, Tivo, and listeners email and calls
<http://wamu.org/programs/kn/07/08/07.php>

61. June 13, 2007 **New York Times** by John Markoff
"That iPhone has a keyboard, but it's not mechanical"
<http://www.nytimes.com/2007/06/13/technology/13phone.ready.html>
62. June 5, 2007 **WAMU Kojo Nnamdi Show**, Washington, DC
"Computer Guy", Microsoft Surface, cell phone problems, and listeners email and calls
<http://wamu.org/programs/kn/computerguys/07/cg070605.php>
63. April 3, 2007 **WAMU Kojo Nnamdi Show**, Washington, DC
"Computer Guy", Physical and virtual convergence, and listeners email and calls
<http://wamu.org/programs/kn/computerguys/07/cg070403.php>
64. March 17, 2007 **Science News** weekly magazine, by Ivars Peterson
"Games Theory", several quotes by me on article about von Ahn's games.
<http://www.sciencenews.org/articles/20070317/bob9.asp>
65. January 18, 2007 **Wall Street Journal**, by Jeremy Wagstaff
"The March of Time", covering DateLens
66. December 31, 2006 **Associated Press**, by Brian Bergstein
"Low-Cost Laptop Could Transform Learning"
<http://www.washingtonpost.com/wp-dyn/content/article/2006/12/31/AR2006123100380.html>
AP story picked up in 175 locations, including:
 - Washington Post
 - BusinessWeek
 - Chicago Tribune
 - Boston Globe
 - ABC News
 - CNN
 - FOX News
 - Eyewitness News
 - Forbes
 - BBC News, UK
 - Sydney Morning Herald, Australia
 - The Age, Australia
 - Economic Times, India
67. December 5, 2006 **WAMU Kojo Nnamdi Show**, Washington, DC
"Computer Guy", Vista, holiday tech gifts, and listeners email and calls
<http://www.wamu.org/programs/kn/06/12/05.php#12452>
68. September 22, 2006 **Salon.com** by Katharine Mieszkowski
"Written out of the story", discussing write-in votes using the Hart eSlate system
http://www.salon.com/news/feature/2006/09/22/delay_write_in/index_np.html
69. May 2, 2006 **WAMU Kojo Nnamdi Show**, Washington, DC
"Computer Guy – life in the fast lane", discussing broadband at home
<http://www.wamu.org/programs/kn/06/05/02.php#10686>
70. April 23, 2006 **Pittsburgh Post-Gazette** by Mark Roth
"Experts see computers getting bigger and smaller at the same time"
<http://www.post-gazette.com/pg/06113/684425-96.stm>
71. March 9, 2006 **Baltimore Sun** by Mike Himowitz
"PhotoMesa a nifty way to organize photographs"
<http://www.baltimoresun.com/technology/bal-bz.himowitz09mar09,0,3518441.column>
72. February 21, 2006 **WAMU Kojo Nnamdi Show**, Washington, DC
"Wiki Sites"
<http://www.wamu.org/programs/kn/06/02/21.php#10383>

73. February 3, 2006 **Baltimore Sun** by Frank D. Roylance
 "Spreading the e-word" (quoted about Sony's new ebook reader e-paper)
<http://www.baltimoresun.com/news/health/bal-hs.eink03feb03.1.3010066.story>
74. January 13, 2006 **Washington Business Journal** by Jennifer Nycz-Conner
 "What was I doing, again?"
<http://www.bizjournals.com/washington/stories/2006/01/16/focus1.html>
75. November 18, 2005 **c|net News.com** by Stefanie Olsen
 "The 'millennials' usher in a new era" (about ICDL)
http://news.com.com/The+millennials+usher+in+a+new+era/2009-1025_3-5944666.html?tag=nefd.lede
76. September, 2005 **Chicago Parent** by Jane Huth
 "Screen time=reading time"
<http://www.chicagoparent.com/main.asp?SectionID=10&SubSectionID=32&ArticleID=529&TM=67415.63>
77. September 22, 2005 **Delphos Herald**
 "Site Seeing"
<http://www.delphosherald.com/page2.php?story=8655&archive=>
78. April, 2005 **Chicago Parent** by Judy Belanger
 "Explore the world by turning pages"
<http://www.chicagoparent.com/main.asp?SectionID=10&SubSectionID=33&ArticleID=306&TM=67012.6>
79. April 5, 2005 **EE Times** by Nicolas Mokhoff
 "Microsoft touts 'thumb-as-stylus' interface progress"
<http://www.eetimes.com/showArticle.jhtml;jsessionid=NHVZOY3L1AQWUQSNDBGCKH0CJUMKJV N?articleID=160500234>
 Also picked up by **Information Week**
80. March 10, 2005 **folha equilibrio (Brazil)** by Sérgio Rizzo
 "Cibertentações" (article about computer-based distraction, describing some of my work.)
81. March, 2005 **Radio City (Ecuador) / BBC Affiliate**
 5 minute radio interview (performed in English, translated into Spanish)
82. March 1, 2005 **The Statesman (India)** by Stanley Theodore
 "Webcrawler: All About E-Attention Deficit Disorder"
<http://www.thestatesman.net/page.news.php?clid=24&theme=&usrsess=1&id=70119>
83. February 28, 2005 **Washington Times** by Ann Geracimos
 "New Chapter in Online Books"
<http://washingtontimes.com/metro/20050227-101610-1015r.htm>
84. February 18, 2005 **North Carolina State Technician** by Danny Frye
 "Research targets computer distraction such as AIM"
<http://technicianonline.com/story.php?id=011167>
85. February 13, 2005 **Bloomberg Radio, "Simply Put"** by Michael Goldman & Tom Moroney
 9 minute interview discussing attention and interruptions on the computer
86. February 10, 2005 **New York Times** by Katie Hafner
 "You There, at the Computer: Pay Attention"
<http://www.nytimes.com/2005/02/10/technology/circuits/10info.html>
 Also picked up by
 - a. **International Herald Tribune (France)**
 - b. **ABS CBN News (Philippines)**
 - c. **CNET News.com**
 - d. **Spartanburg Herald (SC)**
 - e. **Wilmington Morning Star (NC)**
 - f. **Minneapolis Star Tribune (MN)**
87. Nov 11, 2004 **Maryland Public Television** by Frank Batavick
 "Message in a Bottle", Bob the vid-tech guy 30 minute special including 5 minute segment of Allison Druin and me explaining email and instant messaging to children.

88. May 7, 2004 **Baltimore Sun** by Elizabeth Schuman
"Research Lab Crosses Cultures"
89. October 20, 2003 **The Feature** by Jeff Goldman
"A View Into Your Phone"
<http://www.thefeature.com/article?articleid=100146>
90. October 10, 2003 **The Independent** by Charles Arthur
"Treemap Offers a New Way of Visualizing Complex Data. Nothing Beats it for Simplicity and the "wow!" effect when you first use it."
<http://news.independent.co.uk/digital/features/story.jsp?story=451099>
91. July, 2003 **Higher Learning – Technology Serving Education**
"Web Sites: The International Children's Digital Library"
92. July 7, 2003 **Pittsburgh Post-Gazette** by Karen MacPherson
"Web Site Invites Children to Read Online: International Children's Digital Library Caters to Young Around the Globe"
93. June 15, 2003 **San Antonio News 9** by Michael Garfield
"Website a Fun Way to Encourage Kids to Read"
94. April 11, 2003 **San Jose Mercury News** by Dan Gillmor
"Designing New Handhelds to Improve Interactions Between Humans, Computers"
Also picked up by **New Jersey Star Ledger (NJ)**
95. February 12, 2003 **Tech TV Live** by Peter Barnes
Also on **abcnews.com**
"Easy Online Reader: An International Online Nook for Children's Books"
96. January 17, 2003 **Voice of America TV** by Larry Clamage
4.5 minute video of children's design team focusing on ICDL
97. December, 2002 **Sun's Java Website**
ICDL Linked to from <http://java.sun.com/industry/>
98. December 17, 2002 **Associated Press**
Picture of Ben Bederson w/ Bill Gates at Microsoft about DateLens project
 - a. **Seattle Post Intelligencer (WA)**
 - b. **Times Daily (AL)**
 - c. **Sarasota Herald-Tribute (FL)**
 - d. **Naples Daily (FL)**
99. December 16, 2002 **Baltimore Sun** by Alex MacGillis
"Two Study Intricacies of Voting Machines"
Also picked up by AP, excerpts in:
 - a. **USA Today**
 - b. **WMAR (MD)**
 - c. **WBAL-TV (MD)**
 - d. **News Channel 8.net**
100. December 12, 2002 **Philadelphia Inquirer** by Joyce Kasman Valenza
"tech.life @ school | New virtual library offers children much"
101. December 10, 2002 **Le Monde (France)** by Chantal Dussuel
"Livres en ligne pour internautes on herbe"
102. December 10, 2002 **Christian Science Monitor**
News in Brief – ICDL
103. December 5, 2002 **New York Times** by Lisa Guernsey
"Online Library Project Plans a Digital and Cultural Trove for Children"
104. December 1, 2002 **Die Burger (South Africa)**
"Paragrawe"

105. November 30, 2002 **The Vancouver Sun (Canada)**
"Book Endz: Free Kid's Books"
106. November 30, 2002 **La Vanguardia (Spain)**
"Nuevos Proyectos: TalentMatch.com, International Children's Digital Library SoapCity"
107. November 27, 2002 **USA Today** by Karen Thomas
"World Books, Children Click"
108. November 27, 2002 **The Ohama World-Herald**
"TechBytes: Digital Library Links Kids to Other Cultures"
109. November 27, 2002 **Education Week**
"Internet Library for Children Invites the World to Read"
110. November 26, 2002 **The Gold Coast Bulletin (Australia)**
"The Net: News Bytes"
111. November 25, 2002 **intern.de (Germany)**
"Kinderbibliothek"
112. November 25, 2002 **Seattle Post-Intelligencer**
"Kids Books Now Online"
113. November 21, 2002 **AP Wire** by Sam Hannel
"Web Site to Give Free Access to Children's Books"
- a. **ABCNews.com**
 - b. **The Age (Australia)**
 - c. **Ananova.com (UK)**
 - d. **Asia Pacific Media Network**
 - e. **The Baltimore Sun**
 - f. **Globetechnology.com (Canada)**
 - g. **KTVU 2 News (Bay Area)**
 - h. **Indiana Gazette**
 - i. **MSNBC.com**
 - j. **Newsday.com**
 - k. **Rapid City Journal**
 - l. **San Jose Mercury News**
 - m. **Sarasota Herald-Tribune (Florida)**
 - n. **Santa Fe New Mexican**
 - o. **Seattle Post-Intelligencer**
 - p. **Sydney Morning Herald (Australia)**
 - q. **Waterloo/Cedar Falls**
 - r. **WFTV (Florida)**
114. November 21, 2002 **24ur.com (Slovenia)**
"Knjige brezplačno na internetu"
115. November 21, 2002 **Courier Mail (Australia)**
116. November 21, 2002 **United Press International (UPI)**
"New Online Library for Kids Launched"
117. November 21, 2002 **Washington Post (.COM Column)** by Leslie Walker
"A Library for Young Browsers"
- a. **International Herald Tribune**
 - b. **SFGate.com (San Francisco)**
118. November 21, 2002 **Internet Magazine**
"Site Offers Free Access to Kids Books"
119. November 20, 2002 **Magasinet (Norway)**
"Lager verdensbarnebibliotek"
120. November 20, 2002 **Fox 5 TV News** by John Henrehan
Live 1.5 minutes on 5:00pm news

121. November 18, 2002 **All Things Considered on NPR** by David Kestenbaum
"Library for Kids Goes Online" (11 minutes)
122. October, 2002 **Digital Computer Magazine**
Review of PhotoMesa
123. September 23, 2002 **ComputerSweden** by Anders Lotsson
"Square Zoom in the Hand"
124. August 19, 2002 **Ohio Repository** by Joan Renner
"Ensuring validity is a big concern in electronic voting"
125. August 18, 2002 **Ohio Repository** by Joan Renner
"Voting test shows pros, cons"
126. August 12, 2002 **eWeek**, by Anne Chen
"DateLens Stays Organized in C#"
127. August 5, 2002 **Grid Today**
"Bill Gates Highlights Academic Collaboration Key to Future"
128. July 29, 2002 **eWeek**, by Anne Chen
"Gates: .NET Won't Happen Overnight"
129. July 29, 2002 **InfoWorld**, by Matt Berger
"Gates, academics joins on security, shared source"
130. July 29, 2002 **NetworkWorldFusion**, by Matt Berger
"Gates, academics joins on security, shared source"
131. May 9, 2002 **Washington Post (.COM Column)**, by Leslie Walker
"A Visual Rather Than Verbal Future"
132. May 9, 2002 **Washington Post Website**, by Eleanor Hong
"Human-Computer Interaction Lab at the University of Maryland" – Interview with me, Allison Druin, and Ben Shneiderman.
133. August 1, 2001 **Sun Microsystems Java Website**, by Jon Byous
"View With a Zoom: Browse and Zoom Digital Images in One Application" describes PhotoMesa and my treemap algorithms.
134. July 2, 2001 **ComputerWorld** by Sami Lais
"Treemaps Bloom" describes my photo browser application PhotoMesa which uses Jazz
135. June 4, 2001 **ComputerWorld** by Sami Lais
"Software for the 4th Dimension" describes TimeSearcher which uses Jazz
136. August 12, 2000 **New Scientist Magazine**
"Maximizing Minimalism", an article Zoomable User Interfaces, KidPad, and Fisheye Menus
137. Fall, 2000 **College Park**, by Tom Ventsias
"Zooming Ahead", an article about Zoomable User Interfaces
138. June 26, 2000 **ComputerWorld**, by Sami Lais
"Zoomin' Ahead", an article about Zoomable User Interfaces and Fisheye Menus
139. September, 1999 **Sun Microsystems Java Website**, by Jon Byous
An interview with me under their face-to-face column.
140. December 21, 1998 **Baltimore Sun**
Reprinted in New York Newsday, Dec. 30, 1998
"Children first: In designing toys, one lab turns to the experts in the field."
141. April 4, 1997 **The Chronicle of Higher Education** by Jeffrey R. Young
"New Ways of Organizing Data Could Change Nature of Computers"
142. February, 1997 **Nov'Art**
a French Magazine sponsored by Art3000, a society devoted to integration of technology and art.

-
143. October 23, 1996 **The New York Times Online (CyberTimes)** by Ashley Dunn
"Seeing the Forest of Knowledge for the Trees of Data"
144. June, 1995 **The X Advisor**, by Gary Welz
"Peripheral Visions: Zooming Through Information Space on Pad++"

3. TEACHING AND ADVISING

A. COURSES TAUGHT

i. General

Fall 2015	<i>Paths to CS</i> (CMSC198C/D/E) – 96 Students
Fall 2013	<i>Paths to CS</i> (CMSC198C/D/E, 1-3 credits) – 36 Students CMSC198C Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.6 (College comparison: 2.8) Effective teacher: 3.9 (College comparison: 2.9) CMSC198D Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.8 (College comparison: 2.8) Effective teacher: 4.0 (College comparison: 2.9)
Fall 2012	<i>Tech for CS Startups – Team taught</i> (CMSC 498E, 1 credit) – 19 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.0 (College comparison: 3.2) Effective teacher: 3.7 (College comparison: 2.9)
Spring 2012	<i>Reading Seminar: Future of HCI – Team taught</i> (CMSC 838B, 1 credit) – 13 Students
Fall 2011	<i>Social Computing Seminar</i> (CMSC 838B, 3 credits) – 20 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.7 (College comparison: 3.2) Effective teacher: 3.8 (College comparison: 3.2)
Fall 2011	<i>Intro to Human-Computer Interaction</i> (CMSC 434) – 37 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.5 (College comparison: 3.2) Effective teacher: 3.5 (College comparison: 3.0)
Spring 2011	<i>Intro to Human-Computer Interaction</i> (CMSC 434) – 42 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 2.9 (College comparison: 3.2) Effective teacher: 3.0 (College comparison: 3.0)
Fall 2010	<i>Reading Seminar in Human Computation</i> (CMSC 838B, 1 credit) – 9 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.8 (College comparison: 3.2) Effective teacher: 3.9 (College comparison: 3.5)
Fall 2010	<i>Digital Cultures and Creativity</i> (HDCC 105, co-taught) – 56 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 2.4 (College comparison: 2.5) Effective teacher: 3.2 (College comparison: 3.4)
Spring 2010	<i>Object-Oriented Programming I</i> (CMSC 131H) – 10 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.8 (College comparison: 2.9) Effective teacher: 4.0 (College comparison: 3.0)
Fall 2009	<i>Intro to Human-Computer Interaction</i> (CMSC 434) – 39 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.3 (College comparison: 3.2) Effective teacher: 3.5 (College comparison: 3.0)

Spring 2008	<i>Intro to Human-Computer Interaction</i> (CMSC 434) – 32 Students Student Evaluations (Scale: 0 Poor – 4 Excellent) Students “learned a lot”: 3.1 (College comparison: 3.1) Effective teacher: 3.2 (College comparison: 3.0)
Spring 2005	<i>Developing User Interfaces</i> (CMSC 498B) – 14 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.6 Instructor's teaching: 3.8
Spring 2004	<i>Object Oriented Programming I</i> (CMSC 131) – 75 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.0 Instructor's teaching: 2.9
Spring 2003	<i>Information Visualization</i> (CMSC 828B) – 17 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.1 Instructor's teaching: 3.4
Spring 2002	<i>Developing User Interfaces</i> (CMSC 498B) – 25 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.6 Instructor's teaching: 3.9
Fall 2001	<i>Human Computer Interaction</i> (CMSC 434/828S) – 50 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.5 Instructor's teaching: 3.6
Spring 2001	<i>Information Visualization</i> Taught by Catherine Plaisant and Ben Shneiderman due to my illness
Fall 2000	<i>Human Computer Interaction</i> (CMSC 434/828S) – 50 Students No evaluation due to my illness
Spring 2000	<i>Computer Graphics</i> (CMSC 427) – 45 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.4 Instructor's teaching: 3.3
Fall 1999	<i>Human Computer Interaction</i> (CMSC 434/828S) – 44 Students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.9 Instructor's teaching: 3.9
Spring 1999	<i>Zoomable User Interfaces</i> (CMSC 838B) – 15 Students
Fall 1998	<i>Software Engineering</i> (CMSC 435) – 45 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.6 Instructor's teaching: 3.5
Spring 1998	<i>Computer Graphics</i> (CMSC 427) – 24 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.7 Instructor's teaching: 3.7
Spring 1997	<i>Software Engineering</i> (CS460) @ UNM (Didn't receive evaluations)
Fall 1996	<i>Introduction to Data Structures</i> (CS251) - 100 Students @ UNM (Didn't receive evaluations)

Spring 1996	<i>Human Computer Interaction (CS462) @ UNM</i> Student Evaluations (ICES scale: 1 poor - 6 excellent) Course content: 5.2 Instructor: 5.5 Course in general: 5.2
Spring 1996	<i>Computers & Society (CS491/591) @ UNM</i> Introduced this course for the first time in the CS dept. 1 credit seminar Student Evaluations (ICES scale: 1 poor - 6 excellent) Course content: 4.9 Instructor: 4.8 Course in general: 4.8
Spring 1996	Taught one week <i>Computer Ethics @ UNM</i> section of a new five week introductory class on computing at UNM (CS101). Did this two times.

ii. Specialized

Fall 2016	<i>Learning Analytics Research Group (TLTC 708)</i> – 1 credit, 3 students, 20 participants
Spring 2016	<i>Learning Analytics Research Group</i> – 0 credits, 30 participants
Fall 2015	<i>Learning Analytics Reading Group</i> – 0 credits, 20 participants
Spring 2015	<i>Learning Analytics Reading Group</i> – 0 credits, 20 participants
Fall 2003	<i>Readings in HCI Seminar (CMSC 838B)</i> – 1 credit, 13 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.9 Instructor's teaching: 3.8
Fall 2001	<i>Readings in HCI Seminar (CMSC 838B)</i> – 1 credit, 12 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.8 Instructor's teaching: 3.8
Fall 2000	<i>Readings in HCI Seminar (CMSC 838B)</i> – 1 credit, 15 students
Fall 1999	<i>Readings in HCI Seminar (CMSC 838B)</i> – 1 credit, 6 students
Spring 1999	<i>Zoomable User Interfaces (CMSC 838B)</i> – 15 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 3.5 Instructor's teaching: 3.2
Fall 1998	<i>Readings in HCI Seminar (CMSC 838B)</i> - 1 credit, 9 students Student Evaluations (Scale: 0 Poor - 4 Excellent) Course quality: 4.0 Instructor's teaching: 3.8

F. ADVISING RESEARCH DIRECTION

i. Undergraduate

Independent Study Advisor:

2013	Andrej Resevic – new Paths to Computer Science course
2012 – 2013	Danny Michaelis – new Paths to Computer Science course
2010 – 2011	Jonathan Speiser Worked on tournament programmer in educational technologies group
2003-2004	Christian Klein Studied effect of animation when scrolling while reading

2000 – 2002	Rob Sherman Customized undergraduate degree in HCI & Law – focused on voting systems
Fall 2001	Steve Betten Visualizations for real-time monitoring
Spring 2000/ Fall 2000	Beth Weinstein Geographical interfaces for digital library for children
Spring 1999	Nuno Pereria End-user customizable systems
Spring 1996	Philip Eckenroth and Mike Tipping Built a 3D modeling system
Fall 1995	Josh Saiz Novel searching techniques in Pad++ (dynamic indicators)

ii. Master's

Thesis Committee:

2005	Julie Staub in Applied Math. Research on Chaum's scheme for encrypted voter receipts.
1999	George Ziets in Psychology. Advisor is Kent Norman. Research is on gauge design for multiple orders of magnitude.

Independent Study Advisor:

2013	Alina Goldman (iSchool) – Motivation in a self-paced computer science course
2012	Adam O'Sullivan – Computer vision for relative phone location
2000	Stephanie Thibeault (Dance Dept.) Technology and dance
1996	Ellen Killion Built a software library access tool in Pad++
1995	Tom Claus Worked on KidPad
1995	Sam Cancilla Shared calendar application in Pad++
1995	David Rogers Tossing for user interfaces
1995	Richard Bleakman Interfaces for teaching genetic algorithms
1995	Tom Claus Worked on Pad++

iii. Doctorate

Dissertation Advisor:

2017	Tak Yeon Lee, Ph.D. (2017) in Computer Science. "Toward Symbiotic Human-AI Interaction Focusing on Programming by Example". Currently at Adobe Research.
2016	Adil Yalcin, Ph.D. (2016) in Computer Science. "Towards Rapid, Effective, and Expressive Visual Data Exploration". Currently founding startup to commercialize his dissertation research - https://keshif.me/.

-
- | | |
|--------------------------------|---|
| 2014 | Alex Quinn, Ph.D. (2014) in Computer Science. “Crowdsourcing Decision Support: Frugal Human Computation for Efficient Decision Input Acquisition”. Currently Assistant Professor of ECE at Purdue. |
| 2012 | Chang Hu, Ph.D. (2012) in Computer Science. “Crowdsourced Monolingual Translation”. Currently at Microsoft. |
| 2007 | Amy Karlson, Ph.D. (2007) in Computer Science. Research on personal information management. Winner of Microsoft Graduate Fellowship. Two CHI papers nominated for best paper. Worked at Microsoft Research for several years. |
| 2006 | Bongshin Lee, Ph.D. (2006) in Computer Science. “Interactive Visualization for Trees and Graphs”. Co-first place winner of IEEE InfoVis 2004 Contest. Currently at Microsoft Research. |
| 2005 | Hilary Hutchinson, Ph.D. (2005) in Computer Science. “Children’s Interface Design for Searching and Browsing”. Awarded UMD CS Minker Fellowship. Awarded NSF Graduate Fellowship. Currently at Google. |
| 2005 | Bongwon Suh, Ph.D. (2005) in Computer Science. “Image Management Using Pattern Recognition”. Worked at PARC, then at Adobe Research, now Assoc. Prof. at Seoul National University (Korea). |
| 2003 | Lance Good, Ph. D. (2003) in Computer Science. “Zoomable User Interfaces for the Authoring and Delivery of Slide Presentations”. Worked at PARC as researcher for several years, now at Google. |
| 2003 | Juan Pablo Hourcade, Ph.D. (2003) in Computer Science. “User Interface Technologies and Guidelines to Support Children’s Creativity, Collaboration, and Learning”. Currently Assoc. Prof. of Computer Science at the University of Iowa. |
| 1998 | Jason Stewart, Ph.D. (1998) in Computer Science at UNM. Research is on collaborative user interfaces for children using Pad++. Awarded 1995 DEC Fellowship at UNM. |
| Dissertation Committee: | |
| 2020 | Virginia Leigh Byrne (2020) in Education. Advisor is Diane Jass Ketelhut. “The Chilling Effect of Cyberbullying on Women’s Social Presence in Online Discussions” |
| 2019 | Zhenpeng Zhao (2019) in Computer Science. Advisor is Niklas Elmqvist. “Data-Driven Storytelling for Casual Users” |
| 2016 | Uran Oh, Ph.D. (2016) in Computer Science. Advisor is Leah Findlater. “Accessible On-Body Interaction for People with Visual Impairments” |
| 2016 | Kotaro Hara, Ph.D. (2016) in Computer Science. Advisor is Jon Froehlich. “Scalable methods to collect and visualize sidewalk accessibility data for people with mobility impairments” |
| 2014 | Aaron Dinin, Ph.D. (2014) in English. Advisor is Martha Nell Smith. “Hacking Literature: A Study in the Technology of Language.” |
| 2014 | Brian Danielak, Ph.D. (2014) in College of Education. Advisor is Andy Elby. “How Electrical Engineering Students Design Computer Programs.” |
| 2013 | Irene Eleta Mogollon, Ph.D. (2013) in iSchool. Advisor is Jen Golbeck. “Multilingual use of Twitter: Language Choice and Language Bridges in a Social Network.” |
| 2013 | John Alexis Guerra Gomez, Ph.D. (2013) in Computer Science. Advisor is Ben Shneiderman. “Exploring Differences in Multivariate Datasets Using Hierarchies: An Interactive Information Visualization Approach” |
| 2013 | Sureyya Tarkan, Ph.D. (2013) in Computer Science. Advisor is Ben Shneiderman. “Interactive Visual Displays for Results Management in Complex Medical Workflows” |
| 2010 | Taowei David Wang, Ph.D. (2010) in Computer Science. Advisor is Ben Shneiderman. “Interactive Visualization Techniques for Searching Temporal Categorical Data” |

2009	Jerry Fails, Ph.D. (2009) in Computer Science. Advisor is Allison Druin. "Mobile Collaboration for Young Children"
2008	Chunyuan Liao, Ph.D. (2008) in Computer Science. Advisor is Francois Guimbretiere. "PapierCraft: A New Paper-Based Interface To Support Interaction With Digital Documents"
2008	Aleks Aris, Ph.D. (2008) in Computer Science. Advisor is Ben Shneiderman. "Visualization & Exploring Networks Using Semantic Substrates"
2008	Adam Perer, Ph.D. (2008) in Computer Science. Advisor is Ben Shneiderman. "Integrating Statistics and Visualization to Improve Exploratory Social Network Analysis"
2008	Walky Rivadeneira, Ph.D. (2008) in Psychology. Advisor is Kent Norman. "Tag Clouds: How Format and Categorical Structure Affect Categorization Judgement"
2007	Matthias Mayer, Ph.D. (2007) in Dept. Informatik, Universität Hamburg where I was an outside reader. Advisor is Prof. Dr. Arno Rolf. "Visualizing Web Sessions".
2007	Leslie Eugene Chipman, Ph.D. (2007) in Computer Science. Advisor is Allison Druin. "Collaborative Technology for Young Children's Outdoor Education".
2006	Haibin Ling, Ph.D. (2006) in Computer Science. Advisor is David Jacobs. "Techniques for Image Retrieval: Deformation Insensitivity and Automatic Thumbnail Cropping".
2005	Jinwook Seo, Ph.D. (2005) in Computer Science. Advisor is Ben Shneiderman. "Information Visualization Design for Multidimensional Data: Integrating the Rank-By-Feature Framework with Hierarchical Clustering".
2005	Jen Golbeck, Ph.D. (2005) in Computer Science. Advisor is Jim Hendler. "Computing and Applying Trust in Web-Based Social Networks".
2003	Harry Hochheiser, Ph.D. (2003) in Computer Science. Advisor is Ben Shneiderman. Research on interfaces for time-series data.
2002	Jenny Sterling, Ph.D. (2002) in Music. Advisor is Laszlo Payerle. Research on educational software she developed to teach undergrad music form. She won the campus award on educational technologies from the Center for Teaching Excellence.
2001	Mike Beynon, Ph.D. (2001) in Computer Science. Advisor is Joel Saltz. Research on filter-based programming model for grid computing.
2001	Angela Boltman, Ph.D. (2001) in Education, Advisor is Allison Druin. Research on KidPad's effect on children's storytelling.
2001	Jeff Heflin, Ph.D. (2001) in Computer Science. Advisor is Jim Hendler. Research on the semantic web.
2001	Egemen Tanin, Ph.D. (2001) in Computer Science. Advisor is Ben Shneiderman. Research on browsing large online databases using exploratory overviews.
2000	Vasanth Philomin, Ph.D. (2000) in Computer Science. Advisor is Larry Davis. Research on real-time computer vision for in-car navigation and warning systems.
2000	Byoung-Kee Yi, Ph.D. (2000) in Computer Science. Advisor is Christos Faloutsos. Research on similarity search and data mining in time-sequence databases.
2000	Chris North, Ph.D. (2000) in Computer Science. Advisor is Ben Shneiderman. Research on tight coupling between different visualizations.
1999	Richard Potter, Ph.D. (1999) in Computer Science. Advisor is Ben Shneiderman. Research on pixel data access for end-user customization.
1999	Harald Winroth, Ph.D. (1999) in Computer Science at the Royal Institute of Technology (KTH) in Stockholm, SWEDEN. I was invited to be the <i>opponent</i> . Advisor is Jan-Olof Eklundh Research on dynamic projective geometry.

-
- 1999 Ismail Haritaoglu, Ph.D. (1999) in Computer Science. Advisor is Larry Davis. Research on real-time computer vision for tracking people.
 - 1999 Zhijun Zhang, Ph.D. (1999) in Computer Science. Advisors are Vic Basili and Ben Shneiderman. Research on perspective-based usability inspection techniques.
 - 1998 Elaine Raybourne, Ph.D. (1998) in Communication and Journalism at the University of New Mexico. Advisor is Ed Rogers. Dissertation is on the use of MUDs for anonymous role-playing to teach about issues of culture, identity, and power.
 - 1998 Fan-Tao Pu, Ph.D. (1998) in Computer Science, Advisor is David Mount. Research on data structures for view-independent 3D object visibility detection.
 - 1998 Eser Kandogan, Ph.D. (1998) in Computer Science, Advisor is Ben Shneiderman. Research on a technique for window management – elastic windows.
 - 1996 Eric Freeman, Ph.D. (1996) in Computer Science at Yale University. Advisor is David Gelernter. “Lifestreams”, a computer interface that eliminates the need for filenames by indexing all files and connecting applications to a database.

Proposal Committee:

- 2019 Maddalena Romano in Geographic Information Systems (City College of New York) Advisor is Carsten Kessler. “Distributed WiFi for Distributed GIS”
- 2018 Virginia Leigh Byrne in Education. Advisor is Diane Jass Ketelhut. “The Chilling Effect of Cyberbullying on Women’s Social Presence in Online Discussions”
- 2017 Zhenpeng Zhao in HCI. Advisor is Niklas Elmqvist. “A Framework of Media Types for Data-Driven Storytelling”
- 2012 Irene Eleta in iSchool. Advisor is Jen Golbeck. “Multilingual Use of Twitter: Language Choice and Language Bridges in a Social Network”
- 2012 John Alexis Guerra Gómez in Computer Science. Advisor is Ben Shneiderman. “Exploring Differences in Trees by Comparing Structure and Node Value Changes: An Interactive Information Visualization Approach”
- 2011 Alex Quinn in Computer Science. Advisor is me. “Crowdsourcing Decision Support: Frugal Human Computation for Efficient Decision Input Acquisition”
- 2011 Sureyya Tarkan in Computer Science. Advisor is Ben Shneiderman. “Enhancing Timely Management of Workflow with Interactive Visual Displays and Actions for Situation Awareness Processing”
- 2010 Chris Hayden in Computer Science. Advisors are Mike Hicks and Jeff Foster. “Clear and Correct Runtime Upgrades”
- 2010 Chang Hu in Computer Science. Advisor is me. “Translation by Collaboration Between Monolinguals”
- 2008 Walky Rivadeneira in Psychology. Advisor is Kent Norman. “Tag Clouds: How Format and Categorical Structure Affect Categorization Judgement”
- 2007 Jerry Fails in Computer Science. Advisor is Allison Druin. Research on children’s collaborative mobile interfaces.
- 2005 Haibin Ling in Computer Science. Advisor is David Jacobs. Research on invariance in computer vision.
- 2001 Jaime Montemayor in Computer Science. Advisor is Allison Druin. Research on physical programming for children.
- 2001 Harry Hochheiser in Computer Science. Advisor is Ben Shneiderman. Research on interfaces for time-series data.

2001	Angela Boltman in Education. Advisor is Allison Druin. Research on the technology of storytelling.
2000	Jeff Heflin in Computer Science. Advisor is Jim Hendler. Research on knowledge representation on the Web.
1998	Egemen Tanin in Computer Science. Advisor is Ben Shneiderman. Research on query previews.
1998	Thanarat Horprasert in Computer Science. Advisor is Larry Davis. Research on real-time computer vision.
1998	Vasanth Philomin in Computer Science. Advisor is Larry Davis. Research on real-time computer vision.
1998	Chris North in Computer Science. Advisor is Ben Shneiderman. Research on Tightly Coupled Windows.

iv. Post-doc

2009-2012	Tom Yeh working in applied computer vision and HCI (co-mentored with Larry Davis). Currently at Univ. of Colorado, Boulder.
-----------	---

4. SERVICE

A. PROFESSIONAL

i. Office and Committee Memberships

2015 – 2018	SIGCHI Adjunct Chair for Awards
2015	TOCHI Special Issue Editor
2015	Learning @ Scale Program Committee
2014	Learning @ Scale Program Committee
2011	CHI 2011 Papers Subcommittee Co-chair
2010	CHI 2010 Papers Subcommittee Co-chair
2005 – 2016	US ACM Committee member
2004 – 2007	SIGCHI Adjunct Chair for US Public Policy Committee
2000	CHI Conference Video Paper Co-Chair
1999	CHI Conference Video Paper Chair
1998	UIST Technical Program Committee
1998	UIST Conference Panel Chair, Technical Program Committee
1998	CHI Conference Video Co-Chair.

ii. Editorial Boards

2015 – 2018	PeerJ Computer Science , PeerJ
2012 – 2018	Transactions on Computer-Human Interaction (TOCHI) , ACM Press
2007 – 2014	Foundations and Trends in HCI (Editor-in-chief), NOW Publishers
2002 – 2016	Information Visualization , Palgrave Press

iii. Advisory Boards

2016 – 2018	Coursera , Mountain View, CA Coursera Council advisory group
2012 – 2015	Barrie School , Silver Spring, MD Board of Trustees for this independent K-12 school 2013 – 2016: Secretary

2003	Microsoft Research , Redmond, WA University Relations Faculty Advisory Board	
iv. Other non-University committees, commissions, panels, etc.		
2012	Judge for Joan Ganz Cooney, National STEM Video Game Challenge	
vi. Paid Consultancies		
2022 – present	IT Park University Honorary Rector to help lead new tech educational institution	Tashkent, Uzbekistan
2022 – present	EPAM Chief Academic Advisor - help with corporate educational programs and creation of new universities	Newtown, PA
2020 – 2022	Logitech Consultant to help with new product development	Seattle, WA
2009 – present	Consulting as an expert witness largely for patent infringement cases for a range of clients	Windsor, MA
2006 – 2014	Zumobi, Inc. Founder of startup company to create mobile information services software	Seattle, WA
2008 – 2009	Microsoft Research Consultant to help with products related to my research	Seattle, WA
2005 – 2009	Hillcrest Laboratories, Inc. Advisor to help this start-up develop novel in-home entertainment system	Rockville, MD
2003 – 2008	Windsor Interfaces, Inc. President and CEO of start-up company to commercialize some HCIL projects	Chevy Chase, MD
2005	ScholarOne, Inc. Consultant to help them with visualization in their scholarly publication and conference management software	Charlottesville, VA
2004 – 2005	ATTAP, Inc. Consultant to help this start-up develop novel mobile device interfaces for personal information management	New York, NY
2003	Microsoft Research Corp On University Relations Faculty Advisory Board	Redmond, WA
2001 – 2003	PARC Advised on project to support brainstorming	Palo Alto, CA
2000	NASA Goddard Space Flight Center Designed prototype visualizations for satellite constellation control	Greenbelt, MD
1999 – 2001	Swedish Institute of Computer Science (SICS) Worked on European Union KidStory project	Stockholm, Sweden
1998 – 1999	Lockheed Martin Astronautics Advised on use of Pad++ for image analyst application	Denver, CO
1996 – 1997	Sony Research Corporation Advised on use of Pad++ for product development	New York, NY

B. UNIVERSITY

i. Departmental

2013 – 2014	CS Educational Transformation Committee co-chair
2008 – 2012	CS Teaching Evaluation Committee
2008 – 2011	CS Department Council
2007 – 2010	Software Engineering Field Committee chair
2010	UMIACS APT Committee
2009	CS Faculty Search Committee
2007	Chair of APT committee for Evan Golub's promotion to Senior Lecturer
2006	Undergraduate curriculum committee
2006	CS Distinguished Lecturer committee
2004	UMIACS steering committee
2004	High school programming contest
2004	Undergraduate curriculum
2003 – 2005	CS Department Council
2002	CFAR faculty recruiting committee
2001 – 2002	Lab committee
2001 – 2002	On department faculty recruiting committee
2001	Taught 1.5 hour course for high school students in Governor's program.
2000	Represented the computer science department on Capitol Hill for an exhibition to encourage Congressional appropriations for NSF.
2000	Helped with Computer Science Dept. research review day – made departmental demo.
1999 – 2000	Technical staff liaison – to help CS admin and technical staff.
1999	Helped with Computer Science Programming Contest
1996 – 1997	Graduate Admissions coordinator for the UNM Computer Science Department.

iii. University

2016 – 2018	USM Faculty Workload Workgroup
2015 – 2018	DIT Learning Technology Working Group
2015 – 2017	Senate Library Council Committee
2015 – 2018	Senate Educational Affairs Committee
2014 – 2018	Provost's Data Policy Advisory Committee
2013 – 2018	USM Academic Transformation Advisory Committee
2013 – 2018	Future of Information Alliance brainstorming board
2013 – 2018	DIT Technology Accessibility Committee
2013 – 2014	Journalism Faculty Search Committee
2013 – 2014	Learning @ Technology Working Group for DIT
2012 – 2013	Provost's Commission on Blended and Online Education

2012 – 2013	VP IT/CIO – IT Council
2012	CMNS Distinguished Research Scientist Prize committee
2011	VP / CIO Search Committee
2010	UMD Campus-Wide Mobile App Programming Contest Chair
2002 – 2005	Advisory board member for MITH (Maryland Institute for Technology and the Humanities)
2000	Served on search committee for new intellectual property manager for Office of Technology Commercialization.
2000	Ran event for Maryland Day – opened lab to children to use KidPad to tell collaborative stories

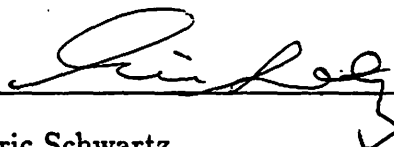
**A Miniature Space-Variant Active Vision System:
Cortex-I**

Benjamin B. Bederson

June 1992

A dissertation in the Department of Computer Science submitted
to the faculty of the Graduate School of Arts and Sciences in
partial fulfillment of the requirements for the degree of Doctor of
Philosophy at New York University

Approved: _____

A handwritten signature in black ink, appearing to read "Eric Schwartz", written over a horizontal line.

Eric Schwartz

Research Advisor

© Copyright 1992
by Benjamin B. Bederson
All Rights Reserved

Abstract

We have developed a prototype miniaturized active vision system whose sensor architecture is based on a logarithmically structured space-variant pixel geometry. A space-variant image's resolution changes across the image. Typically, the central part of the image has a very high resolution, and the resolution falls off gradually in the periphery. Our system integrates a miniature CCD-based camera, pan-tilt actuator, controller, general purpose processors and display. Due to the ability of space-variant sensors to cover large work-spaces, yet provide high acuity with an extremely small number of pixels, space-variant active vision system architectures provide the potential for radical reductions in system size and cost. We have realized this by creating an entire system that takes up less than a third of a cubic foot. In this thesis, we describe a prototype space-variant active vision system (**Cortex-I**) which performs such tasks as tracking moving objects and license plate reading, and functions as a video telephone.

We report on the design and construction of the camera (which is $8 \times 8 \times 8$ mm), its readout, and a fast mapping algorithm to convert the uniform image to a space-variant image. We introduce a new miniature pan-tilt actuator, the Spherical Pointing Motor (SPM), which is $4 \times 5 \times 6$ cm. The basic idea behind the SPM is to orient a permanent magnet to the magnetic field induced by three orthogonal coils by applying the appropriate ratio of currents to the coils. Finally, we present results of integrating the system with several applications. Potential application domains for systems of this type include vision systems for mobile robots and robot manipulators, traffic monitoring systems, security and surveillance, telerobotics, and consumer video communications.

The long-range goal of this project is to demonstrate that major new applications of robotics will become feasible when small low-cost machine vision systems can be mass-produced. This notion of "commodity robotics" is expected to parallel the impact of the personal computer, in the sense of opening up new application niches for what has until now been expensive and therefore limited technology.

Acknowledgements

This work is the result of a collaborative effort. It was a pleasure to have had the opportunity to work with such fine minds as I found in Eric Schwartz (my thesis advisor), Richard Wallace (my colleague and constant teacher), and Ping-Wen Ong (my fellow student and supporter).

Eric has been doing research leading up to this project for at least ten years. He found the funding for it, and had many of the critical ideas that got it moving. The Spherical Pointing Motor was his original idea, and only through (literally) years of bouncing it off each other, did we get it right.

Richard is largely responsible for the digital signal processing in the system. If it weren't for him, we wouldn't have a video display. He had infinite patience in helping me work through problems. He was always available to assist me, and helped to keep the project well directed.

Ping-wen came through at the last minute to integrate his license plate tracking software to work with **Cortex-I**. In addition, he was a motivating force in my studying Chinese. While not directly critical to this project, studying Chinese helped me keep my sanity during various traumas – such as when the first prototype went flying through the air as I got a great electric shock – a few hours before we departed for Chicago where **Cortex-I** was first publicly demonstrated.

Everyone at the NYU Robotics Lab was always optimistic and supportive as **Cortex-I** slowly took shape. Bud Mishra, especially, had good ideas, and was kind enough to read this thesis twice.

And of course, I thank my family, friends, and cat (Tria) who have had to put up with me as I've been counting down the months to finish this, and start the next phase of my life.

Contents

Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
1 Introduction	1
1.1 Motivation	1
1.2 Background	5
1.3 Space-variant images	10
1.4 Space-variant image processing	10
1.5 System description	15
2 Camera	18
2.1 Introduction	18
2.2 Camera optics	21
2.2.1 Zoom lens	22
2.3 Image sensor	23
2.3.1 Sensor technologies	24
2.3.2 Timing signals	26
2.3.3 Vertical smearing	28
2.3.4 Electrical noise	31
2.3.5 Future Designs	32
2.4 Fast logmap computation	32
3 Platform Pantilt	36
3.1 Introduction	36
3.2 Background	38
3.3 Mechanism	39
3.4 Forward kinematics in two dimensions	41
3.5 Inverse kinematics in two dimensions	43
3.6 Forward kinematics in three dimensions	44
3.7 Inverse kinematics in three dimensions	45
3.8 Limits of motion	46
3.9 Simulation of kinematics	46
3.10 Prototype Platform Pantilt	47

4 Spherical Pointing Motor	49
4.1 Spherical pointing motor theory	49
4.2 Theoretical Calculation of motor position	54
4.2.1 Position given currents	54
4.2.2 Currents given position	55
4.2.3 Illustration of design constraint	56
4.3 Calibration	56
4.4 Dynamics and control	64
4.5 Design Issues	69
4.6 Controller description	70
4.7 Prototype spherical pointing motor	71
4.7.1 Accuracy and precision measurements	71
4.7.2 Velocity measurements	72
5 Results	78
5.1 System architecture	78
5.2 Image processing	80
5.3 Motion tracking	80
5.4 License plate tracking & reading	82
5.5 Cortex-I limitations	82
A Glossary	87
B Variable List	90
C Constant List	92
Bibliography	93

List of Figures

1.1	Complex logmap, $\log(z)$	4
1.2	Truncated pyramid	9
1.3	Lookup tables $S(i, j)$ and $R(i, j)$	11
1.4	A TV image, an inverse logmap image, and a forward logmap image	12
1.5	Connectivity graph	12
1.6	Random map image and connectivity graph	13
1.7	Cortex-I photo	16
2.1	Images from custom logmap sensor chip	19
2.2	Camera photo	19
2.3	Zoom lens photo	20
2.4	Biconvex lens system	21
2.5	Camera mechanical drawing	22
2.6	Zoom lens illustration	23
2.7	Zoom lens mechanical drawing	24
2.8	Images from the zoom lens	25
2.9	A CMOS VBJT pixel circuit	25
2.10	Timing signals used to control the CCD image sensor.	26
2.11	Schematic diagram for CCD driver board.	27
2.12	Vertical smearing elimination	29
2.13	Desmearing results	31
3.1	Platform Pantilt photo	37
3.2	Commercial pan-tilt actuator photo	38
3.3	2D illustration of bad design for Platform Pantilt	40
3.4	2D illustration of design for Platform Pantilt based on Stewart Platform	40
3.5	2D illustration of good design for Platform Pantilt	41
3.6	3D illustration of design for Platform Pantilt that is used	42
3.7	Triangles used to solve equations	43
3.8	The limits of motion in two dimensions.	47
3.9	Simulation of the forward kinematics equations of this section	48
4.1	Spherical Pointing Motor photo	50
4.2	Illustration of internal coil SPM	50
4.3	Illustration of external coil SPM	51
4.4	Torque on a current loop	52
4.5	Internal coil SPM panned 90°	53
4.6	External coil SPM tilted 90°	53
4.7	The coordinate system used for SPM calculations	55
4.8	Calibration pattern	57
4.9	Theoretical and calibrated currents vs. position for SPM	63
4.10	Model of system response to a single impulse.	64
4.11	Uncontrolled ringing measurement	65

4.12	Ideal model of system response to a dual impulse.	66
4.13	Model of system response with inexact timing to a dual impulse.	67
4.14	Two-step control strategy measurement	68
4.15	SPM magnetic circuit simulation with and without iron return	70
4.16	Illustration of our experimental setup for measuring motor accuracy	73
4.17	Results of open-loop control strategies to reduce the ringing of the SPM with 1.75° of ringing	74
4.18	Results of open-loop control strategies to reduce the ringing of the SPM with 0.70° of ringing	75
4.19	Results of open-loop control strategies to reduce the ringing of the SPM with 0.35° of ringing	76
5.1	System architecture overview.	79
5.2	Segmentation result using connectivity graph	81
5.3	Sequence of images showing license plate tracking	83
5.4	OCR results of license plate tracking	84
5.5	Actual and estimated position of the SPM while tracking license plate	85

Chapter 1

Introduction

Computer vision is a field that typically requires very fast and expensive machines. By using a radically different image architecture along with novel imaging and actuation technologies, we have created a system that we strongly believe will allow us not to solve vision problems better than before, but to solve them dramatically faster, smaller and cheaper.

The key innovation is the use of *space-variant images*¹ which are images whose *resolution* changes across the image. The central part of the image has a very high resolution and the resolution falls off gradually in the periphery. The main advantage of this type of image is the small amount of data it contains while maintaining a high *field width* and central resolution.

This thesis discusses image processing with space-variant images, and a miniature space-variant *active vision* system, **Cortex-I**, that integrates a camera, a two degree-of-freedom motor to *actuate* the camera, and a loosely coupled parallel computing architecture to process the images and control the motor and camera.

1.1 Motivation

Traditionally, very little computer vision research has focused on real-time systems. Typical systems spend several seconds or even minutes on solitary images. This approach has not produced a general real-time device. Part of the problem is the philosophy often followed of “we need to do basic research, so we’ll write the necessary software and the hardware will eventually catch up.”

This ignores the fundamental problem of bandwidth. Using a standard 512×512 image at video rates of 30 frames per second, 8 megapixels per second need to be processed. Even with specialized image-processing hardware (e.g., DataCube’s MaxVideo system), the best general-purpose systems are still processing only the simplest scenes successfully in real-time. It is not likely that any technology in the near future will be able to supply the required processing power.

This problem is not unavoidable because typically, only a small part of the image contains the feature being examined at the moment. What is needed is a way to selectively focus *attention* on the interesting part of the image while maintaining some level of awareness of the rest of the image so interesting events can be recognized as such and then attended to.

Multiresolution images supply the opportunity to reduce bandwidth while allowing focusing of attention on one region and awareness of a wider field. Two relevant parameters of an image are the field width and the resolution. The field width specifies how large a portion of the scene is contained in the image. The resolution specifies how much information is contained in a fixed portion of the image. Neither of these parameters can be arbitrarily reduced without severely degrading the system performance. The human visual system solves the bandwidth problem without compromising either field width or *foveal*

¹Emphasized words are defined in Appendix A

resolution. It does so with the use of the retina, which is a space-variant sensor. It has a resolution that is high in the center (called the *fovea*), low in the periphery and it changes smoothly between them.

Any system that uses a space-variant sensor must aim the sensor properly. Since space-variant sensors only have high resolution in the fovea, the current region of interest must be continuously tracked or *foveated*. Such a sensor must be mounted in a device that can aim the sensor with precision. A system with this capability is an example of an *active* or *attentive vision system*.

A *logmap* sensor is a useful type of space-variant sensor that is modelled on the human visual system (see Section 1.3). The space complexity aspect of logmap sensors is particularly attractive and has been analyzed in detail [42]. In this work, a *spatial quality measure*, Q_s , for sensors is defined to be the ratio of its work-space to maximal resolution. Logmap sensors can each achieve comparable Q_s to conventional uniform sensors that are one to four orders of magnitude larger. Image simulation of the human space-variant architecture suggests that this ratio may be as high as 10,000 : 1 [49]. In current implementations of space-variant machine vision systems, logmap sensors with between 1000 and 2000 pixels have comparable Q_s to conventional sensors in the range of 256×256 to 512×512 , a compression of between 60 : 1 to 250 : 1. Moreover, as shown in [42], Q_s grows exponentially with the number of pixels in the logmap sensor, thus providing a highly favorable route to upgrading sensor quality. The high compression ratios cited above for the human visual system derive from the fact that human acuity, which is roughly one-arc minute in the fovea, when extrapolated over a 120 degree visual field, yields a constant resolution sensor with roughly 0.2 gigapixels. The logmap sensor with comparable Q_s has roughly 10^4 to 10^5 pixels.

In this discussion, compression refers to the ratio of the number of pixels in a conventional uniform sensor to that of a space-variant sensor with the same Q_s . The issue of image compression, in the conventional (e.g. JPEG) usage is an independent issue not addressed in this thesis, but we point out that a form of *progressive video coding*, in which a video image is represented by a sequence of logmap images, is one application that has benefitted from the high compression ratio of the logmap transform. If each logmap image is centered on a different point in the video image and clipped so that we preserve the highest frequency information available at each point in the video picture, we can progressively construct a video image of increasing detail. Rojer [43] reported some experiments with progressive logmap video coding, and discussed the problem of selecting the best sequence of gaze points.

A space-variant sensor requires pointing the sensor at the region of interest. If the sensor is not pointed (or foveated) properly, the desired object will fall somewhere on the periphery of the sensor resulting in a lower resolution image. So it is important to develop efficient eye movement mechanisms.

A sensor movement system must find interesting objects, track them as they move, and account for movement of the device the sensor is mounted in. A short description of the way the human system solves the second two parts of this problem follows. It should be noted that the human visual system is able to track objects and foveate on them to within the accuracy of the retina's highest resolution. There are four principal types of eye movements as reviewed by Robinson [41].

1. **Saccades:** These are discrete movements and move the eye quickly (300° - 400° /second) from one fixation point to another. They also correct errors of the pursuit system. There is a large latency (150 ms) between fixations. There are typically less than four saccades per second.
2. **Pursuit Eye Movements:** The pursuit system tracks moving targets and keeps the current target foveated. It has a latency of 50ms.
3. **Vergence:** This controls the depth that the two eyes fixate on together. It is the slowest system and has a latency of over 200ms.
4. **Vestibular Systems:** The Vestibular Ocular Reflex and Optokinetic Reflex systems maintain the gaze of the eye, counteracting for head movements. I.e., if the head moves to the right, the eye moves to the left. This is one of the fastest types of eye movement and has a latency of only 14ms. These get sensory input from the semicircular canals that effectively constitute an angular inertial accelerometer with a frequency range from 0.017 to 17Hz.

In addition to these four systems, there is a fifth type of movement known as physiological nystagmus. This is a small higher frequency jitter at 30 to 80Hz. These movements are very important to the low-level visual process, but are not an issue in the larger movements of object tracking with which we are concerned here.

In our work, we focus on saccadic motions which are the simplest to implement and are used for both attention and tracking.

The mapping of the retina to the visual cortex is another very important characteristic of the human visual system. Pioneering work by Hubel and Wiesel [26,27] showed how some low-level feature detection performed on the retinal image was mapped to the V1 area in the visual cortex. Then Schwartz showed in [47,48] that the mapping from the retina to the visual cortex takes a very specialized form. Specifically, the retina can be looked at as a polar coordinate system where each point represents one photo-sensitive cell. It gets mapped to what is essentially a rectangular coordinate system in the visual cortex. This can be represented mathematically by a complex logarithm and is depicted graphically in Figure 1.1. This mapping is called the logmap.

The point in retinal space, P_1 , can be denoted $re^{i\theta}$ and the point in cortical space, P_2 , can be denoted by the complex number $x + iy$. If we take the log of P_1 , we get:

$$\log(re^{i\theta}) = \log r + i\theta$$

Substituting

$$x = \log r \quad \text{and} \quad y = \theta,$$

we get

$$\log(re^{i\theta}) = x + iy$$

or,

$$\log(P_1) = P_2.$$

So the natural logarithm of a point in polar coordinates is transformed to a point in rectangular coordinates.

This mapping has some interesting geometric properties. Specifically, fovea-centered circles get mapped to vertical lines and radial lines get mapped to horizontal lines. This has the effect of transforming scale and rotation in fovea-centered retinal images to translation in cortical images. For many computer vision applications, it is easier to recognize translation of an object than scaling or rotation, so this provides another justification for a complex log sensor geometry. However, since objects get distorted as they move away from the fovea, it is practical to use a tracking system to keep the object of interest centered on the fovea. In the present work, we do not attempt to make use of these geometric features of the complex log mapping, but rather, we use this mapping for its inherent bandwidth reduction.

One problem in using an analytic space-variant mapping, such as the logarithm is the existence of a singularity at the origin of the coordinate system. Figure 1.1b shows this singularity. The left points of each triangle in the range all represent the same single point in the domain. Our solution to this problem is to introduce a small real constant, α , into the mapping and to use a map function of the form $\log(z + \alpha)$. The constant, α , is used in an analogous fashion in models of the human visual system: it is a measure of the size of the central linear representation of the human fovea [48]. There are still difficulties with the $\log(z + \alpha)$ mapping because there is a discontinuity in the range. This discontinuity, however, is much easier to handle than the singularity in the original mapping.

As can be seen in Figure 1.1d, the pixels on the edge are cut off in different places yielding 3, 4, and sometimes 5-sided pixels. This makes operations such as convolution difficult as it is not immediately clear how to handle the missing and extra adjacent pixels. We discuss a solution to this problem in Section 1.4

We have assumed that an interesting object has been located. But how do we find interesting objects throughout the visual field? This question is a very important one and is called *attention*. Yarbus [58] examined this problem in humans by tracking eye movements in individuals as they examined a scene.

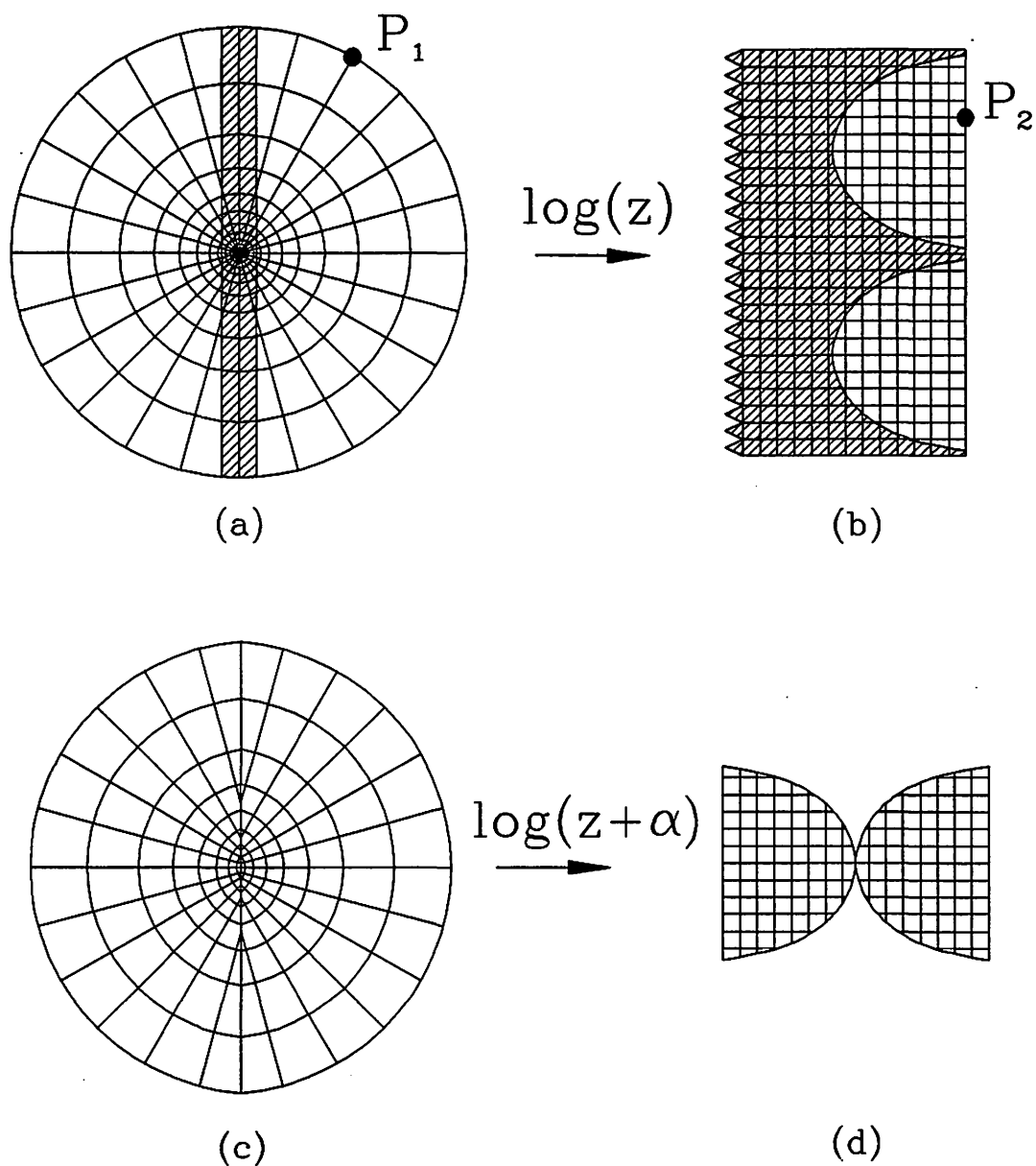


Figure 1.1: This shows the complex logmap, $\log(z)$ in a) and b) and its relative, the mapping, $\log(z + \alpha)$ in c) and d). It can be seen that the $\log(z + \alpha)$ map results from cutting the hatched region out of the $\log(z)$ map. Note that rings centered around the fovea in the domain get mapped to vertical lines in the range and radial lines get mapped to horizontal lines. Here, P_1 gets mapped to P_2 . The $\log(z)$ map has a singularity in the center while the $\log(z + \alpha)$ map has a discontinuity along the vertical meridian.

From this and other experiments, Yarbus showed that individuals tend to have consistent eye movement patterns that are different across individuals. He also showed that with supervision and a high-level goal, eye movement patterns (mainly saccades) could be controlled. For instance, in one experiment, Yarbus showed a picture to a subject and asked the subject to remember the position of the people and objects in the room, the response was a very unfocused scan covering the entire image. On the other hand, when the subject was requested to estimate how long the unexpected visitor had been away from the family, the response was to spend almost all the time examining the faces of the people. These experiments are important because they show how low-level control of the visual system is influenced by high-level processes.

An excellent review of selective fixation methods in both human and computational systems can be found in Abbott [1]. Two researchers not covered in this review are Rojer and Califano.

Rojer [43] addressed the issue of attention using space-variant images and showed several strategies for determining where to look next. He first compared several heuristic algorithms for attention. He quantified the results for attention by random fixation with and without inhibition, attention from a spatial derivative, and attention based on motion and regions. He then introduced a variation on the Hough transform that he calls the Collapsed Hough transform. It allows the feature to be of high dimension while using a "motor space" of only two dimensions, the minimum necessary to control fixation. He demonstrates results with this method to locate faces in a natural scene.

Califano et. al. [14] introduces a technique for foveation based on a two-level resolution system. They have implemented three modes of attention. One detects conflicts between object models, typically occurring when multiple objects differ only upon examination at a higher resolution. The second looks for areas that are not currently covered by any hypothesis. The last explores previously unexamined areas, much as some of Rojer's heuristic procedures do.

1.2 Background

Researchers have been examining the problems associated with space-variant sensors and active vision for many years, both in theoretical and practical ways.

On the theoretical side, Aloimonos et. al. looked at many of the shape-from-X problems [5], proving that an active search strategy is more efficient than a passive one, and that ill-posed and nonlinear problems become well-posed and linear. Tsotsos analyzed the relative complexity of active versus passive search [54]. He claims that an active search strategy may gain improvements in computation time based on worst-case time complexity functions he derived using hypothesize-and-test strategies.

Many groups have built active vision systems to experiment with some of these ideas. Krotkov [33] built what is often regarded as the first "robot head". Many other groups as detailed in Chapter 4 followed suit. The four groups described here have all built working systems that use some sort of multi-resolution technique, or in Dickmanns case, multiple foveation points.

At the Universities of Genoa and Pisa in Italy, Van der Spiegel et. al. and Sandini et. al. are doing collaborative work with Bajcsy at the University of Pennsylvania [6,32,45,46,55]. They have built a space-variant sensor and are working on geometrical and tracking algorithms for it.

Van der Spiegel et. al. has been developing a VLSI space-variant sensor based on the complex-log map, $\log(z)$. It is based on *Charge Coupled Device* (CCD) technology. They fabricated a working chip that produces images with 30 rings of 64 pixels each with a uniform fovea of 102 pixels. Each pixel gives a linear output in response to incident light intensity.

Because CCD technology will not allow direct connection between greatly different sized components, the design must be broken up into several stages, each normalizing its data before passing it on to the next. In addition, there is a limit to the size of the photo-sensitive area. These characteristics led to a design consisting of three sets of ten rings. Each new set of rings has twice as many pixel cells, each half as big as the previous set necessitating different clocks for each set of rings. The design also has a "blind spot" where the signals necessary for the center of the chip access it. The blind spot is actually

one ray that is the width of 2.5 pixels.

Sandini et. al. simulated logmap images by sub-sampling *uniform images*. They have examined three problems: 1) recognition of arbitrarily scaled and rotated 2D shapes; 2) tracking of a moving target; and 3) evaluation of distance of an object using optical flow. All three problems take advantage of the space-variant structure of the image. These experiments only used the peripheral pixels and did not take into account the uniform fovea. They did, however, simulate the blind spot accurately.

The first problem Sandini examined is based on the standard complex logmap that converts rotation and scaling to translation for foveated scenes. Using simple binary objects, they implemented a template matching algorithm based on cross-correlation and achieved a recognition rate of 97% for 50 objects that were arbitrarily rotated and scaled. They claim that the errors are largely due to the blind spot.

The second problem, that of tracking a moving target, was examined with the intent of seeing how the low resolution of the image affected the system. The target used was a bright spot over a dark background moving in parabolic and circular paths. The target was identified by subtracting the background velocity (computed directly from the known camera motion) from the optic flow. This yielded a velocity estimate for the target that was located by computing the center of mass. The camera was then positioned based on the velocity of the target. When the error reached a threshold, a "saccade" was performed correcting the error.

For the third problem, they assumed the camera was moving with pure translation towards a foveated object with constant speed. They wanted to compute the distance of the object from the camera. The optical flow vectors for this situation are radial pointing outwards for a uniform image. For the logmap image, however, the vectors are horizontal lines since radial lines get mapped to horizontal lines in the $\log(z)$ map that they use. The vector's amplitude is constant across the image and is proportional to the distance the object has moved.

These preliminary experiments show how the geometrical structure of the logmap image affects some basic vision problems. The low resolution of the image does not seem to be a problem for this simple situation, although the blind spot does cause some difficulties.

Dickmanns and Graefe at the Universität der Bundeswehr München [20] have shown their real-time vision system to work in non-trivial difficult situations. Their approach is to maintain a world model of the situation, focus processing on selected regions, and use the temporal proximity of the images to incrementally process each new image. They find that the faster they run the system, the easier the processing becomes. As the images get closer together temporally, the difficulty of the correspondence problem decreases as the images change less and less between frames. They have demonstrated three working systems. One drives a car autonomously on an autobahn construction site. Another balances a stick in one-dimension on a moving cart. The third docks a vehicle on an airbed.

They define their task as looking "for the inherent structure and ... to find a computer architecture which is well matched to the task of visual motion control." [21, p. 224]. The most important concept is that of *temporal continuity* which is the basic idea that if two pictures of a natural scene are taken within a few milliseconds of each other, they will normally be very similar to each other. They have found that most features don't move more than a pixel or two between frames.

Another important concept is that the appropriate behavior of a system typically depends on only on a small number of features. Only the area of the image containing these features needs to be examined. This area is often less than 10% of the whole image. In a dynamic system, the location of these features is usually known with fairly good precision since they probably change very little between successive images. This approach influences the choice of hardware as each region of interest is best processed in different ways. A dynamic system is better off with a coarse grained parallelism with one processor handling each region of interest. Massively parallel SIMD machines, on the other hand, are not well suited to this approach since they would have to process all pixels the same way. They would waste 90% of their time processing the uninteresting regions. In addition, SIMD machines have no mechanism for processing different regions with different algorithms at the same time.

The final idea in the work of Dickmanns and Graefe is that of a world model. Their systems work in limited domains, and are not intended to work generally. The autonomous vehicle, for instance, works

only on autobahns in good condition without other cars. Because of this, a world model can be accurately created that describes what is likely to be seen and how objects in the model are likely to act and react. The vehicle system knows about lane divider lines in the road, the edge of the road, side roads, road curvature, etc. It can predict how things are likely to change based on system state parameters (i.e., the speed of the vehicle). With each new image, the locations and shapes of the features can be accurately predicted and the world model can be updated.

The world model of Dickmanns and Graefe is a full 3D model while the images are 2D perspective projections of the world. The conversion from the 2D projection to 3D is non-unique. They solve this by first taking the 2D projection of the world model to estimate where the features are. They use knowledge based feature detection on the image (to be described shortly), and finally use a recursive least squares state estimation technique to bypass the non-unique inversion of the perspective projection and update the world model. This is a Kalman filtering [34] approach based on the Jacobian of the image features.

The world model technique runs into problems in at least three situations that are: 1) analyzing the scene at startup; 2) losing the location of features due to extreme circumstances; and 3) when new objects appear in the scene. The initialization is not difficult because that is not time critical. The other two situations, however, are very difficult problems that have not been substantially addressed in this work.

Their knowledge based feature detection is based on the world model. They predict the approximate location of the feature and use a controlled correlation technique to locate the feature precisely. This substantially reduces the cost compared to a standard correlation. They only check a small area surrounding the predicted location of the feature. They reduce the problem to one dimension by checking a path orthogonal to the expected orientation of the feature. The result is an extremely fast updating of feature position and shape.

The architecture of the system is based on up to fourteen 8086-based single board computers. Each processor examines a single region of interest. There is also one system processor that coordinates communication between the other processors. There is one wide-angle camera to examine the periphery and one camera with a telephoto lens to examine specific details. They both have pan and tilt control. The system has a video bus that can transmit up to four images simultaneously.

They have autonomously driven a small van at speeds of up to 60 mph on an autobahn construction site with no other vehicles. The van has run 20 km without human intervention in clear and moderately rainy weather. It is able to enter the road from acceleration strips, switch lanes, and go on and off exit ramps.

This approach clearly has merit and has been demonstrated to work in a variety of relatively uncluttered environments where a fairly simple and accurate model of the world exists. Its ability to generalize to complex systems was not demonstrated.

A large research group directed by Brown and Ballard at the University of Rochester [13] are exploring some of the basic concepts of active vision systems centering on eye and body movements. Their stated goals are to study real-time vision in cognition and movement.

Their system, called "The Rochester Robot", consists of two cameras mounted on a Puma robot. Each camera has separate pan and common tilt control. The Puma can move the "head" in a two meter radius with six degrees of freedom.

Their approach in fighting the bandwidth problem is to use special-purpose hardware. They are using a DataCube MaxVideo image processing system for low-level image processing and a 128 processor BBN Butterfly computer. They incorporated a second camera to have two different resolutions. One camera has a high fieldwidth and low resolution while the other has low fieldwidth and high resolution. They also talked about building a *pyramid* architecture with five levels of 64×64 images where each level subsamples the input image at a different rate.

One of the problems they have examined is that of automatically maintaining vergence on an object as the body moves. They are also studying saccade and pursuit mechanisms. Finally, they are exploring the difference between self and world motion. They note that fixating systems have effective space-variant resolution due to blurring in the periphery. However, this is based on the finite integration times

of the photo-sensitive elements of the sensor. It is not clear if egomotion is rapid enough to cause such blurring.

They create a depth map for a foveated object via motion parallax by moving the head sideways while fixating on a point. Objects in front of the point appear to move in the opposite direction of the head motion while objects behind the point move in the same direction as the head. The velocity of the object motion is proportional to the distance from the fixation point. By computing the disparities of the images taken from different positions (that they don't discuss), a depth map relative to the fixated point can be computed.

Another technique they discuss uses self-motion for dynamic segmentation. While moving and fixating on a point, the object that is fixated on becomes isolated because everything around it moves and blurs. They process this by computing a weighted average of input images over time. There are some potential problems with this technique as it is dependent proper fixation. They did not demonstrate any results with this technique.

They have also been investigating pursuit of moving objects. Their technique is based on correlating the image with a filter, which they can do very fast with the DataCube. However, their DataCube system can only perform up to 8×8 correlation at video rates. The result is that only small features that move slowly can be tracked.

Olson and Potter [36] have explored the problem of real-time vergence control: maintaining the gaze of two cameras on a single point in world coordinates. The Rochester Robot has one dominant eye that is under the control of the saccade and pursuit mechanisms. The dominant eye foveates an object, and the other eye is then aimed at the foveated object. The basic control loop grabs the left and right images, estimates the disparity, and re-aims the second eye. The difficult problem here is how to compute the disparity in real-time. They use a mechanism based on the Cepstral filter described by Yeshurun and Schwartz [59].

The Cepstral filter is computed by taking the power spectrum of the Fourier Transform of the log of the power spectrum of the Fourier Transform of the input image. The input image is a window from the left and right images spliced together. The results are peak values at $(\pm(w + d_h), \pm d_v)$ where d_h and d_v are the estimated horizontal and vertical disparities and w is the width of one image window. By analyzing the algorithm and implementing it very carefully on a Sun 3/260, they were able to speed it up from 2.6 seconds to under 0.1 seconds, which is fast enough for their system.

The Cepstral filter can be described in a couple of ways. Yeshurun and Schwartz describe it as extracting a periodic term in the log power spectrum of the original and shifted image. This periodic term corresponds to the disparity. Olson and Potter, on the other hand, prefer to think about it as a variation of the autocorrelation function where the log favors the signal with the broadest spectrum. These are impulses, so the result is an autocorrelation-like function that enhances sharp changes and is thus easier to interpret.

They have demonstrated results, but state that they still need to address the issue of the appropriate window size for the filter. They also want to work on integrating vergence with the saccadic and pursuit movements of the dominant eye. Although the individual parts of the Rochester Robot all work, they have not yet been fully integrated and have been tested only on simple geometric objects.

At the David Sarnoff Research Center, Burt built a working discrete approximation to a logmap space-variant sensor with a Laplacian pyramid approach. He [12] describes the sensor along with algorithms for its use in foveating, tracking, and interpreting objects:

A Gaussian pyramid is a set of images, each one half the resolution of the one before. A Laplacian pyramid is created by taking the difference between each level of a Gaussian pyramid. A full Laplacian pyramid contains all of the information in the original image. Burt's pyramidal approach is to take a set of Laplacian pyramid images and extract a window of the same size from each one. The result, called a *truncated pyramid*, is a set of images that take a finer and finer look at a smaller and smaller portion of the image. This approach is depicted in Figure 1.2. He constructed special-purpose hardware that computes this type of truncated pyramid. Crowley [17,18] also investigated pyramids for computer vision. He showed how multiple resolution pyramids can be used to match stereo scan lines.

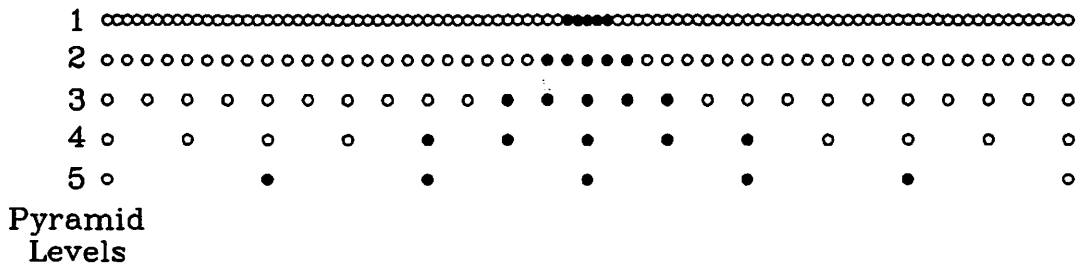


Figure 1.2: This shows the truncated pyramid structure used by Peter Burt. Each successive level represents a lower resolution image. The filled in circles represent pixels present in the pyramid. Notice that there are a constant number of pixels in the pyramid at each level where they cover a larger area at each level. This figure adapted from [12, p. 980].

The use of this architecture is based on coarse to fine analysis of an image. First, Burt picks a window size that can be processed by the CPU quickly. A window size of 16×16 is reasonable. His algorithm examines the image at the lowest level as this represents a low resolution representation of the entire image. If there is any evidence of a region of interest, then it examines the next level where a smaller area at higher resolution is represented. The algorithm climbs the pyramid in this manner with each new level directing the processing to a more specific area until the highest level is reached and the feature is fully processed. Assuming the original image is $512 \times 512 = 252,144$, and examining one window per frame, a feature will be fixated with the highest resolution after traversing all six levels. This involves processing $6 \times 16 \times 16 = 1536$ pixels, an effective data reduction of $252,144/1,536 \approx 164$.

This technique effectively replaces mechanical eye movements with electronic ones. Instead of physically moving a sensor, a region is examined within a fixed sensor. This technique obviously is much faster than a mechanical system, but is limited to examining scenes that fit within the limited field of a single image.

This architecture is able to track moving targets from a moving platform, such as when viewing a moving car from an airplane. A traditional approach is to analyze the motion between each pair of successive frames. This is computationally expensive and error prone if the relative motion of the object is small compared to the camera. Instead, if the relative motion of the background can be computed, then it can be subtracted out between successive frames and the slowly moving object will stand out. This is similar to how the human system operates. If a person is looking out a car window, s/he will often track a fixed object at which point relatively slowly moving objects will stand out.

The algorithm used to estimate the background motion is to first pick a level in the pyramid, k . The algorithm computes the displacement between frames 1 and 2 at level k . This is a relatively simple step because of the reduced resolution at this level. This computed displacement is a first-order estimate that can then be refined. Frame 2 is now shifted by this displacement but because of the low resolution, will not match exactly with frame 3. The algorithm computes the displacement at level $k-1$ and this process continues it reaches a good estimate. With nearly constant velocity, only two or three iterations have been needed to lock onto the background motion at full image resolution. As pointed out by Ballard and Brown, the foveated objects can then be recognized by dynamic segmentation.

Burt is using a pyramidal architecture that approximates a discrete version of a complex log mapping, $\log(z)$. This architecture allows him to choose the resolution appropriate for the current task, but with that comes the extra burden of always having to choose the correct level to use. This system is not mounted on a movable platform and no real-time results of the overall system were reported.

A fairly common solution to the problem of not having enough resolution and field-of-view at the same time is to use two cameras, one with a large field-of-view, and one with a small one. Burt [12, p. 979] points out that this approach will give trouble when attempting to foveate an object. Suppose

that an interesting feature is seen in the periphery and the system needs to foveate on it. Since only low resolution data is available, a rough estimate can be made to re-aim the camera, but it is possible that the feature will still be outside the fovea. There will be no new information and thus it something of a hit or miss situation. This is opposed to a smoothly space-varying sensor (such as one with a logmap architecture) where after the first movement, even if the feature is not fully foveated, it will fall on a higher resolution portion of the sensor that will enable the system to make a second movement based on more accurate information. The system will quickly converge yielding a well-foveated feature.

1.3 Space-variant images

A space-variant image sensor is characterized by an irregular pixel geometry. We have experimented with several sensor designs, the common features of which are large scale changes between the smallest and largest pixels, wide field of view, and a pixel population far smaller than that of a conventional uniform image sensor. One such space-variant sensor is the complex log mapping, $\log(z + \alpha)$, in which the pixel pattern approximates the sensor geometry of the human eye [42,43].

We often want to display a logmap image on a TV screen, or to compute one from a *TV image*. Thus we can formally define the logmap as a mapping from a TV image, $I(i, j)$, where $i \in \{0, \dots, \text{NROWS} - 1\}$ and $j \in \{0, \dots, \text{NCOLS} - 1\}$. Let $L(u, v)$ be the logmap, with $u \in \{0, \dots, \text{NSPOKES} - 1\}$ and $v \in \{0, \dots, \text{NRINGS} - 1\}$.

The forward mapping from TV image space to logmap space is specified by the *spoke and ring lookup tables*, $S(i, j)$ and $R(i, j)$ (Figure 1.3), where again $i \in \{0, \dots, \text{NROWS} - 1\}$ and $j \in \{0, \dots, \text{NCOLS} - 1\}$. Let $a(u, v)$ be the area (in TV pixels) of a logmap pixel (u, v) .

$$a(u, v) = \sum_{i,j} 1 \mid S(i, j) = u \text{ and } R(i, j) = v. \quad (1)$$

The logmap (or *forward map*) image (Figure 1.4c) is defined by

$$L(u, v) = \frac{1}{a(u, v)} \sum_{i,j} I(i, j) \mid S(i, j) = u \text{ and } R(i, j) = v. \quad (2)$$

The *inverse map*, illustrated in Figure 1.4b, is

$$L^{-1}(i, j) = L(S(i, j), R(i, j)). \quad (3)$$

The relationship between a space-variant image and the lookup tables $S(i, j)$ and $R(i, j)$ is illustrated by comparing Figure 1.3 with Figure 1.4. The values in the lookup tables depict the row and column addresses of pixels in the space-variant image array. We observe that if n is the number pixels for which $a(u, v) > 0$, then $n \leq \text{NSPOKES} \times \text{NRINGS}$, and we define $\text{NPIXELS} = n$.

We present a fast algorithm to compute the logmap from a TV image using $S(i, j)$ and $R(i, j)$ in Section 2.4.

1.4 Space-variant image processing

We are able to perform standard image processing algorithms on logmap images with the use of a *connectivity graph*. We define the connectivity graph, $G = (V, E)$, to be the graph whose vertices, V , stand for sensor pixels and the edges, E , represent the adjacency relations between pixels. Associated with a vertex p is a pixel address (u, v) . Thus we write $(u(p), v(p))$ for a pixel coordinate identified by its graph vertex, or $p = \phi(u, v)$ for a vertex identified by its pixel coordinate. Then, $V = \{p_0, \dots, p_{\text{NPIXELS}-1}\}$.

The definitions of $S(i, j)$ and $R(i, j)$ do not constrain the mapping. Let E_F be the set of edges representing connections between adjacent pixels in the forward map. The edge $(p, q) \in E_F$ provided

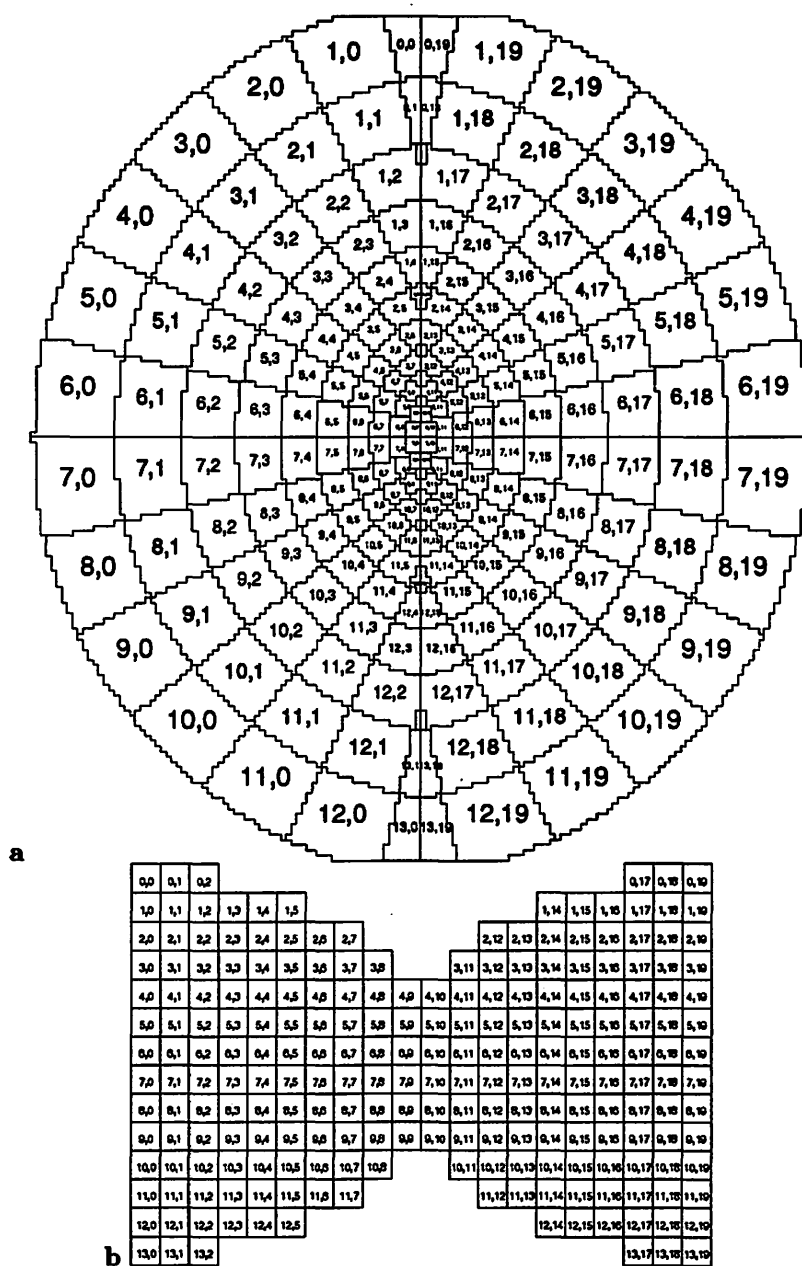


Figure 1.3: a) The lookup tables $S(i, j)$ and $R(i, j)$ combined into one figure. b) The forward map space. The correspondence between inverse and forward map pixels is shown.

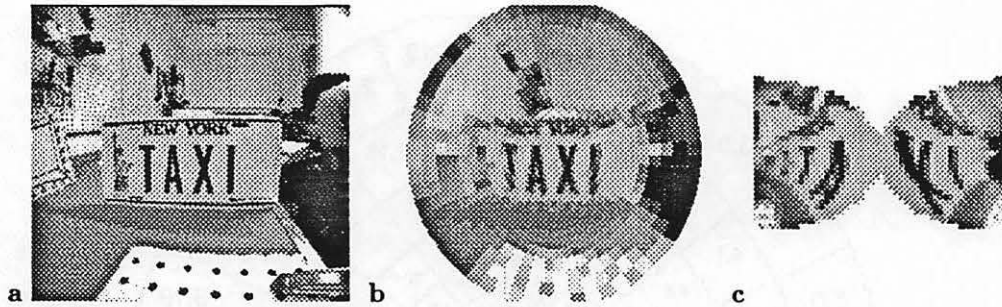


Figure 1.4: A sample of a) a TV image, $I(i, j)$, b) The inverse logmap image $L^{-1}(i, j)$, and c) The forward logmap image $L(u, v)$.

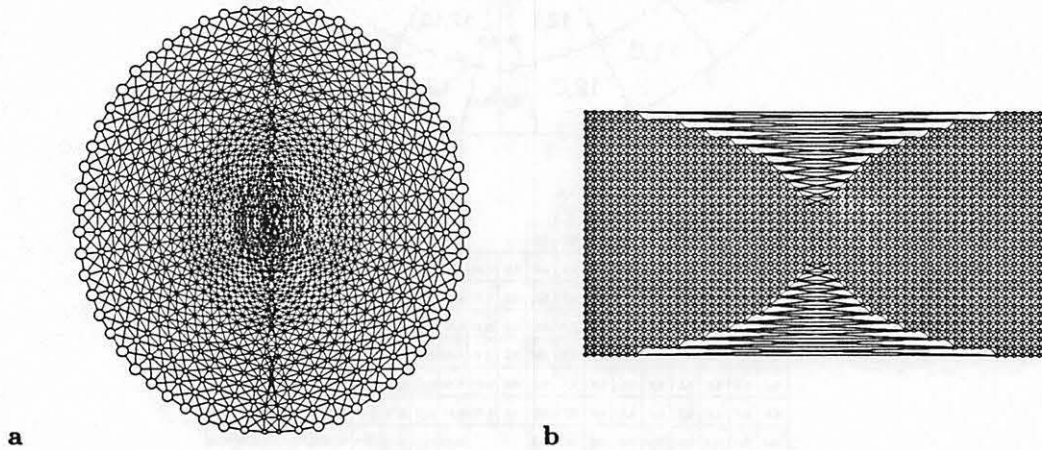


Figure 1.5: The connectivity graph for a $\log(z + \alpha)$ sensor geometry. Circles stand for graph vertices and lines stand for graph edges. The diameter of the circles in the inverse map is proportional to the area of the pixel. The graph explicitly represents the neighbor relations across the vertical meridian. a) The inverse map; and b) The forward map.

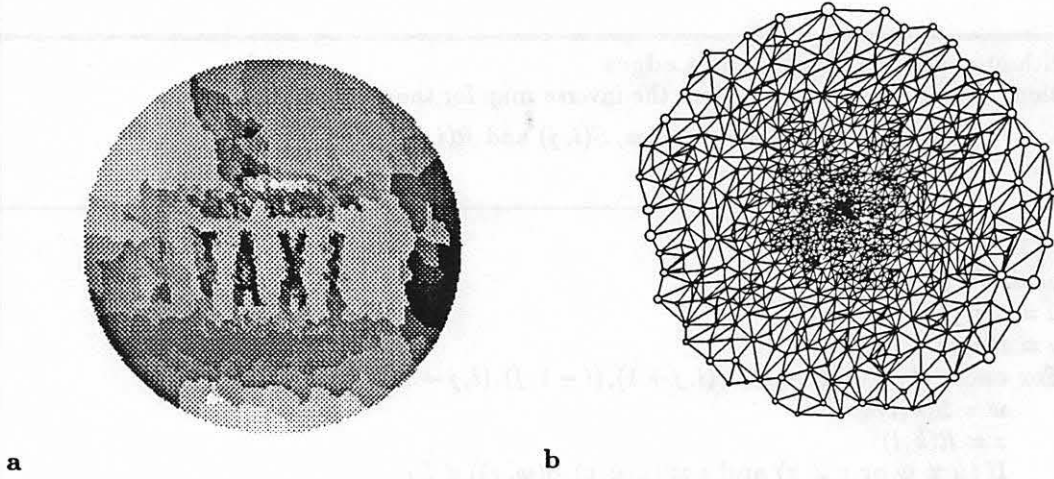


Figure 1.6: a) The image from a sensor having a random pixel geometry. b) The connectivity graph for this sensor.

$(u(p), v(p))$ is a neighbor of $(u(q), v(q))$. For the $\log(z + \alpha)$ mapping (Figure 1.1c) E_F does not contain the connections across the vertical meridian. To obtain all the connections in the sensor, we place the inverse image adjacency relations in the edge set E_I :

$$(p, q) \in E_I \quad \text{provided } \exists i, j, k, l \text{ such that} \\ S(i, j) = u(p), R(i, j) = v(p) \text{ and} \\ S(k, l) = u(q), R(k, l) = v(q) \text{ and} \\ (i, j) \text{ is a neighbor of } (k, l) \quad (4)$$

The connectivity graph edge set, E , may be $E = E_I$ or $E = E_I \cup E_F$, depending on the map. Resulting from the discretization of the complex log function $\log(z + \alpha)$ in the tables $S(i, j)$ and $R(i, j)$, some of the 8-adjacent neighbors in the forward map correspond to nonadjacent pixels in the inverse map. In this case, we chose $E = E_I \cup E_F$.

The algorithm to compute the connectivity graph first allocates one vertex, p , for each pixel (u, v) . The graph edges in E_F are just the 8- or 4- edges from the forward map image. The graph edges in E_I are computed by scanning the lookup tables $S(i, j)$ and $R(i, j)$ with a 2×2 operator. The following algorithm, `compute_inverse_map_edges`, finds the edges E_I for the 4-neighbors.

Algorithm: `compute_inverse_map_edges`

Function: Computes the edges from the inverse map for the connectivity graph.

Input: Spoke and ring look-up-tables, $S(i, j)$ and $R(i, j)$

Output: Inverse map edges, E_I

```

 $E_I = \{\}$ 
For each TV pixel  $(i, j)$ 
     $u = S(i, j)$ 
     $v = R(i, j)$ 
    For each  $(k, l) \in \{(i+1, j), (i, j+1), (i-1, j), (i, j-1)\}$ 
         $w = S(k, l)$ 
         $z = R(k, l)$ 
        If  $(u \neq w \text{ or } v \neq z)$  and  $e = (\phi(u, v), \phi(w, z)) \notin E_I$ 
            Then  $E_I = E_I \cup \{e\}$ .
    End For
End For

```

The `compute_inverse_map_edges` algorithm can be changed in a trivial way to pick up 8-neighbors. Another detail is that some points (i, j) may be outside the domain of the map, and $S(i, j)$ and $R(i, j)$ are undefined at those points. Also, this simplified algorithm does not take into account the fact that not all points (k, l) lie inside the bounds of the TV image.

Each graph vertex, p , in the connectivity graph is represented by a constant size data structure in memory. The graph edges are represented by one field containing a list of pointers. The structure for p may also contain fields such as the pixel array coordinates $(u(p), v(p))$, the pixel area $a(p) = a(u(p), v(p))$, etc. We define the pixel centroid $\mu(p)$ to represent the centroid of the TV pixels that map to p .

$$\mu(p) = \frac{1}{a(p)} \sum_{i,j} (i, j)^T \mid S(i, j) = u(p) \text{ and } R(i, j) = v(p). \quad (5)$$

We define the set of neighbors of p to be denoted by

$$\mathcal{N}(p) = \{q \mid (p, q) \in E\}. \quad (6)$$

The centroid and neighbors of a pixel are useful in writing image processing operations on space-variant images. For example, we can define a simple edge detector as

$$e(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} (L(p) - L(q))^2. \quad (7)$$

Note that the use of the neighbor-list allows us to define image processing operations on pixels having different numbers of neighbors. This eliminates the need for special cases for pixels at the boundary of the image.

We use the pixel centroid, $\mu(p)$, to take into account the relative differences in pixel size. We illustrate this with a plane-fitting edge operator. Given $m = |\mathcal{N}(p)| + 1$ points

$$\{r_0, \dots, r_{m-1}\} = \{p\} \cup \mathcal{N}(p) \quad (8)$$

and their gray values, $L(r)$, we fit a plane $L = \mathbf{a} \cdot (i, j, 1)^T$ to minimize some error measurement. Applying the least squares error criteria for linear equations, we can find a plane by solving for

$$\mathbf{A} = \mathbf{aB} \quad \text{where } \mathbf{A} = (L(r_0), \dots, L(r_m)), \quad \mathbf{B} = \begin{pmatrix} \mu(r_0) & \dots & \mu(r_m) \\ 1 & \dots & 1 \end{pmatrix} \quad (9)$$

The solution of $\mathbf{a}$ for $|\mathcal{N}(p)| \geq 2$ is

$$\mathbf{a} = \mathbf{A}\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1} \quad (10)$$

If not all points are colinear, then $(\mathbf{B}\mathbf{B}^T)^{-1}$ always exists. Since $\mathbf{M} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}$ is constant for each pixel, p , we precompute and store this matrix $\mathbf{M}(p)$ for each graph vertex p . At run time, we only need to compute $\mathbf{A}\mathbf{M}(p)$ for every graph node p .

A simple relaxation procedure, the soap bubble algorithm, is also defined naturally on the connectivity graph. If p is a seed point, then $L_{i+1}(p) = L(p)$, otherwise,

$$L_{i+1}(p) = \frac{1}{1 + |\mathcal{N}(p)|} \left(L(p) + \sum_{q \in \mathcal{N}(p)} L_i(q) \right). \quad (11)$$

The definitions given so far have expressed functions of a pixel and only its immediate neighbors. Sometimes a local operator may map from a larger neighborhood, however. One operation helpful in defining such operations is the $\mathcal{N}_2$ function.

$$\mathcal{N}_2(p) = \{r \mid r \in \mathcal{N}(q) \text{ and } q \in \mathcal{N}(p) \text{ and } r \notin \mathcal{N}(p)\}. \quad (12)$$

We can define a space-variant sensor geometry that is not a complex logarithm, but is the result of a stochastic process. We define this map by selecting seed points randomly from a density function defined so that more points will be in the center of the image. We then create the map by growing each seed point until its neighbor is met using a simple relaxation procedure. The map is called the *random map* and is illustrated in Figure 1.6. To obtain the connectivity graph for this mapping, we use only $E = E_I$.

One application for the random map would be to use it with large area CCD image sensors with defective pixels. Large CCD sensors (1K×1K or 2K×2K pixels) are extremely expensive because of their low production yield. These sensors are much cheaper when they have defective pixels. We could create a random map with the associated connectivity graph for each of these sensors, skipping the bad pixels. This would allow for a very high resolution camera that would not be prohibitively expensive. We also could modify the $\log(z + \alpha)$ mapping to skip bad pixels in such a sensor.

A variety of space-variant sensor geometries have been discussed. Some have been simulated and some have been fabricated [15,42,43,45,46,55]. We can define a connectivity graph for any of these mappings, and then use our image processing library directly on these sensors. The connectivity graph provides a generic approach to image processing on an unconstrained sensor geometry [57].

1.5 System description

The *Cortex-I* (Figure 1.7) integrates a custom miniature logmap camera, *pan-tilt actuator*, controller, general purpose processors, and display. The entire system takes up a volume less than a third of a cubic foot.

The camera (Chapter 2) consists of a miniature commercially available CCD image sensor and a custom lens assembly mounted in the actuator. The camera image is mapped to a logmap image with a fast algorithm described in Section 2.4. Its maximum resolution is 0.175 degrees per pixel and its horizontal field of view measures 33°. The sensor has a fixed focus 4mm lens with a manually changeable aperture (3 sizes). Imaged objects more than 40 mm away from the lens are in focus. The system outputs up to 30 frames per second and measures 256 gray levels per pixel. The camera head (CCD and lens assembly) measures only 8 × 8 × 10 mm. A driver board controls the camera, providing timing signals to the sensor and converting analog sensor data to 8-bit digital data.

A preliminary actuator called the *Platform Pantilt* (Chapter 3) moves a platform by raising and lowering two shafts with linear stepper motors that are attached to the platform with universal joints. This actuator was built as a fall-back position before the Spherical Pointing Motor was operational. The Platform Pantilt measures 7.5 × 7.5 × 12 cm and can move at rotational velocities up to 100°/sec.

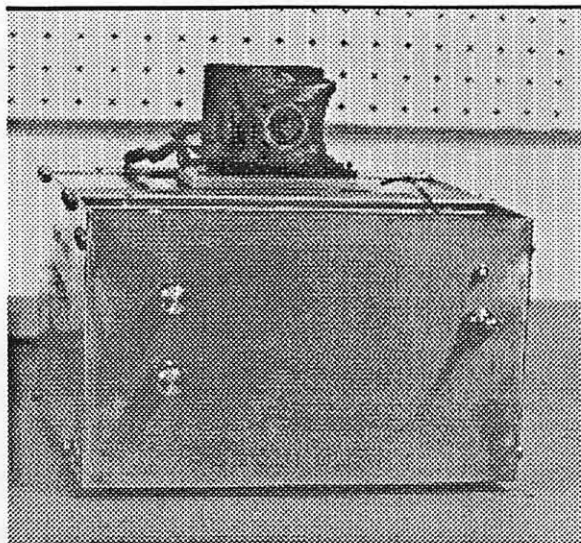


Figure 1.7: **Cortex-I**. Its dimensions are $14 \times 22 \times 22$ cm.

The *Spherical Pointing Motor* (SPM) replaces the Platform Pantilt because of its superior accuracy, speed, and size. The SPM (Chapter 4) is a novel pan-tilt actuator using three orthogonal motor windings to achieve open-loop pan-tilt actuation of the camera sensor in a small, low-power package. The SPM can orient the sensor through approximately 60° pan and tilt, at speeds of several hundred degrees per second. It measures $4 \times 5 \times 6$ cm and its mass is 170 grams.

The controller consists of a camera driver, a 2 MIPS programmable microcontroller (Motorola MC68332), a video display driver, a 12 MIPS digital signal processor (Analog Devices AD2101) DSP that maps the uniform camera image to a space-variant image, and an optional DSP board that acts as a video telephone. The actuator and camera are mounted to the chassis ($14 \times 22 \times 22$ cm) and connected by twisted-pair cables. The prototype is powered from a standard 110 Volt AC line, but uses less than 25 watts and could be battery powered.

The microcontroller controls the SPM and runs the application software. The MC68332 may be connected to a host processor for applications development, to upload sensor data, or to receive pan-tilt commands. The MC68332 is a 32-bit 16 MHz microcontroller integrating peripheral controls, such as programmable digital control lines and timing signals, directly on chip. We connected 192 Kbytes RAM and 128 Kbytes EPROM to the microcontroller. The microcontroller inputs the logmap data via a high speed serial interface from the DSP.

We have implemented image processing demonstrations in the microcontroller ROM. A simple motion tracking program, that turns the camera to center the observed motion field, runs at 16 frames per second (fps). The ROM also contains a test pattern, image binarization (22 fps) and a motor motion demo. We also use the connectivity graph [57], to implement image smoothing (3 fps), edge detection (2 fps) and connected components (1 fps), illustrating both the generality of the programming model and the limitations of the microcontroller in the current prototype.

The Analog Devices 2101 digital signal processor (DSP) reads the uniform CCD image and maps it to a space-variant image. It outputs the logmap over a high-speed serial connection. The DSP board combines an AD 2101 12 MHz DSP having two 2 MHz serial ports, 8 Kbytes internal RAM, 80 Kbytes external RAM, and 64 Kbytes boot EPROM. A second identical DSP board functions as a video display driver, generating an RS-170 video output suitable for display on a standard TV monitor. An optional third DSP board is used as a video transmitter that can send 4 frames per second of logmap images over

a standard voice-grade telephone line.

Several applications using **Cortex-I** are described in Chapter 5.

Chapter 2

Camera

2.1 Introduction

The first problem to solve in building Cortex-I is how to acquire logmap images. We have pursued two routes to solve this problem. The first is to develop a custom VLSI sensor chip with a logmap geometry that will return a logmap image directly. The second is to use a commercially available sensor chip that returns conventional rectangular images and map them into logmap images.

The main advantages of a custom logmap sensor chip are the high frame rate and small resultant system size. This is because the geometry of the chip performs the mapping intrinsically. The sensor layouts that we have designed consist of several thousand pixels on chip, and this is a very small amount of pixels to digitize. It is likely that custom sensor chip could support rate of thousands of frames per second, contingent on the ambient light intensity. However, at the present time, we do not have a fully working space-variant chip.

The main advantages of an off-the-shelf conventional (uniform) sensor chip are low price, high image quality, and immediate availability. Because uniform sensors have been refined for many years and are produced in great quantities, they have extremely high quality which is difficult to match with custom sensor chips. The image from a uniform sensor, however, must be mapped to a logmap image according to the definitions in Section 1.3, and this is computationally expensive since the final logmap image size of thousands of pixels must be supplied through a read-out bottleneck of hundreds of thousands of pixels on the chip. However, with careful design of the readout and logmap emulation algorithms, we have achieved frame rates of up to 30 frames per second, without resorting to excessively complex readout electronics.

We are currently developing a custom logmap sensor chip in collaboration with Synaptics, Inc. The first prototype based on *CMOS* technology has about 500 pixels and does not work to our satisfaction. Figure 2.1 shows some images from this chip. We are currently working on both improving the quality of the image, and increasing the number of pixels.

We also developed a camera based on a uniform sensor chip designed to be mounted on the Spherical Pointing Motor. It uses a low-resolution area CCD sensor, and returns a digital image directly to the computer in our vision system which is then mapped to a logmap image. We made two lenses. One has a fixed field of view (Figure 2.2) and the other has an actuated two position zoom lens (Figure 2.3).

Most other computer vision researchers have used off-the-shelf video cameras for their vision systems. This is certainly a reasonable route to take as it avoids the cost and time involved in developing a custom camera. One group [32,46,55], however, has developed a custom CCD logmap sensor. Like us, they have found it a very time consuming and difficult path. Their current sensor has approximately 2,000 pixels and returns reasonable images.

So why build our own camera? Simply stated, the reason is size. This includes both the size of the

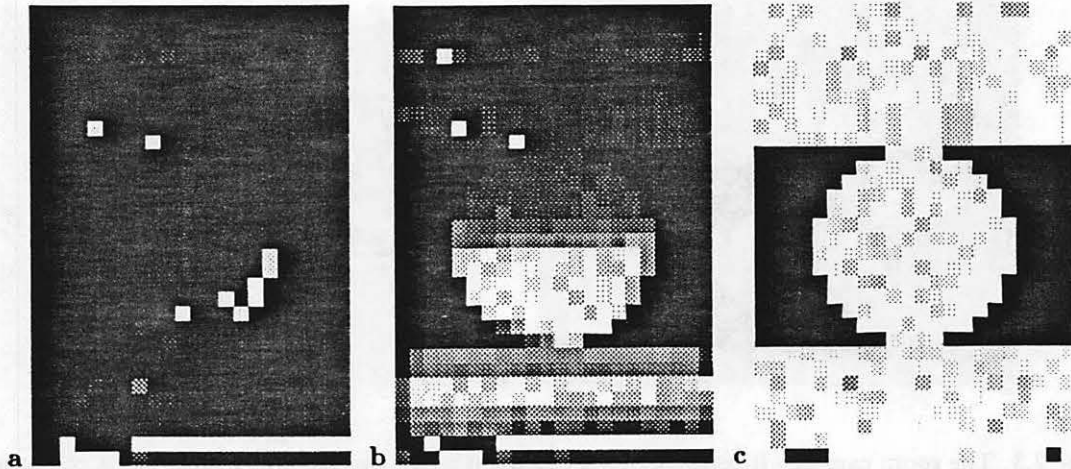


Figure 2.1: Images from our custom logmap sensor chip. They are the logmap images from the chip, and have not been mapped back to scene coordinates. They are scenes of a) an all black scene, b) a horizontal edge with the top half black and the bottom half white, and c) an all white scene. The circle in the center of the images is an artifact of the chip geometry.

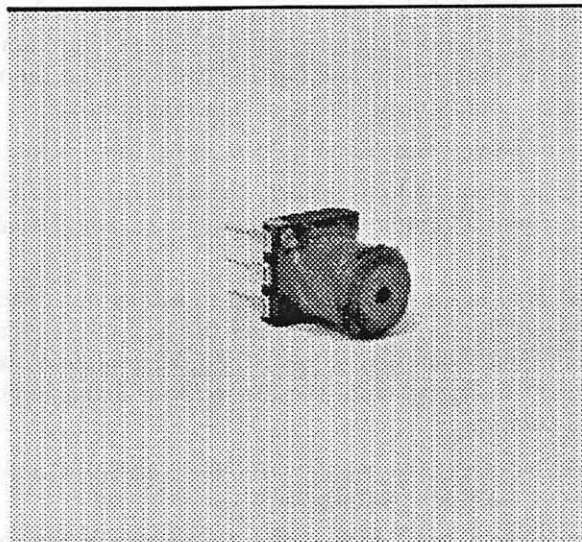


Figure 2.2: The fixed field of view camera. It consists of a four element lens assembly attached directly to a CCD sensor.

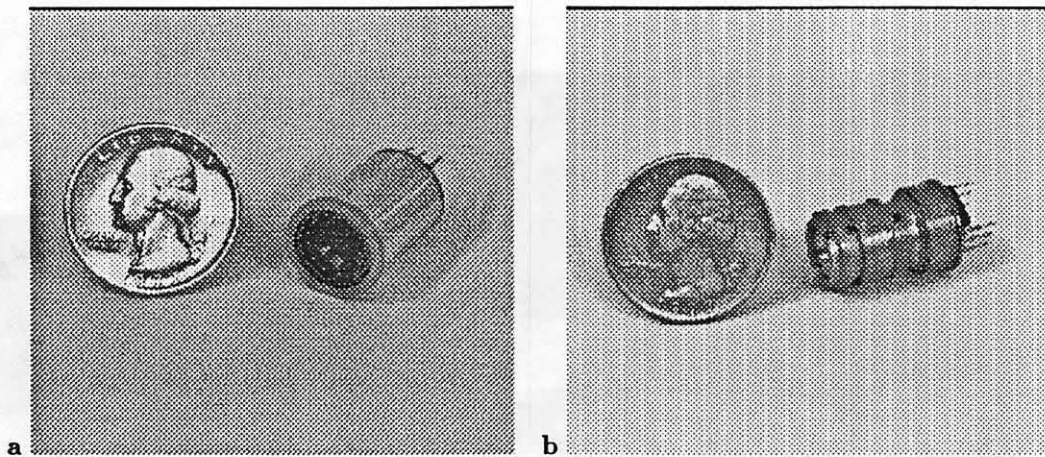


Figure 2.3: The zoom camera. It consists of a six element zoom lens assembly including a coil to actuate the zoom lens. It is attached directly to the CCD sensor. It is shown a) with and b) without the light baffle.

camera head, and the necessary associated readout electronics. Even the smallest commercial cameras available today have camera heads that are several centimeters long and weigh hundreds of grams. To satisfy the goals of this project, we need a camera head that is about one centimeter long and that weighs a couple of tens of grams. Because we are using space-variant images where the high-resolution part of the image is along the optical axis, we can use lenses that have lower quality off the optical axis, as long as the optical quality falls off slower than the resolution of space-variant image. This enables us to make a smaller camera than is available commercially.

In addition, most industrial video cameras output a standard analog RS-170 video signal which entails further electronics to digitize the data and grab the frame. Our camera readout circuitry avoids this entire step by digitizing the sensor data and putting it directly on the processor's data bus.

We designed the camera to fulfill several design criteria. They are:

- The camera head had to be very small and light in order to be actuated by the SPM.
- In order to minimize the associated electronics, the camera should be read out in a digital form suitable for input directly to a computer. By avoiding a standard analog video signal, the system is greatly simplified.
- It must have sufficient field of view and resolution to satisfy our benchmark's requirements. Our chosen problem is to track a moving car and read the license plate on it. The fixed field of view lens does not have sufficient resolution to read the license plate at large enough distances. The zoom lens, however, can be used to track the car in the wide angle position, and read the license plate in the telephoto position.
- Because we use the camera to generate space-variant images, it is not necessary for the lenses we use to maintain high optical quality across the entire field.

In this chapter, we discuss the details of the optics, the electronic readout of the camera, and a fast algorithm to map the uniform image to a logmap image.

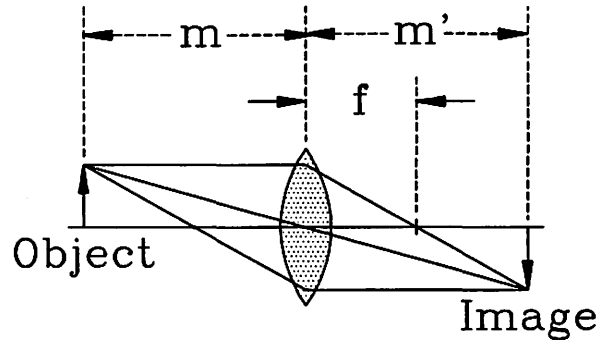


Figure 2.4: Illustration of imaging an object with a simple biconvex lens.

2.2 Camera optics

The optics of a camera consists of the elements necessary to focus the light from an external scene to make an image on a sensor. In our case, this includes the lenses to focus the light, the lens housing to hold the lenses and an actuator to move the zoom lens. Some of the issues in designing such a system are the field of view of the lens, optical quality across the field of view, availability of lenses, and the zoom mechanism. In this section, we examine these issues in the design of our camera.

One of the simplest optics designs consists of a single biconvex lens. Such a system, depicted in Figure 2.4, produces an inverted image at a distance

$$m' = (mf)/(m - f) \quad (13)$$

from the lens [28, P. 46] where f is the focal length of the lens, m is the object distance and m' is the image distance. Since the sensing element is a fixed distance from the lens, we would like m' to be constant. If m is large compared to f , then m' will not vary by much as m varies. By building a camera with a small enough focal length, we can avoid including a focusing mechanism as the image will always be in focus.

For this or any optics system, the system must be designed so that in coordination with the sensor being used, we obtain the desired field of view. The best way to determine the field of view of a camera is to simulate the path of light rays from outside the camera through the lenses to the sensor. Because the surfaces and index of refraction of each lens element is known, it is possible to compute the refractive path of each ray across each surface. This is a process called ray-tracing. Using a commercial ray-tracing package, we determined that a 4 mm focal length lens with our sensor would yield a 33° field of view, which is about what we wanted.

Unfortunately, such a simple optical system has severe chromatic aberrations. This means that the focal length of the lens is different for different wavelengths of light resulting in different image planes for different colors. For us, this would mean an out of focus image since real-world scenes are not monochromatic.

Chromatic aberrations can be corrected by adding more lens elements [28, P. 157–163]. The design of such systems is rather complicated, and rather than reinvent the wheel, we use such a lens system from the eyepiece of a telescope. This lens system, known as an orthoscopic lens assembly, makes up the entirety of our fixed field of view lens. The mechanical drawing of the camera with these lenses is shown in Figure 2.5. This camera head is $8 \times 8 \times 10$ mm and weighs less than 6 grams.

This camera was meant to be used to test our system, and in that it was successful. We were able to test the pan-tilt actuation of the camera, the electronic readout, and image processing in the space-variant domain. The field of view and resolution are not high enough, however, to solve our license plate

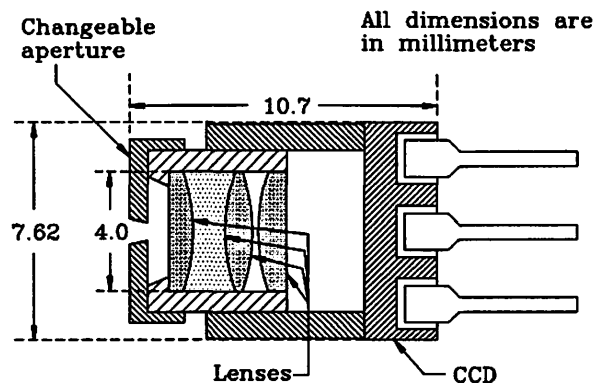


Figure 2.5: Mechanical drawing of the fixed field of view camera. The 4 element lens is a 4 mm orthoscopic lens that is corrected for chromatic aberrations.

problem well. When the license plate is far enough away to track its motion (i.e., when the entire license plate is in the field of view), there is barely enough resolution to read the characters on it. We could solve this problem by using either a higher resolution sensor or having a lens with controllable field of view. We chose the latter.

2.2.1 Zoom lens

We designed a zoom lens capable of operating between a 4 and 13 mm focal length. This corresponds to a 33° and a 10° field of view, respectively, which is sufficient to solve our license plate problem. The lens is actuated between positions with a *solenoid*. Some of the issues in designing this zoom lens are its focusing, the lens availability, the brightness across the field of view, and the mechanical assembly and actuation.

The simplest zoom lens can be built by just moving a single lens, such as the one depicted in Figure 2.5, between the object and the image. The image will change magnification, but the focal plane of the image will move significantly, according to Equation 13. The situation can be improved by adding two lenses (Figure 2.6). Here, as the middle negative lens moves between the two outer positive lenses, the overall focal length changes, and while the focal plane still moves, its motion is greatly reduced. The focal plane is at the same position for exactly two different zoom positions. This can be adjusted so that the two positions are the minimum and maximum zoom, which is what we use since we can actuate it only to these two positions. This type of zoom lens system is examined in great detail by Bergstein [9,10]. Our zoom lens system is based on that in Figure 2.6 with the modification that the last lens is replaced with the orthoscopic lens of the fixed field of view camera to improve the chromatic aberrations. In addition, an internal aperture is added to limit light reflecting off the walls of the lens housing.

In designing the zoom lens, we were limited by the availability of commercial lenses. Because of the extreme difficulty of finding very small lenses meeting our specifications, we had to build the zoom lens somewhat larger than we had anticipated. One solution to this problem is the custom manufacturing of small lenses ourselves. We are able to make plano-concave and biconcave lenses using optical quality epoxy with a release agent so the epoxy does not stick to the mold. We pour the epoxy into brass tubing and place metal spheres of the appropriate diameter on the ends of the tubing. The epoxy dries and we remove the spheres, leaving a lens inside the tubing. We have made some prototype lenses using this technique and are in the process of making a new zoom lens assembly with them.

One problem with multiple element lenses such as the zoom lens is internal reflection of light off of the inside of the lens housing. This occurs most often with light entering the lens assembly at large

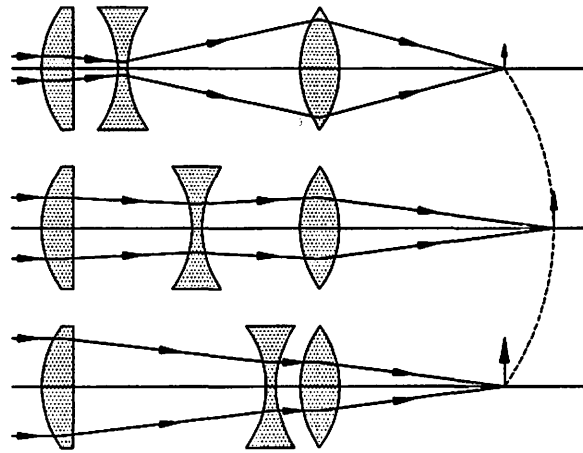


Figure 2.6: Illustration of zoom lens. The image plane moves as the magnification of the lens changes. (Adapted from [51, P. 275])

angles. This can be greatly reduced by inserting an aperture stop in the lens between lens elements. This is a disk of metal that blocks all light rays very near the edge of the lenses where most of the unwanted light is.

The aperture stop increases the vignetting of the lens, which is the reduced illumination of the image plane in the periphery of the image. This is because light entering the lens assembly at large angles gets cut off by the stop that otherwise would end up in the periphery of the image. In our case, the vignetting is not very severe, and can be compensated for in software.

The mechanical assembly holding the lenses had to be designed to allow a very small force to actuate the central lens element while not allowing light to leak into the system. At the same time, the lenses must all be adjustable so the system can be focused. A mechanical drawing of the assembly we designed is in Figure 2.7. We mounted the two outer lenses on threads so they are easily focusable. The central lens slides inside the lens housing, attached to two pins that connect to a small cylinder outside the housing. This lens is actuated by applying current to a coil of wire wrapped around it. There are two press-fit rings that act as stops to control the limits of motion of the central lens. Two pieces of polarized plastic are put in the lens assembly, one fixed and one rotatable, to control the light intensity. A thin cylindrical baffle fits over the entire assembly to block light from leaking in the slots. The zoom lens is pictured with and without the baffle in Figure 2.3.

The zoom lens is actuated with a small solenoid. A coil of wire is wound around the small cylindrical sheath attached to the zoom lens by pins. Because the camera is mounted directly against the pole of the permanent magnet in the rotor of the SPM, applying current to the coil results in a force on the zoom lens in either direction depending on the polarity of the current. Although the zoom lens is moderately well-focused throughout its range, we only have the capability of actuating it to two positions, one at each end of its range of motion.

Images taken using the zoom lens are in Figure 2.8.

2.3 Image sensor

In designing our camera, we had to choose between two image sensor technologies, CMOS and CCD. They are both silicon-based VLSI technologies, and both are able to transduce light to voltages. In this section, we discuss the basics of these technologies and explain the details of the hardware and software

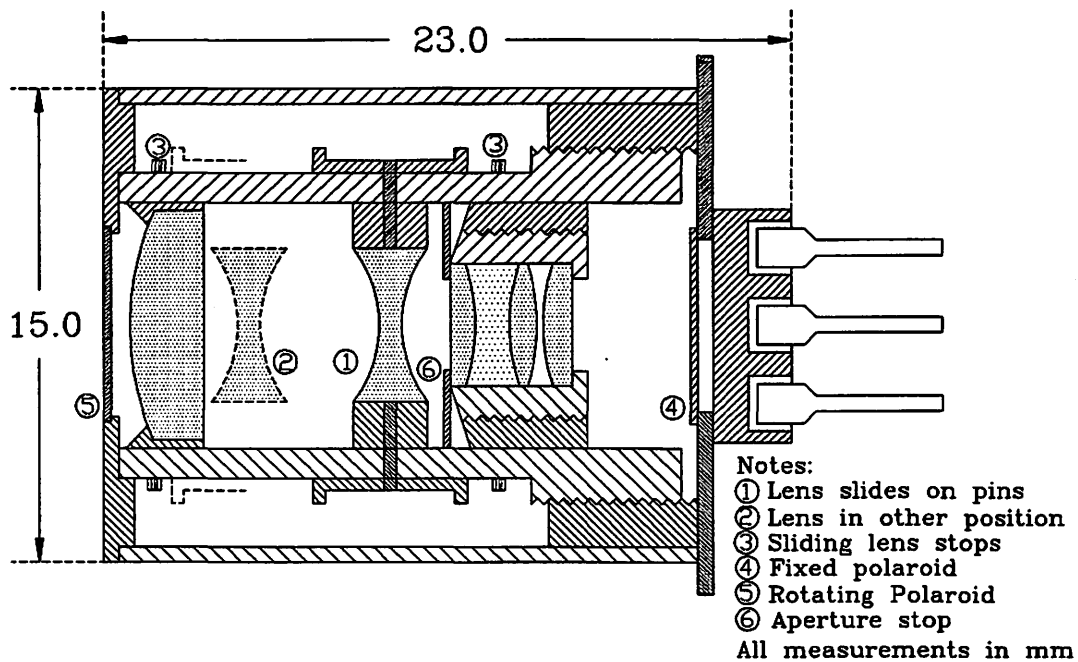


Figure 2.7: Mechanical drawing of the zoom camera.

we developed to use the sensor.

2.3.1 Sensor technologies

CMOS image sensors are based on bipolar junction transistors (BJT's) which are inherently light sensitive (Figure 2.9). If the base of a BJT is not connected, the current through the transistor will be proportional to the light incident on it. We can increase the dynamic range of the sensor by taking the logarithm of this current. We take this logarithm by connecting two field effect transistors to the BJT [35]. The chip we are making with Synaptics uses this approach.

A CMOS image sensor consists of an array of VBJT's with an addressing scheme that allows each transistor to be accessed directly. The result of this scheme is that an image frame may be read from the sensor by scanning through each transistor sequentially. The next frame can be read directly without delay. The maximum frame rate is determined by the speed with which each transistor can be addressed. CMOS image sensors are usually fast and easy to use, but they are relatively large. The smallest pixel in our current design is 10 microns.

CCD image sensors, on the other hand are denser with pixel sizes down to 5 microns, resulting in their more common usage. They work by accumulating charge, proportional to the incident light, in an electronic "bucket". Each bucket must accumulate (or integrate) charge before it can be read out. This results in a different usage than CMOS sensors. Between each frame read out, the CCD must integrate light. Thus, the maximum frame rate is determined not only by the readout speed (or scan time), but also by the integration time.

CMOS sensors are usually capable of addressing each pixel individually. CCD sensors, however, work by shifting the contents of each row of pixels vertically through the other pixels to the edge of the array where they are either read out through a horizontal shift register, or stored in a frame buffer on chip for future readout. During this shifting process, as the image is passed through every pixel in its column,

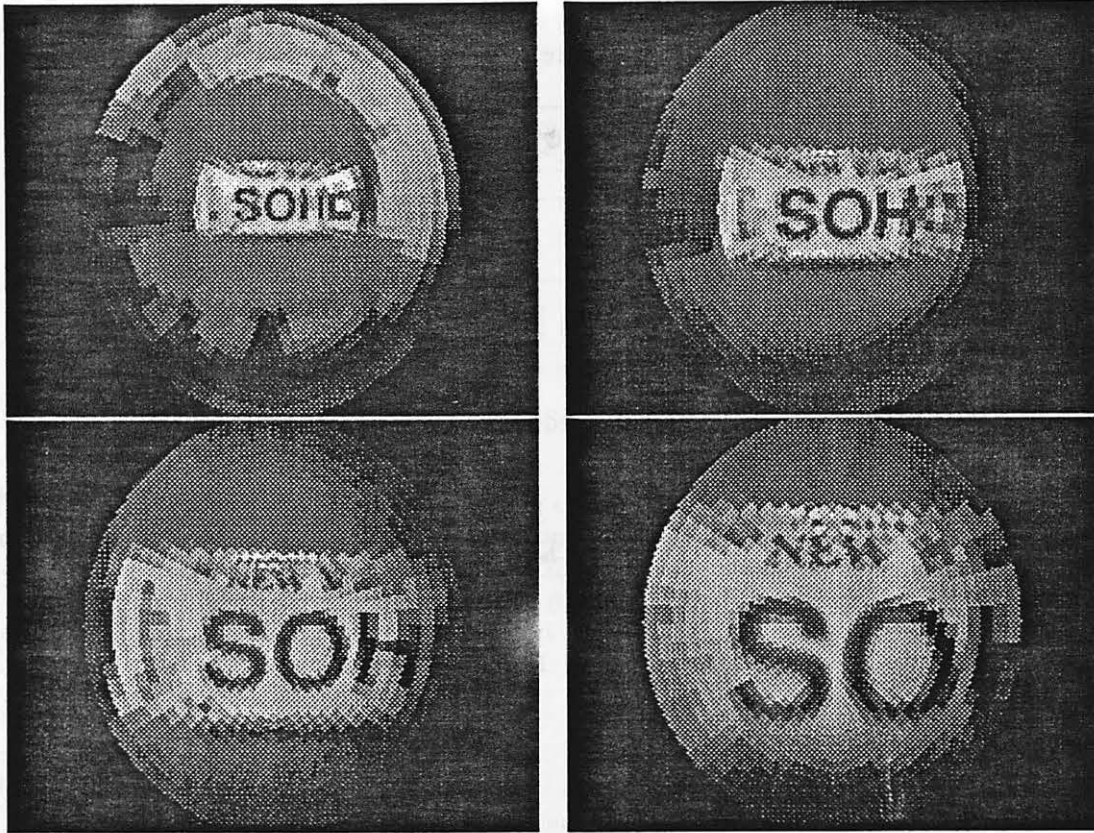


Figure 2.8: The same scene imaged with the zoom lens in four different positions (manually actuated).

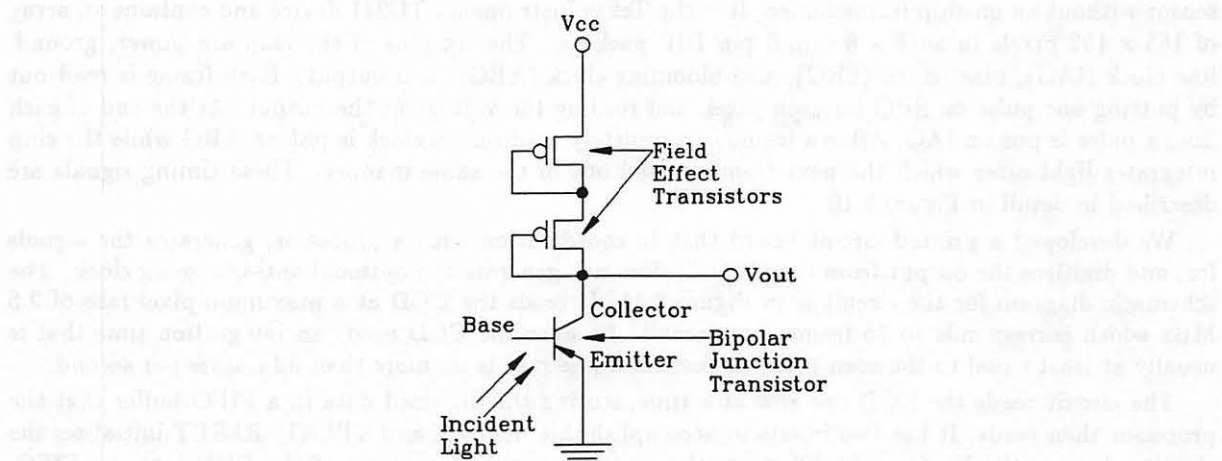


Figure 2.9: A vertical bipolar junction transistor hooked up with two field effect transistors to produce a photo cell with a logarithmic output.

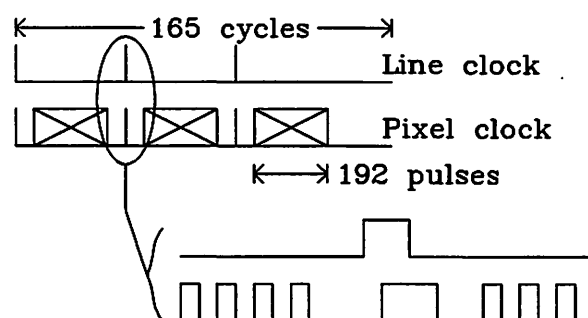


Figure 2.10: Timing signals used to control the CCD image sensor.

the sensor continues to integrate light. This results in *vertical smearing* where each pixel has an influence on every other pixel in its column, where the amount of influence is proportional to the ratio of scan time to integration time. Because of this, some CCD sensor chips have a frame buffer that is not light sensitive to which the image is shifted very quickly. The image can then be read from this frame buffer relatively slowly without affecting image quality. In this case, there is still some vertical smearing, but it is greatly reduced because the transfer time to the frame buffer is very small.

2.3.2 Timing signals

A CCD needs many different signals to control its readout and light integration. As already stated, each pixel in the CCD consists of an electronic “bucket” accumulating charge. If the light intensity is too great, or the integration time is too long, the bucket can overflow into its neighbors. Thus, one pixel can influence its neighbors. This is called blooming. Many CCD sensors have an antiblooming control where the top portion of the bucket is discarded at regular intervals. This eliminates blooming for all but the most intense illuminations, or longest integration times.

We chose the smallest rectangular image sensing device commercially available, which is a CCD sensor without an on-chip frame buffer. It is the Texas Instruments TC211 device and contains an array of 165×192 pixels in an 8×8 mm 6 pin DIP package. The six pins of the chip are power, ground, line clock (IAG), pixel clock (SRG), anti-blooming clock (ABG), and output. Each frame is read out by putting one pulse on SRG for each pixel, and reading the voltage at the output. At the end of each line, a pulse is put on IAG. After a frame is completely read out, a clock is put on ABG while the chip integrates light after which the next frame is read out in the same manner. These timing signals are described in detail in Figure 2.10.

We developed a printed circuit board that in coordination with a processor, generates the signals for, and digitizes the output from the CCD. It does not generate the optional anti-blooming clock. The schematic diagram for the circuit is in Figure 2.11. It reads the CCD at a maximum pixel rate of 2.5 MHz which corresponds to 75 frames per second. Because the CCD needs an integration time that is usually at least equal to the scan time, the usable frame rate is no more than 32 frames per second.

The circuit reads the CCD one row at a time, storing the digitized data in a FIFO buffer that the processor then reads. It has two inputs to accomplish this, RESET and START. RESET initializes the circuit, clearing the FIFO. START starts the circuit readout of one row of the CCD into the FIFO. Because the FIFO is dual-ported, after one pixel time, the processor can start reading that row from the FIFO. The CCD automatically shifts each row down one row when IAG is pulsed, so each time the processor reads a row, it gets the next row from the CCD. The circuit does not do anything at the frame level. Rather, it is up to the processor to read the correct number of rows and then let the CCD integrate for a fixed amount of time before reading the next frame. A description of the system architecture and

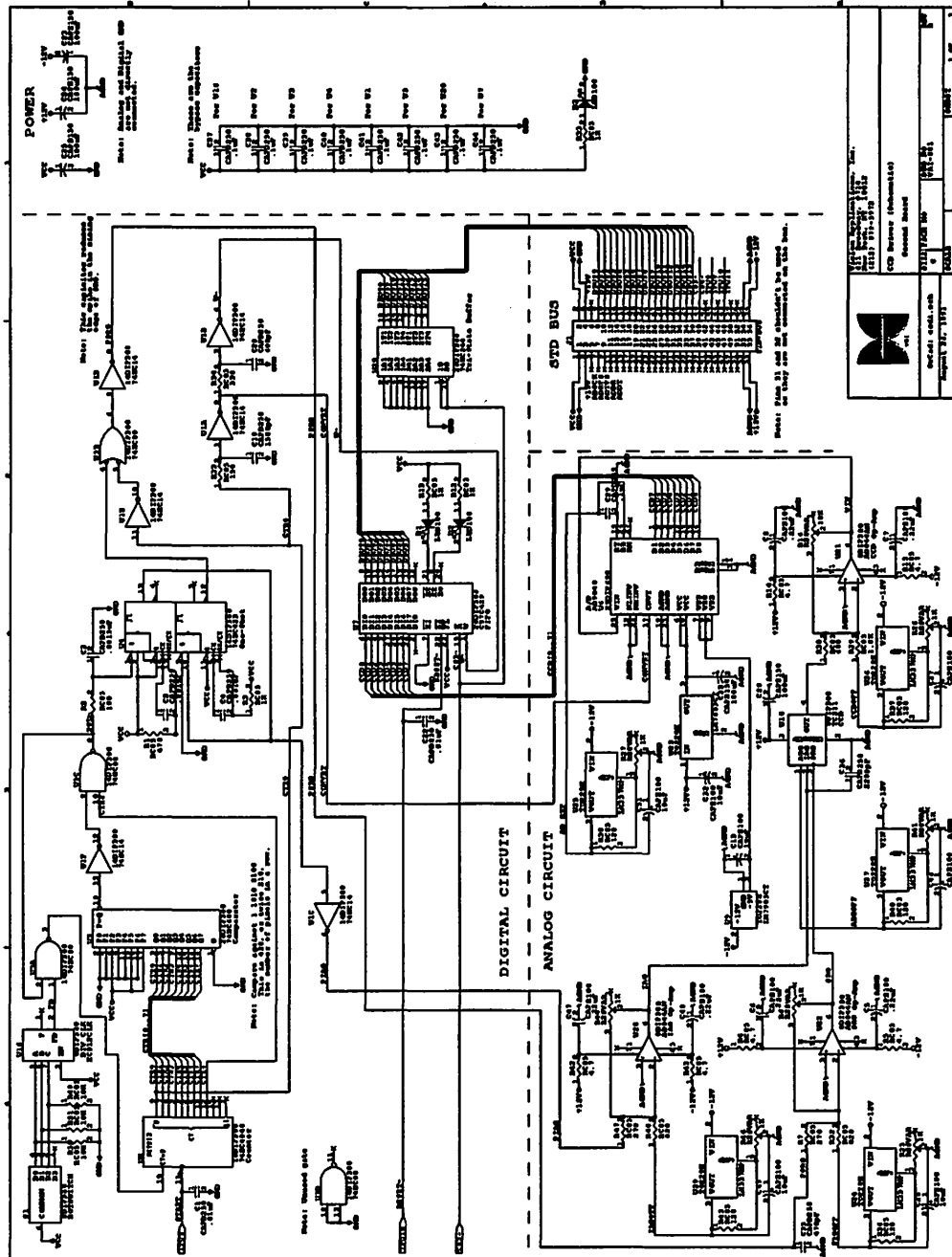


Figure 2.11: Schematic diagram for CCD driver board.

the processors used to read from the CCD driver board can be found in Chapter 5.

The circuit is divided into a digital and an analog section. The above mentioned part of the circuit is all digital. The analog part converts the signals to the necessary levels, and amplifies and digitizes the CCD output to be stored in the FIFO.

2.3.3 Vertical smearing

The CCD we use has a significant image degradation problem called vertical smearing. The processor must read the CCD as quickly as possible because the CCD continues to integrate light as the image is shifted out of the chip. Each pixel that is read out actually consists of itself plus a fraction of all other pixels in that column. The vertical smearing increases with the ratio of scan time to integration time.

We model the vertical smearing exactly for static scenes with a linear algebraic system, enabling us to eliminate the smearing in software. We use the definitions of TV and logmap images of Section 1.3. Because each pixel read out is a linear combination of the original pixel and all the other pixels in that column, the original pixel value can be retrieved by solving a set of linear equations. Let us call the pixel that is read out from the CCD $I'(i, j)$ and the actual pixel value before it is read out $I(i, j)$ where $i \in \{0, \dots, \text{NROWS} - 1\}$ and $j \in \{0, \dots, \text{NCOLS} - 1\}$. Then for each column, j , we have a set of pixels, $I'(i, j)$, and want to calculate the actual values, $I(i, j)$. We say the integration to scan-time ratio is $\text{INT_SCAN} : 1$. This means that a read out pixel, $I'(i, j)$, contains $\text{INT_SCAN}/(\text{INT_SCAN} + 1)$ of the actual pixel value, $I(i, j)$, and $1/(\text{INT_SCAN} + 1)$ of all the other pixels in the column. We can write this as

$$I'(i, j) = \frac{\text{INT_SCAN}}{\text{INT_SCAN} + 1} I(i, j) + \frac{1}{\text{NROWS}(\text{INT_SCAN} + 1)} \sum_{k=0}^{\text{NROWS}-1} I(k, j) \quad (14)$$

We must solve these NROWS linear equations in NROWS unknowns for the actual pixel values, $I(i, j)$. This must be done separately for each column. This can be done off-line and then calculating each actual pixel, $I(i, j)$, would involve NROWS multiplications and NROWS additions. Because we are interested in logmap images that are derived from the original TV image, we do not want to desmear the original TV image. Instead, we examine the vertical smearing problem in the logmap domain where we find an approximate solution.

In logmap images, each pixel is the average of many TV image pixels and these pixels come from multiple columns. Again, we want to find the actual logmap pixel value (which is the average of actual TV pixel values) from the read out logmap pixel values (which are the averages of readout TV pixel values), and we want to do this without referring to the TV pixels themselves. We call a read out logmap pixel, $L'(u, v)$, and an actual logmap pixel, $L(u, v)$, where $u \in \{0, \dots, \text{NSPOKES} - 1\}$ and $v \in \{0, \dots, \text{NRINGS} - 1\}$. These are defined by Equation 2 of Section 1.3. Now we can solve for the actual value of a logmap pixel analogously to rectangular pixels. From Equation 14, we write

$$L'(u, v) = \frac{1}{a(u, v)} \sum_{i,j} \left(\frac{\text{INT_SCAN}}{\text{INT_SCAN} + 1} I(i, j) + \frac{1}{\text{NROWS}(\text{INT_SCAN} + 1)} \sum_{k=0}^{\text{NROWS}-1} I(k, j) \right) | \quad (15)$$

$S(i, j) = u \text{ and } R(i, j) = v$

$$L'(u, v) = \frac{\text{INT_SCAN}}{(\text{INT_SCAN} + 1)a(u, v)} \sum_{i,j} I(i, j) | (S(i, j) = u \text{ and } R(i, j) = v) + \frac{1}{\text{NROWS}(\text{INT_SCAN} + 1)a(u, v)} \sum_{i,j} \sum_{k=0}^{\text{NROWS}-1} I(k, j) | \quad (16)$$

$(S(i, j) = u \text{ and } R(i, j) = v)$

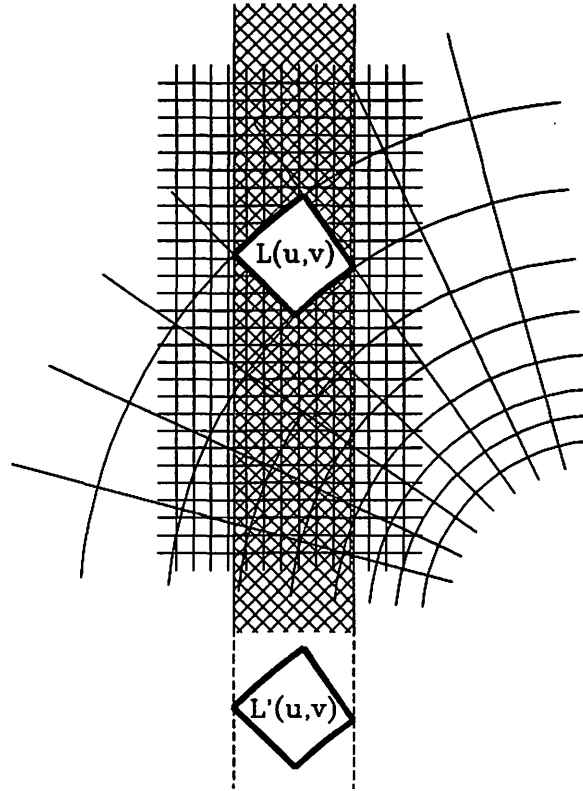


Figure 2.12: Elimination of vertical smearing. Actual logmap pixel $L(u, v)$ is calculated from read out logmap pixel $L'(u, v)$ and a weighted sum of all other logmap pixels sharing TV pixels in the same columns.

$$L'(u, v) = \frac{\text{INT_SCAN}}{\text{INT_SCAN} + 1} L(u, v) + \frac{1}{\text{NROWS}(\text{INT_SCAN} + 1) a(u, v)} \sum_{i,j} \sum_{k=0}^{\text{NROWS}-1} I(k, j) | (S(i, j) = u \text{ and } R(i, j) = v) \quad (17)$$

The second term of Equation 17 represents the sum of TV pixels in the same column as the logmap pixel $L(u, v)$. This is represented in Figure 2.12. This can not be calculated in terms of logmap pixels exactly because some logmap pixels have only some of their TV pixels in the same column as $L(u, v)$. In addition, each TV column is weighted differently depending on the number of TV pixels in a column of $L(u, v)$.

We approximate Equation 17 using only logmap pixels. We use weighted logmap pixels where the weight of each pixel is the total number of TV pixels from that logmap pixel corresponding to TV pixels in $L(u, v)$. We define the weights, $W(u, v, w, x)$, for each logmap pixel, $L(u, v)$, using the total weight, $W_{TOT}(u, v)$, to scale each logmap pixel as defined in Equations 18 and 20. Note that the definition for $W_{TOT}(u, v)$ counts only the TV pixels in a column that map to $L(u, v)$.

$$W_{TOT}(u, v) = \sum_{i,j} \sum_{k=0}^{NROWS-1} 1 \mid S(i, j) = u \text{ and } R(i, j) = v \text{ and } S(k, j) \neq \text{invalid} \quad (18)$$

$$W(u, v, w, x) = \sum_{i,j,k} 1 \mid S(i, j) = u \text{ and } R(i, j) = v \text{ and } S(k, j) = w \text{ and } R(k, j) = x \quad (19)$$

$$W(u, v, w, x) = \sum_{i,j,k} 1 \mid S(i, j) = u \text{ and } R(i, j) = v \text{ and } S(k, j) = w \text{ and } R(k, j) = x \quad (20)$$

We can rewrite Equation 17 using this approximation to get

$$L'(u, v) = \frac{INT_SCAN}{INT_SCAN + 1} L(u, v) + \frac{1}{(INT_SCAN + 1) W_{TOT}(u, v)} \times \sum_{k=0}^{NROWS-1} \sum_{l=0}^{NCOLS-1} W(u, v, k, l) \cdot L(k, l) \quad (21)$$

The result of this approximation is that pixels are smeared horizontally no more than the width of any logmap pixel sharing TV pixels in the same column. This is because logmap pixels sharing TV pixels in the same column may extend to other columns not covered by the original logmap pixel. This, however, is a large improvement over the original vertical smearing that extended over the full height of each column.

The implementation of this desmearing approach is complicated by the use of the logmap. In the rectangular case, each column of TV pixels can be solved independently because each TV pixel in a column depends only on other TV pixels in the same column. Logmap images, on the other hand, have no columns. Each logmap pixel depends on the set of logmap pixels sharing TV pixels in the same column as itself. The other logmap pixels in this set, however, do not generally depend on the same set of pixels. This means that there is no way to solve the desmearing problem for any subset of the image. Instead, the entire image must be desmeared in one step. Referring to Equation 21, this involves solving NPIXELS linear equations with NPIXELS unknowns where NPIXELS is the number of pixels in the logmap image.

This at first may seem a daunting task, but the matrix to solve is sparse. For our logmap, there are 1,376 pixels. The number of pixels that each pixel depends on varies with the particular pixel, but varies from 12 to 246 with a mean of 86. This means that an average row in the matrix of linear equations will have only 86/1376 or 6% non-zero entries.

An image can be desmeared by taking advantage of the sparseness of the matrix and by doing most of the computation off-line. Equation 21 can be rewritten as a matrix problem, $\mathcal{A}\mathcal{X} = b$, where $\mathcal{X}$ is the unknown actual logmap pixel values ($L(u, v)$ in a 1D vector form), b is the known read-out logmap pixel values (L' in a 1D vector form), and $\mathcal{A}$ is the matrix of coefficients multiplying $\mathcal{X}$. We first invert $\mathcal{A}$ off-line using sparse matrix inversion techniques ([39, P. 72-81]). To desmear a given image, we vectorize it into b , then just compute the result, $\mathcal{X} = \mathcal{A}^{-1}b$. This still requires a large amount of computation. Fortunately, $\mathcal{A}^{-1}$ is also sparse and thus b only needs to be multiplied with non-zero entries of $\mathcal{A}^{-1}$. This can be sped up further by eliminating near-zero values of $\mathcal{A}^{-1}$. Figure 2.13 shows the result of applying these desmearing techniques to an image taken with our camera. Results are shown using the full $\mathcal{A}^{-1}$, using only values of $\mathcal{A}^{-1}$ whose absolute values are greater than 0.001, and greater than 0.01. Severe degradation along the vertical meridian results for the last case. The vertical meridian is the most sensitive area because here, there are the most pixels in a column. Because each pixel depends on so many other pixels, their weights are small. Results showing timing and the number of elements in $\mathcal{A}^{-1}$ used are shown here where the execution time is for a SUN Sparc 2.

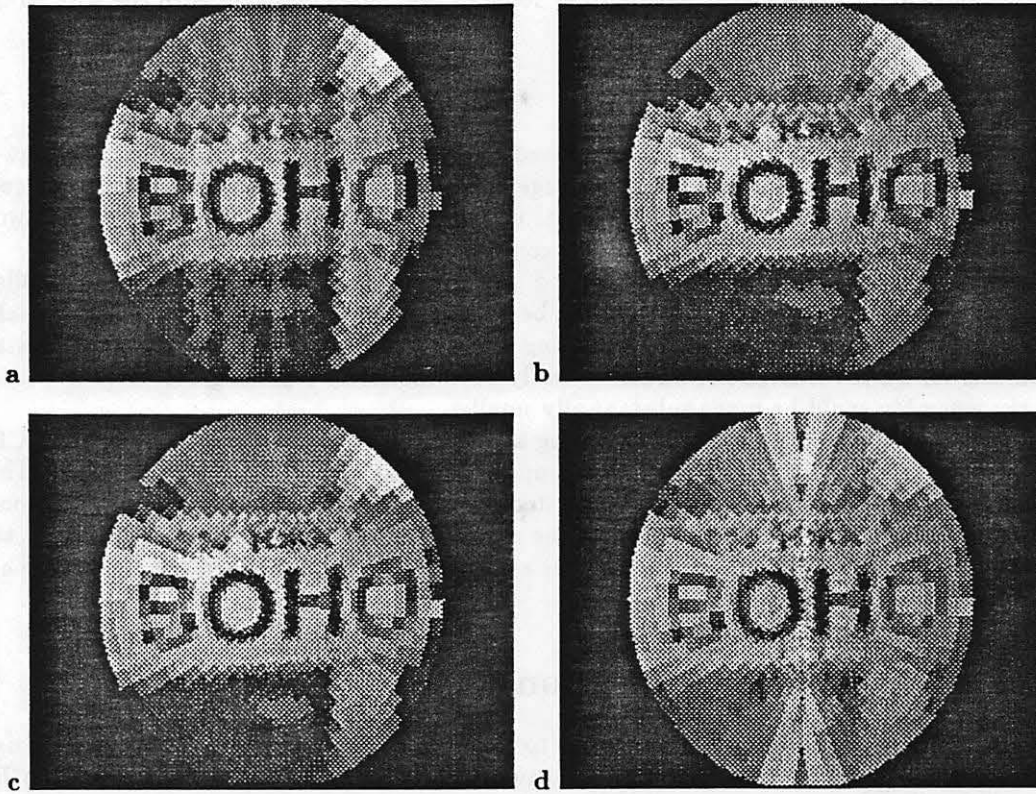


Figure 2.13: a) Image from miniature camera with vertical smearing; b) desmeared image; c) desmeared image using only $|A^{-1}| > 0.001$; d) using only $|A^{-1}| > 0.01$.

A^{-1} cutoff	% of A^{-1} used	Execution time
—	30%	1.36 seconds
.001	8%	0.58 seconds
.01	3%	0.19 seconds

2.3.4 Electrical noise

One of the biggest difficulties in obtaining a high quality image from the sensor is electrical noise. Noise comes from two main sources, electromagnetic interference (EMI) and noise in the readout circuit due to high frequency digital circuitry. The noise in the circuit can be greatly reduced through standard noise reduction techniques [25], such as using separate power supplies for the digital and analog parts of the circuit, using ground planes in the circuit, and placing related parts near each other.

EMI is especially bad in our case because the camera is mounted in the SPM approximately 20 cm from the readout circuit. To make matters worse, extremely thin wires (40 gauge) must be used to connect to the CCD in the SPM. This is because when the SPM moves the camera, it must also bend the wires attached to the camera. The connecting wires must be highly compliant in order to maximize the accuracy of the motor. The very thin wire is used for approximately 5 cm. We reduced the EMI by using twisted pair cables for the connections to the CCD. This entails twisting each signal wire with a ground wire.

In addition to these design techniques, we designed and had manufactured a printed circuit board for the readout circuit. This board substantially reduced the noise compared with the wire wrap boards we were using before.

2.3.5 Future Designs

In the future, the camera would be most improved by using a higher resolution image sensor. Because a higher resolution sensor will very likely be larger physically, the optics would have to be redesigned because the field of view and the off-axis optical quality would have to be increased. This would also necessitate using more computational power to compute the larger logmap.

The zoom lens could be further miniaturized. Its current large size is due to the unavailability of certain small lenses. In the future, we expect to be able to manufacture any lens we need ourselves with epoxy as discussed. In addition, the lens housing could be miniaturized. We made the lens assembly with threads for all the lenses so it would be easily adjustable. By replacing the threads with press fit joints, the assembly could be made substantially smaller.

The image quality could be improved by using a differential transmission scheme for the CCD output signal. A surface mount differential output op-amp would be connected directly to the CCD. This makes positive and negative outputs that would be twisted together. The circuit board would then convert this back to a regular signal for use by the rest of the circuit. The advantage of this technique is that both the positive and negative signals are affected the same way by EMI and when the signals are combined back together, the noise is subtracted out.

2.4 Fast logmap computation

The final step in using the miniature camera is to map the uniform output image into a logmap image using the look-up-tables $S(i, j)$ and $R(i, j)$ of Section 1.3. This can be done by averaging the TV pixels that map to each logmap pixel as specified by $S(i, j)$ and $R(i, j)$. This approach, however, is relatively slow requiring several instructions per TV pixel.

We have developed a fast algorithm to compute the logmap based on a run-length mapping¹. This algorithm avoids accessing the look-up-tables for each pixel and it executes with an average of just over one instruction per TV pixel. The $S(i, j)$ and $R(i, j)$ look-up-tables are first run-length encoded off-line with the algorithm `run_length_encode`. The CCD image is then mapped, line by line, as it is read out into the logmap image with the algorithm `fastmap`.

The algorithm `run_length_encode` analyzes $S(i, j)$ and $R(i, j)$ in raster order to find the *runs*, or sequences of length ℓ in which

$$S(i, j) = \dots = S(i, j + \ell - 1)$$

and

$$R(i, j) = \dots = R(i, j + \ell - 1).$$

We partition the raster scan of the TV image into a sequence of NRUNS runs where `run.len[k]` is the length of the k th run and `run.addr[k]` is the address of the logmap pixel of the k th run.

The algorithm `fastmap` first zeros the logmap. It then goes through the TV image in raster order and through the runs in linear order. For each run, k , it adds `run.len[k]` TV pixels to the logmap pixel indexed by `run.addr[k]`. After the entire image has been mapped, the resulting logmap image must be normalized. For each logmap pixel, $L(u, v)$, we set $L(u, v) = L(u, v)/a(u, v)$. `fastmap` shows the pixels in each run being summed in a loop. However, in practice we avoid the overhead associated with each loop iteration by unrolling the loops. We call a routine, `mapn` which consists of n inline summing statements of the form `*lm.addr += I(tv.index++)`.

¹Patent pending.

The number of runs, $NRUNS$, depends on the mapping function given by the look-up-tables $S(i, j)$ and $R(i, j)$. For a wide variety of interesting mappings, such as the logmap, $NRUNS \ll NROWS \times NCOLS$, and $NSPOKES \times NRINGS \ll NROWS \times NCOLS$. Let us assume that the statements in `fastmap` that add the TV pixel to the logmap pixel and increment the TV pixel address takes one assembly language instruction. Let us also assume that we either unroll the loop computing each run (or use a DSP processor with zero-overhead loops) so that there is no overhead during each run. If the overhead between each run takes b instructions, then the mapping will take $NROWS \times NCOLS + b \times NRUNS + 2 \times NSPOKES \times NRINGS$ instructions, or $1 + \frac{b \times NRUNS}{NROWS \times NCOLS} + \frac{2 \times NSPOKES \times NRINGS}{NROWS \times NCOLS}$ instructions per TV pixel.

Our system uses a TV image of $165 \times 192 = 34,650$ pixels and there are 6,074 runs. The logmap image is $34 \times 56 = 1904$ pixels, and the number of "overhead" instructions between each run, b , is 4. Then the mapping takes 1.81 instructions per TV pixel.

Algorithm: `run_length_encode`

Function: This creates the run length and address tables for a mapping based on the spoke and ring look-up-tables.

Input: $S(i, j)$ and $R(i, j)$ look-up-tables

Output: Run length table, `run_len` and run address table, `run_addr`.

```

length ← 0
run_index ← 0
old_address ← -1
For each TV pixel, (i,j), in raster order
  If ((j = NCOLS-1) OR (R(i,j) = invalid)) Then
    address ← -1
    spoke ← 0          { a(spoke,ring) must equal 0 }
    ring ← NSPOKES/2
  Else
    address ← S(i,j) × NRINGS + R(i,j)
    spoke ← S(i,j)
    ring ← R(i,j)
  End If
  If (((j > 0) AND (address ≠ old_address)) OR (j = NCOLS-1)) Then
    run_len[run_index] ← length
    run_addr[run_index] ← old_spoke × NRINGS + old_ring
    run_index ← run_index + 1
    length ← 1
  Else
    length ← length + 1
  End If
  old_address ← address
  old_spoke ← spoke
  old_ring ← ring
End For
NRUNS ← run_index

```

Algorithm: fastmap

Function: This maps the uniform TV image, $I(i, j)$, to the logmap image, $L(u, v)$. It uses the run length and address tables computed by `run_length_encode`.

Input: `run_len` and `run_addr` run tables
TV image, $I(i, j)$

Output: Logmap image, $L(u, v)$

```

tv_index ← 0           { The TV image, I, is accessed in raster order with the index,
                        tv_index. }
For each pixel  $p \in L$    { Zero logmap }
     $L(p) \leftarrow 0$ 
End For
For each run,  $k$ 
     $lm\_addr \leftarrow \text{address\_of}(L) + \text{run\_addr}[k]$ 
    For  $l = 0$  to  $(\text{run\_len}[k] - 1)$ 
        { The next two lines can be written with one C instruction,
           $*lm\_addr += I(\text{tv\_index}++)$ . Note that the ' $*lm\_addr$ ' notation
          denotes the memory location referenced by  $lm\_addr$  }
         $*lm\_addr \leftarrow *lm\_addr + I(\text{tv\_index})$ 
         $tv\_index \leftarrow tv\_index + 1$ 
    End For
End For
For each pixel  $p \in L$    { Normalize logmap }
     $L(p) \leftarrow L(p) / a(p)$ 
End For

```

Chapter 3

Platform Pantilt

3.1 Introduction

Once we have a miniature camera for acquiring space-variant images, we need a method of actuating the camera. We would like a pan-tilt mechanism that is accurate, fast, small, inexpensive and has low power requirements.

We have constructed two new actuators meeting these requirements. The first is called the Platform Pantilt, a pan-tilt actuator based on two linear stepper motors (Figure 3.1) described in this chapter. The second actuator, the Spherical Pointing Motor (SPM), incorporates both pan and tilt into a single two degree of freedom motor (Figure 4.1). It consists of three orthogonal motor windings in a permanent magnetic field, configured to move a small camera attached to a gimbal. It is an absolute positioning device and is run completely open loop. It is described in Chapter 4. Both of these actuator designs are different than traditional pan-tilt actuators in that they avoid the inefficiencies involved with one motor moving another motor.

Pan-tilt mechanisms have been a source of inspiration and frustration to computer vision researchers. One source of inspiration is nature, since humans and animals rely on their pan-tilt apparatus to achieve wide field-of-view visual sensing.

The argument from nature is not by itself a compelling reason to build robot eyes that pan and tilt. The alternative to mechanical pan-tilt action is electronic scanning, i.e. computer control of a fixed set of cameras having a combined wide field-of-view. Selective attention can be implemented by a variety of addressing methods, for example the inverted pyramid of Burt [11,12]. Such "software pan-tilt" mechanisms are considerably more convenient and reliable than their motorized counterparts, but we argue that they ultimately are more expensive. A large sensor, which is required to efficiently use software pan-tilt, has a quadratically increasing pixel load. The alternative to providing the extra processor bandwidth and memory needed for such an approach is to physically move the sensor, as is the usual choice in "active-vision" approaches. It is our opinion that at the current state of technology, this is a considerably more economical approach.

Given today's inexpensive CCD camera arrays, electronic scanning may appear at first glance to be superior to the mechanical approach. Certainly such electronic pan and tilt is faster than mechanical pan and tilt. It also is more reliable, as there are no moving parts involved. But there are several arguments in favor of using mechanical actuation. They are:

cost We have shown at least one example in which the mechanical approach is less expensive. Our prototype SPM carries a camera having a 33 degree field of view. The actuator can pan and tilt the camera through a range of 60 degrees in each axis, resulting in an overall field of view of 93 degrees in each axis. To achieve the same result for the same CCD array in a nonmechanical configuration, we would have to build an array of 9 CCD sensors and lenses, arranged so that their

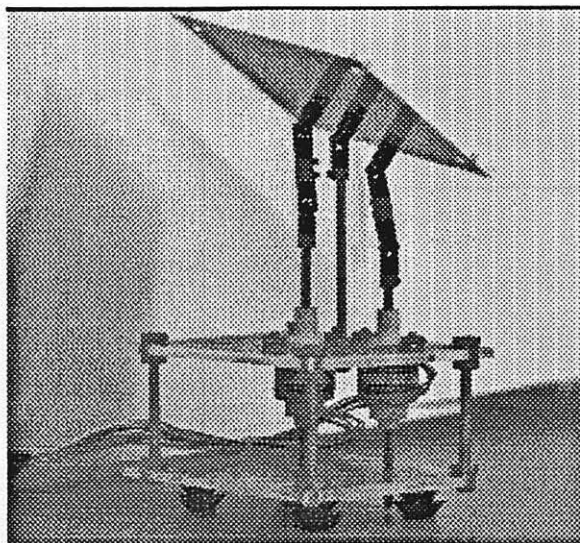


Figure 3.1: The Platform Pantilt. This stepper motor based actuator is capable of positioning a load of 80 grams, for example a small video camera. Dimensions are $7.5 \times 7.5 \times 12$ cm.

fields-of-view are adjacent. Even at the current low prices, 9 CCDs and lens assemblies cost an order of magnitude more than the parts in the SPM. The cost of CCDs increases exponentially with size. Of course, with either approach there is an electronic circuit driving the system. For the SPM, this circuit consists of a motor driver and PWM controller, plus the components necessary to digitize the camera signal. For the fixed array of cameras, the electronics for the camera signals must be repeated 9 times, so we claim that the cost of support electronics is comparable in either case.

registration A system with multiple image sensors would have to be very carefully calibrated in order to register the multiple images. Without knowing precisely how the external world mapped onto each sensor in relation to the other sensors, it would be very difficult if not impossible to use such a system.

optics Using space-variant images with the fovea at the center of the image, and thus in line with the optical axis of the lenses allowed us to miniaturize the camera in a way that would not otherwise have been possible. Our miniature camera has severe off-axis optical degradation. The low resolution in the sensor periphery, however, is lower than the optical resolution which makes an ideal match for our system. This would not work, however, in a system where the foveation point was changed electronically to points off the optical axis, because the optical resolution would be lower than the sensor resolution. New lenses would have to be designed that have high resolution in the periphery. It would be much harder to design such lenses that would be as small as our current lenses.

custom sensor We plan on replacing our uniform off-the-shelf sensor with a custom sensor that has an intrinsic logmap geometry. We have built our entire software platform with this in mind, never using the original uniform TV image. We will therefore be able to make this replacement without changing our software substantially. Once we have a true logmap sensor, it will no longer be possible to control the foveation point electronically.

The simplest and most obvious pan-tilt mechanism uses a two-stage motor-on-motor (MOM) design. The first motor turns the mechanism through one degree of freedom, usually pan, and the second through

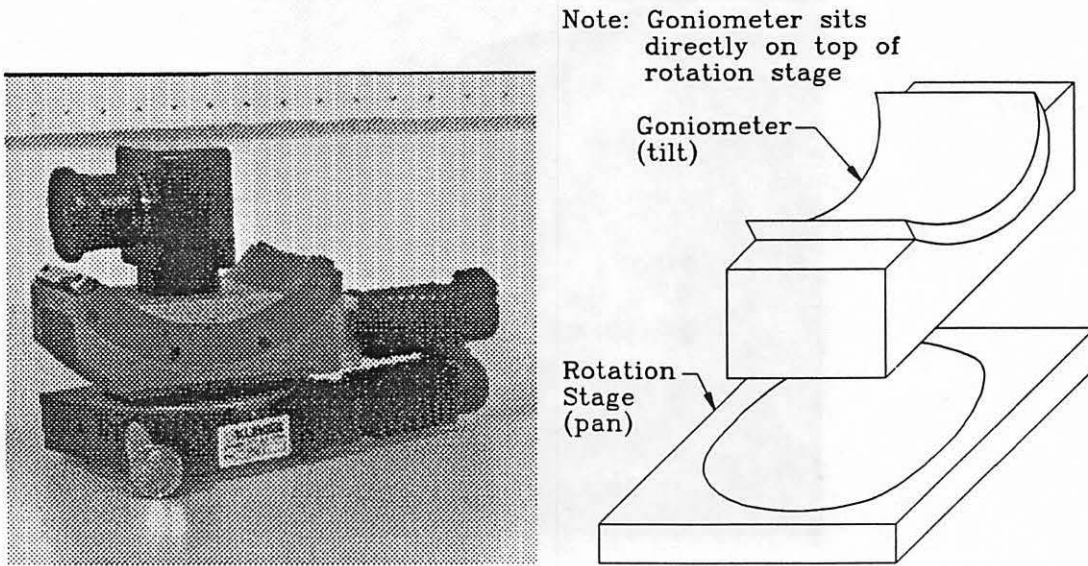


Figure 3.2: Photo and illustration of a traditional motor-on-motor design. Actuator from Klinger Scientific. Dimensions are $10 \times 11 \times 22$ cm (not including the camera or the rack-mounted controller/driver).

the other d.o.f, usually tilt. The second motor must be powerful enough to move the camera sensor. The first must move both the camera and the second motor. The MOM design therefore usually consists of one larger motor and one smaller one. Such a design is inefficient because, as we show, it is not necessary to carry one motor on top of another one. An example of such a design is the pan-tilt actuator available from Klinger Scientific (Figure 3.2).

The traditional solution to the inefficiencies of the MOM design is a parallel approach. This typically involves two motors which through some kind of linkage, actuate the rotor without having one motor moving the other. However, linkages are bulky and are difficult to make accurately. The Platform Pantilt is based on this method.

The Platform Pantilt moves a platform by raising and lowering two shafts with linear stepper motors that are fixed to the platform. The platform is attached to the base at a third point by a fixed shaft. All the shafts have some flexibility with different kinds of joints. This design is similar in some respects to the six degree of freedom Stewart platform [22,52,53]. The Platform Pantilt measures $7.5 \times 7.5 \times 12$ cm and can move at rotational velocities up to $100^\circ/\text{sec}$. The precision is somewhat limited, however, due to the use of stepper motors.

3.2 Background

One source of frustration is acquiring or building, then calibrating and controlling the mechanism itself. Until recently, no manufacturers have provided pan-tilt mechanisms specifically designed for computer control of cameras. Low-speed inexpensive motorized pan-tilt camera platforms are now available, for example from Edmund Scientific (catalog number F38,485), but these have no computer controls. Remote control pan-tilt devices intended for security camera applications, for example the Graystone Model V370PT, tend to be too slow (6 degrees per second) and too heavy (16 lbs.) for many robotic applications. Few of these devices have computer controls or position feedback information. The motion picture industry has also developed computer controlled pan-tilt devices. One example is the Kaleidoscope

Hothead II, made by Shepperton Film Studios. Although a digital interface is available, this device is intended to carry a large load such as a 35mm motion picture camera, and although it achieves relatively high speed ($140^\circ/\text{sec}$), it is very expensive. Another source of pan-tilt mechanisms is the lighting industry, which applies them to stage lighting, exhibitions and advertising. A high-speed computer controlled pan-tilt mechanism for stage lights is available from Multiline, but its bracketing and high cost make it unappealing for robot vision. A number of manufacturers (for example Aerotech, Daedel, Unislide, and Klinger Scientific) make computer controlled mechanisms that can be assembled into pan-tilt devices. Better suited for manufacturing and optics applications, the high resolution of these devices (less than 1 arc minute) makes them overkill for many computer vision applications and contributes to their high cost.

Perhaps because of their familiarity and availability, if not their generality and programmability, robot arms have often been the choice of vision researchers trying to actuate their cameras. Baloch and Waxman used a 5-axis robot arm mounted on top of a mobile robot as a camera pointing mechanism [8]. Allen also reported mounting a TV camera on a robot arm [4]. Raviv used a cartesian manipulator to implement camera pan, tilt, roll and translation [40].

Finally, a number of researchers including ourselves have embarked on building their own pan-tilt devices from scratch. Krotkov built what is now recognized as the first robot head, a computer controlled mechanism for moving two cameras [33]. Abbot and Ahuja report an 11 degree of freedom mechanism to control pan, tilt, vergence, horizontal translation, and lens parameters [2,3]. From Osaka University, Kawarabayashi et. al. report building an active vision "head" to control pan, tilt, vergence, zoom and focus parameters of a pair of cameras [29]. Dickmanns also reports a "fast" two-axis pan-tilt device carrying two cameras, one with a wide-angle view and the other telephoto, mounted atop their robotic automobile [20]. At Harvard, Clark and Ferrier constructed a seven degree-of-freedom "head" to control pan, tilt, and the vergence, focus and aperture of two cameras [16]. At least one two-eye system, the Rochester Robot, contains independent pan controls for two cameras on a tilting platform, in contrast to other systems in which vergence is coupled [13]. The pan-tilt mechanism in all of these designs is a MOM design.

Although many vision researchers are also interested in controlling parameters other than pan and tilt, we have restricted our attention in this thesis to these two degrees of freedom. We only mention here that pan-tilt mechanisms are not the whole story. We would like to build inexpensive computer controls for such camera parameters as focus, zoom, and aperture. In some cases it may be desirable to control camera roll and XYZ translation as well. With two or more cameras present, as in a stereo vision system, we would like to control the vergence angle between the camera as well as their baseline separation. These interesting subjects will remain outside the scope of the present discussion.

In this chapter we motivate the design and analyze the forward and inverse kinematics of the Platform Pantilt in order to prove that the linkage is mechanically sound.

3.3 Mechanism

The Platform Pantilt consists of a moving platform attached to a base with one fixed shaft and two shafts driven by linear stepper motors (Figure 3.1). The mechanism linking the stepper motors to the platform needs to transmit force and allow rotation while maintaining a 1-1 correspondence between motor and platform positions. This is accomplished by a combination of four universal joints and one pin joint.

Our original design had only one universal joint on each shaft. This, however, does not work because as the platform is rotated, the horizontal component of the distance between the joints decreases. The two-dimensional case depicted in Figure 3.3 shows what goes wrong with only one joint on the drive shaft. The shafts are rigidly fixed perpendicular to the base and to the platform. The distance between the joints is fixed. When the drive shaft extends, the angle between the drive shaft and the base must decrease which it cannot do.

One technique to overcome this problem, as is done with a Stewart Platform [22,24,52,53], is to attach

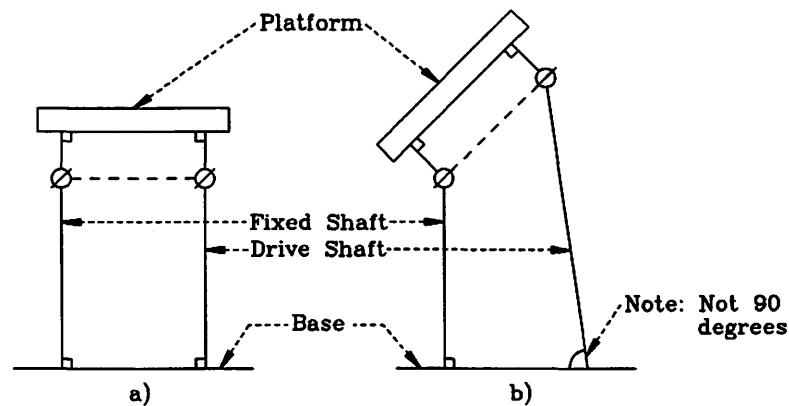


Figure 3.3: A non-working two-dimensional mechanism in two different positions with one pin joint on each shaft. Note that in b), the lower right angle is not fixed at 90° as it should be.

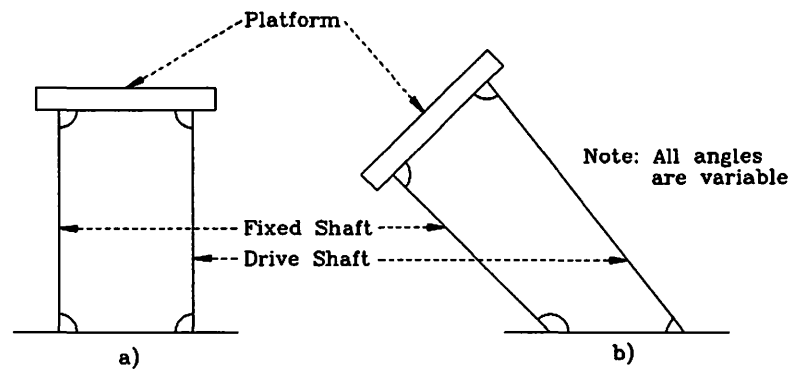


Figure 3.4: A two dimensional mechanism based on the Stewart Platform. Here, the drive shafts must be mounted on spherical joints.

the drive shafts to the base and to the platform with spherical joints (See Figure 3.4). Although this works in the two dimensional case, it does not work in three dimensions because the forward kinematics problem does not have a unique solution. The simplest description of the problem occurs when both drive shafts are fixed at the same height. In this configuration, the platform will have one uncontrollable degree of freedom as it rotates around the fixed shaft. This problem could be resolved by constraining the passive rotation of one of the motors to a single degree of freedom. However, this technique is not well-suited to miniaturization for two reasons: 1) It is difficult to build very small gimbal joints that could support the motor and platform; and 2) The motors themselves must be rotated which means that the motors must be large enough to counter their own rotational inertia.

The technique we have chosen avoids the problem of rotating the mass of the motors. The motors are fixed perpendicular to the base, and an extra joint is added to the drive shafts. Figure 3.5 illustrates the two dimensional case. Here, as the drive shaft extends and retracts (corresponding to h_1 increasing and decreasing in length), point C moves straight up and down. Point B is a fixed distance, s , from point C , and a fixed distance, d , from the fixed point A . So, point B moves along the point defined by the intersection of the two circles centered at A and C with radii d and s , respectively. Although there

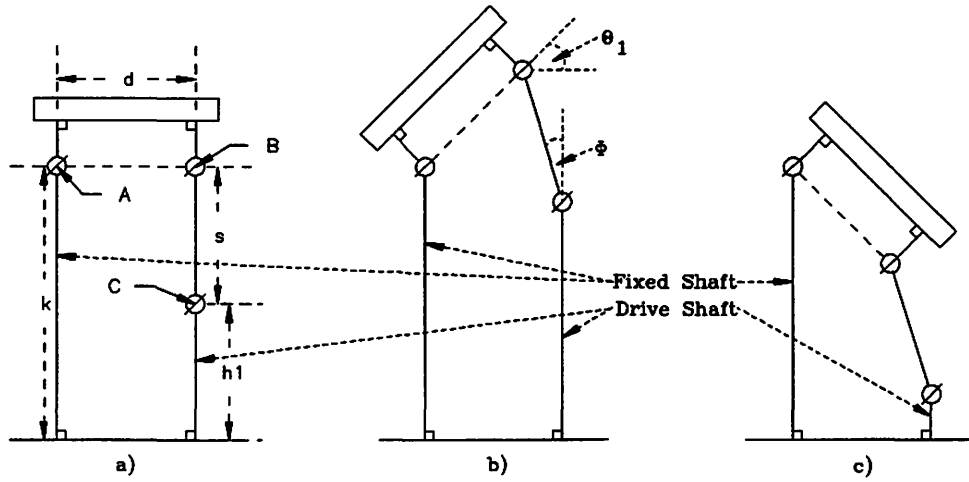


Figure 3.5: A two dimensional version of the mechanism we use shown in three positions. All the connections between the shafts and the platform and base are rigid. The dashed line between points A and B represents a fixed distance and does not represent a physical connection.

are two solutions for the point B, only one of them is physically attainable.

The straightforward extension of this mechanism to three dimensions by adding another drive shaft rigidly linked with two universal joints to the platform does not work. The platform would have one degree of uncontrollable freedom for all sets of drive shaft positions. The solution is to constrain one of the drive shaft linkages to two dimensions. This is easily done by replacing the lower universal joint on one of the drive shafts with a single degree of freedom pin joint. With this modification, the mechanism has no singularities. This is depicted in Figure 3.6.

With this approach, the kinematics are most easily analyzed by first looking at the two dimensional case and then extending the solution to three dimensions.

3.4 Forward kinematics in two dimensions

The forward kinematics problem for the two dimensional system (Figure 3.5) can be simply stated as "Given h_1 , what is Θ_1 ?" That is, given the motor position, what is the angle of the platform. We define the origin to be at the bottom of the fixed shaft and we use homogeneous coordinates. We define the angle of the platform using the motor position and then solve using the known constraints. The angle of the platform is:

$$\Theta_1 = \sin^{-1} \left(\frac{B_y - A_y}{d} \right) \quad (22)$$

where A is the fixed point $(0, k, 1)$ and B is dependent on the motor position. C is the variable point $(d, h_1, 1)$. We can calculate B by taking the vector $(0, s, 1)$ rotated about an angle ϕ , and adding it to point C where ϕ is constrained by the fact that point B is a fixed distance d from point A.

$$B = (0, s, 1) \cdot [\text{rot } \phi] \cdot [\text{trans } C]$$

where

$$[\text{rot } \phi] = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

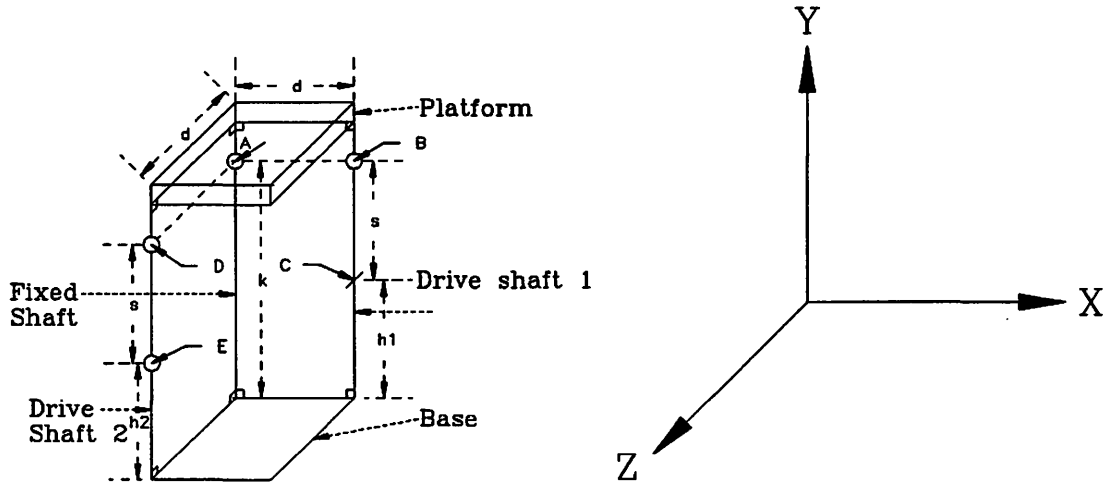


Figure 3.6: left) The three dimensional mechanism we use. Note that circles at points A, B, D and E denote universal joints with two degrees of freedom while the line at point C denotes a pin joint with only one degree of freedom. right) The coordinate system used in this section.

and

$$[\text{trans } C] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ C_x & C_y & 1 \end{bmatrix}.$$

Then, expanding B and substituting $d = C_x$ and $h_1 = C_y$, we can rewrite

$$B = (d - s \sin \phi, h_1 + s \cos \phi, 1). \quad (23)$$

Because B is a fixed distance d from point A , we add the constraint

$$(B_x - A_x)^2 + (B_y - A_y)^2 = d^2$$

and plug in the equations for B_x and B_y , using $A_x = 0$ and $A_y = k$ to get

$$(d - s \sin \phi)^2 + (h_1 + s \cos \phi - k)^2 = d^2. \quad (24)$$

We now want solve for ϕ . Expand Equation 24, plugging in $h_1 = C_y$, $k = A_y$ and $d = C_x$ to get

$$-2sC_x \sin \phi + s^2 \sin^2 \phi + (C_y - A_y)^2 + 2s(C_y - A_y) \cos \phi + s^2 \cos^2 \phi = 0$$

$$\underbrace{-2dC_x \sin \phi}_a + \underbrace{2s(C_y - A_y) \cos \phi}_b = \underbrace{-(C_y - A_y)^2 - s^2}_c$$

$$a \sin \phi + b \cos \phi = c \quad (25)$$

Using the triangle in Figure 3.7a, we can write

$$\sin \phi = \frac{m}{\sqrt{m^2 + 1}}$$

$$\cos \phi = \frac{1}{\sqrt{m^2 + 1}}$$

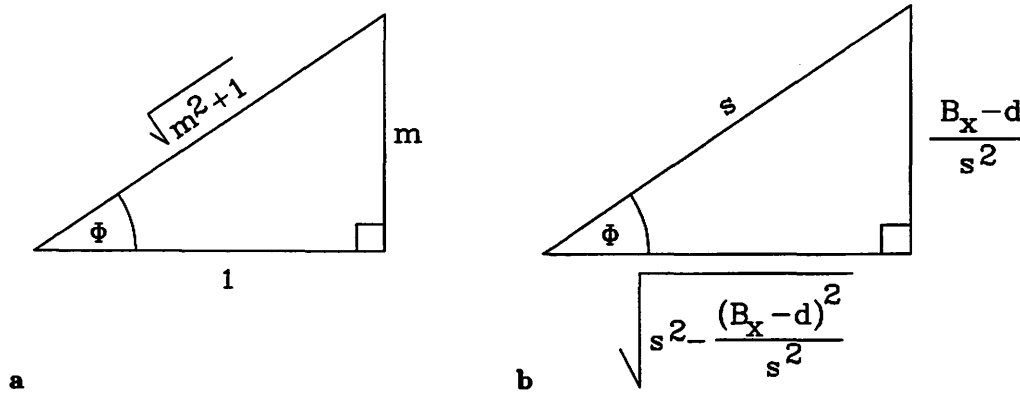


Figure 3.7: a) Triangle used to solve Equation 25. b) Triangle used to solve Equation 26.

Equation 25 becomes

$$\begin{aligned}
 \frac{am}{\sqrt{m^2+1}} + \frac{b}{\sqrt{m^2+1}} &= c \\
 \frac{(am+b)^2}{m^2+1} &= c^2 \\
 a^2m^2 + 2abm + b^2 - c^2m^2 - c^2 &= 0 \\
 (a^2 - c^2)m^2 + (2ab)m + (b^2 - c^2) &= 0 \\
 m = \frac{-2ab \pm \sqrt{4a^2b^2 - 4(a^2 - c^2)(b^2 - c^2)}}{2(a^2 - c^2)} \\
 m = \frac{-2ab \pm 2c\sqrt{a^2 + b^2 - c^2}}{2(a^2 - c^2)}
 \end{aligned}$$

We can now solve directly for Θ_1 using Equation 22. From Equation 23, we have the value for B_y . We know that $A_y = k$, and $C_y = h_1$. Then,

$$\begin{aligned}
 \Theta_1 &= \sin^{-1} \left(\frac{h_1 + s \cos \phi - k}{d} \right) \\
 \Theta_1 &= \sin^{-1} \left(\frac{h_1 - k + s/\sqrt{m^2+1}}{d} \right)
 \end{aligned}$$

3.5 Inverse kinematics in two dimensions

The inverse kinematics problem for the two dimensional system can be stated by "Given Θ_1 , what is h_1 ?" This is solved similarly to the forward kinematics problem. Now we know point B and we must find point C in order to find h_1 .

$$h_1 = C_y$$

where

$$\begin{aligned}
 C &= (0, -s, 1) \cdot [\text{rot } \phi] \cdot [\text{trans } B] \\
 C &= (B_x - s \sin \phi, B_y - s \cos \phi, 1)
 \end{aligned}$$

and we know that $C_x = d$. We can expand C_x to get

$$B_x - s \sin \phi = d$$

We can write B as

$$\begin{aligned} B &= (d, 0, 1) \cdot [\text{rot } \Theta_1] \cdot [\text{trans } A], \\ B &= (A_x + d \cos \Theta_1, A_y + d \sin \Theta_1, 1) \end{aligned}$$

We also know that $C_y = h_1$, so we can expand C_y , plugging in for B_y and get

$$h_1 = A_y + d \sin \Theta_1 - s \cos \phi \quad (26)$$

We know $\sin \phi$, but need $\cos \phi$. We can find it by using the triangle in Figure 3.7b. Then we can see that

$$\cos \phi = \frac{\sqrt{s^2 - (B_x - d)^2 / s^2}}{s}$$

Plug this in to Equation 26 and get

$$h_1 = A_y + d \sin \Theta_1 - \sqrt{s^2 - \frac{(B_x - d)^2}{s^2}}$$

Plug in for B_x and get

$$h_1 = A_y + d \sin \Theta_1 - \frac{1}{s} \sqrt{s^4 - (a_x + d \cos \Theta_1 - d)^2}$$

3.6 Forward kinematics in three dimensions

The forward kinematics in three dimensions can be calculated in an analogous manner to the two dimensional case, but the math becomes extremely messy so an alternative approach is used. We assume the two dimensional case has already been solved and thus we know the position of point B . Then, given h_2 , we want to know Θ_2 .

$$\Theta_2 = \sin^{-1} \left(\frac{D_y - A_y}{d} \right) \quad (27)$$

Point A is known to be $(0, k, 0, 1)$, and point E is $(0, h_2, d, 1)$. Then, as point D is a fixed distance from points E , A , and B , it can be solved for by computing the intersection of the three spheres centered at those points. The equations describing these spheres are:

$$(D_x - E_x)^2 + (D_y - E_y)^2 + (D_z - E_z)^2 = s^2 \quad (28)$$

$$(D_x - A_x)^2 + (D_y - A_y)^2 + (D_z - A_z)^2 = d^2 \quad (29)$$

$$(D_x - B_x)^2 + (D_y - B_y)^2 + (D_z - B_z)^2 = 2d^2 \quad (30)$$

The solution for point D involves solving these three quadratic equations and three unknowns, D_x , D_y , and D_z , as follows. Expand these equations, rewriting the constants and we get:

$$D_x^2 + D_y^2 + D_z^2 + aD_x + bD_y + cD_z = d \quad (31)$$

$$D_x^2 + D_y^2 + D_z^2 + eD_x + fD_y + gD_z = h \quad (32)$$

$$D_x^2 + D_y^2 + D_z^2 + iD_x + jD_y + kD_z = l \quad (33)$$

where

$$\begin{aligned}
 a &= -2E_x \\
 b &= -2E_y \\
 c &= -2E_z \\
 d &= s^2 - E_x^2 - E_y^2 - E_z^2 \\
 e &= -2A_x \\
 f &= -2A_y \\
 g &= -2A_z \\
 h &= d^2 - A_x^2 - A_y^2 - A_z^2 \\
 i &= -2B_x \\
 j &= -2B_y \\
 k &= -2B_z \\
 l &= 2d^2 - B_x^2 - B_y^2 - B_z^2
 \end{aligned}$$

Subtract Equation 32 from Equation 31 and Equation 33 from Equation 32 to get rid of the quadratic terms. We end up with equations for two planes.

$$(a - e)D_x + (b - f)D_y + (c - g)D_z = d - h \quad (34)$$

$$(e - i)D_x + (f - j)D_y + (g - k)D_z = h - l \quad (35)$$

The intersection of these two planes is a line, so we find its parametric equation and plug it into one of the original spheres (Equation 28). Note that because of the choice of the coordinate system, $a = e = g = k = 0$, and thus the coefficients of D_x in Equation 34 and of D_x in Equation 35 go to zero. Now pick two points, q_0 and q_1 on this line.

$$\begin{aligned}
 q_{0y} = 0 &\Rightarrow q_{0z} = \frac{d - h}{c}, \quad q_{0x} = \frac{h - l}{-i} \\
 q_{1x} = 0 &\Rightarrow q_{1y} = \frac{h - l}{f - j}, \quad q_{1z} = \frac{d - h - (b - f)(h - l)/(f - j)}{c}
 \end{aligned}$$

Note that these equations are not solvable when the denominator is equal to zero which occurs only at extreme positions. Now $q_0 + t(q_1 - q_0)$ on $t \in [-\infty, \infty]$ is the parametric formula for the line satisfying Equations 34 and 35. This can be plugged into Equation 28 to get one quadratic equation in t , which because $E_x = 0$, is equal to

$$[q_{0x} + t(q_{1x} - q_{0x})]^2 + [q_{0y} + t(q_{1y} - q_{0y}) - E_y]^2 + [q_{0z} + t(q_{1z} - q_{0z}) - E_z]^2 = s^2$$

Solve for t and plug into $q_0 + t(q_1 - q_0)$ to get point D , and finally plug into Equation 27 to get Θ_2 .

3.7 Inverse kinematics in three dimensions

Fortunately, the inverse kinematics problem in three dimensions is much easier than the forward one. We want to find h_2 given Θ_2 .

$$h_2 = E_y$$

We get E by taking the vector $(0, -s, 0, 1)$ rotated ψ_1 and ψ_2 about the x and z axes, respectively, and adding it to point D , where point D is calculated by taking the vector $(0, 0, d, 1)$ rotated Θ_1 and Θ_2 about the z and x axes, respectively, and adding it to point A .

$$\begin{aligned} D &= (0, 0, d, 1) \cdot [\text{rotx } \Theta_2] \cdot [\text{rotx } \Theta_1] \cdot [\text{trans } A] \\ &= (d \sin \Theta_2 \sin \Theta_1, k - d \sin \Theta_2 \cos \Theta_1, d \cos \Theta_2, 1) \end{aligned}$$

where

$$[\text{rotx } \Theta] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Theta & \sin \Theta & 0 \\ 0 & -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } [\text{rotx } \Theta] = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 & 0 \\ -\sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$E = (0, -s, 0, 1) \cdot [\text{rotx } \psi_1] \cdot [\text{rotx } \psi_2] \cdot [\text{trans } D] \quad (36)$$

$$= (D_x + s \cos \psi_1 \sin \psi_2, D_y - s \cos \psi_1 \cos \psi_2, D_z - s \sin \psi_1) \quad (37)$$

Now we have the constraints $E_x = 0$ and $E_z = d$, so we get

$$E_z = d \Rightarrow \psi_1 = \sin^{-1} \left(\frac{d \cos \Theta_2 - d}{s} \right)$$

$$E_x = 0 \Rightarrow \psi_2 = \sin^{-1} \left(\frac{-d \sin \Theta_2 \sin \Theta_1}{s \cos \psi_1} \right)$$

Finally, we can plug ψ_1 and ψ_2 into E_y in Equation 37

$$\begin{aligned} h_2 &= k - d \sin \Theta_2 \cos \Theta_1 - \\ &\quad s \cos \left(\sin^{-1} \left(\frac{d \cos \Theta_2 - d}{s} \right) \right) \cos \left(\sin^{-1} \left(\frac{-d \sin \Theta_2 \sin \Theta_1}{\sqrt{s^2 - (d \cos \Theta_2 - d)^2}} \right) \right) \end{aligned}$$

3.8 Limits of motion

The limits of motion are depicted in Figure 3.8. The counterclockwise rotation is limited by the universal joints that can not rotate more than 90° . To achieve this, length s must be greater or equal to length d . The clockwise rotation achieves its maximum when points A , B , and C are colinear. This occurs when $\cos \Theta = d/(d + s)$. So,

$$\Theta_{\max} = 90^\circ,$$

and

$$\Theta_{\min} = -\cos^{-1} \left(\frac{d}{d + s} \right).$$

3.9 Simulation of kinematics

To verify the correctness of the equations of this section, we implemented a software graphics simulation of both the forward and inverse kinematics. Figure 3.9 shows the solution of the forward kinematics problem with the Platform Pantilt shown in several positions.

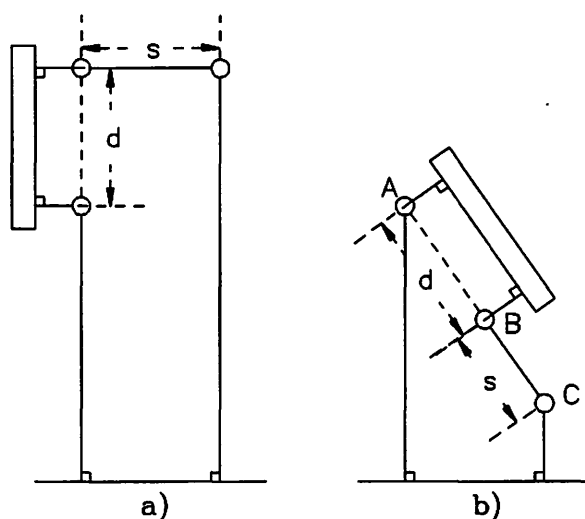


Figure 3.8: The limits of motion in two dimensions.

3.10 Prototype Platform Pantilt

The prototype Platform Pantilt was made with miniature 1" diameter linear stepper motors. The linear motion has a maximum travel of 1.875" and travels 0.004 inches/step. At a maximum of 550 steps per second, the motors can travel 2.2 linear inches/second. Our prototype motor has the following geometry (where the constants refer to Figure 3.6).

Parameter	Value
d	1.9 cm
s	2.8 cm
k	6.4 cm

The total size of the Platform Pantilt is $7.5 \times 7.5 \times 12$ cm. The platform has the maximum rotation, $\Theta_{max} = 90^\circ$ when $h_{max} = 6.1$ cm and the minimum rotation, $\Theta_{min} = -66^\circ$ when $h_{min} = 2.0$ cm. So the average theoretical maximum rotational velocity is equal to $(\Theta_{max} - \Theta_{min}) / (h_{max} - h_{min}) \cdot 2.2 \text{ in/sec} = 214^\circ/\text{sec}$.

In practice, because stepper motors' strength is inversely proportional to their speed, they sometimes miss steps at high velocity. Thus the Platform Pantilt can be run reliably only at about 100 degrees/second.

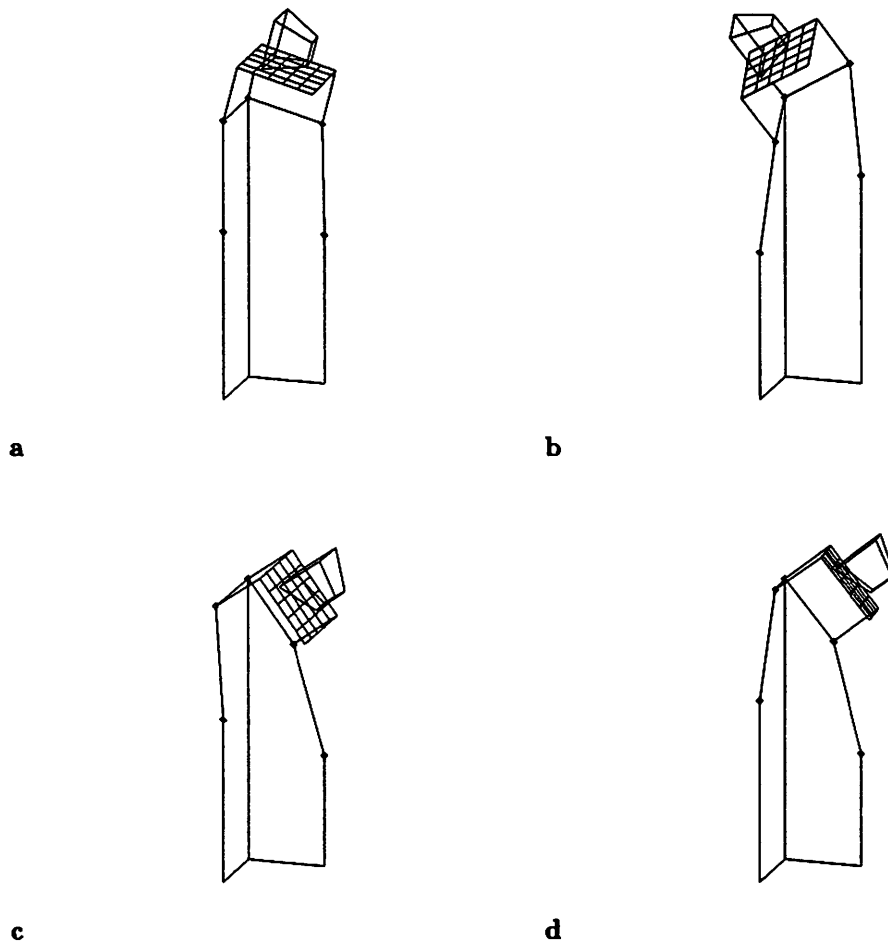


Figure 3.9: Simulation of the forward kinematics equations of this section. The Platform Pantilt is shown in four different positions.

Chapter 4

Spherical Pointing Motor

To overcome the limitations of the parallel motor approach, we designed and built the Spherical Pointing Motor¹ (SPM), a miniature single direct drive motor with two degrees of freedom. The SPM is capable of panning and tilting a load of 6 grams, for example, the CCD-based camera described in Chapter 2, at rotational velocities of up to several hundred degrees per second. Consisting of only wire wound around a metal form, a simple gimbal, and a small permanent magnet, the SPM is arguably the smallest and least expensive possible device capable of pointing a camera accurately and at high speed.

4.1 Spherical pointing motor theory

The Spherical Pointing Motor is an absolute positioning device, designed to orient a small camera sensor in two degrees of rotational freedom. The basic idea is to orient a permanent magnet to the magnetic field induced by three orthogonal coils by applying the appropriate ratio of currents to the coils. A simple way to understand this device follows: The net magnetic field of the three coils may be visualized as defining a vector (dipole) oriented at angles (Θ, Φ) on the unit sphere. The angles (Θ, Φ) are determined by the three coil currents. The rotor dipole then aligns itself with the net coil field to provide the actuation.

We have built two types of SPMs: one having the coils on the inside, free to rotate on a gimbal inside a fixed magnetic field, (Figure 4.2); the other having the coils on the outside and the magnets attached to the gimbal inside the coils (Figure 4.3). It is possible to build the smallest possible motor using the external coil design, so that is the approach we focus our attention on.

In this chapter, we look at the transfer function taking input current to pan and tilt angles. We point out some design constraints on the configuration of the coils and the permanent magnets, and show why the external coil design is preferable to the internal one. We discuss calibration of the motor, its dynamics and control.

Both designs are constrained by two principles that affect the range of motion of the motor. The SPM is meant to be used as a pointing device. As such, it has a *home position*, defined as the initial resting position from which the motor can make pan or tilt excursions of limited extent. Assuming that we want the home position to be centered within the possible excursions, we are led to the following constraints:

1. The permanent magnets must be positioned so that the field they define is orthogonal to both axes of rotation of the gimbal when it is in home position.
2. The camera must be positioned on the rotor so that its optical axis is orthogonal to both axes of rotation of the gimbal when it is in home position. Note that this is equivalent to being aligned

¹Patent application #07/731,639, entitled "Spherical Pointing Motor" was submitted to the U.S. Patent office in 1991.

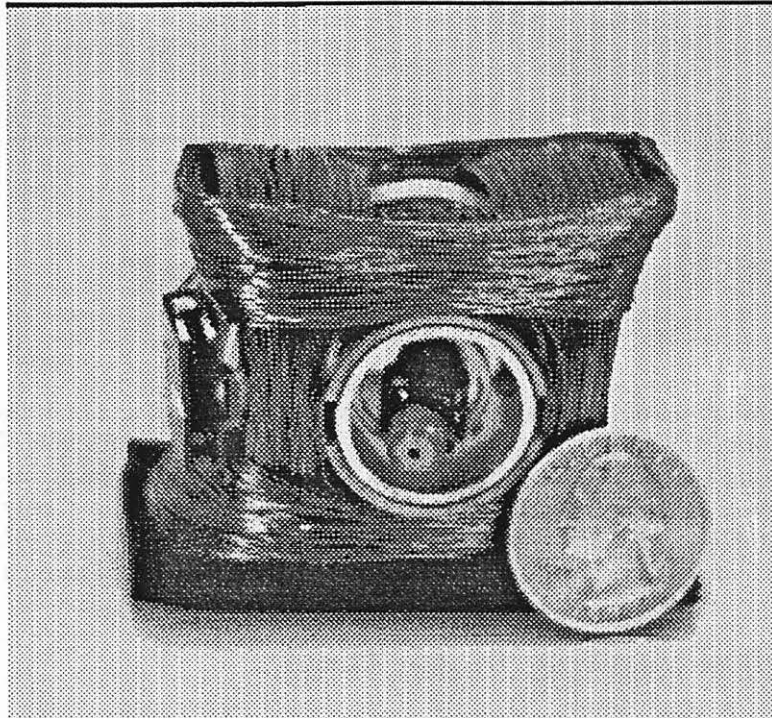


Figure 4.1: The Spherical Pointing Motor. At the center is a miniature camera consisting of a single CCD sensor chip and a lens assembly that fits on the rotor of this motor.

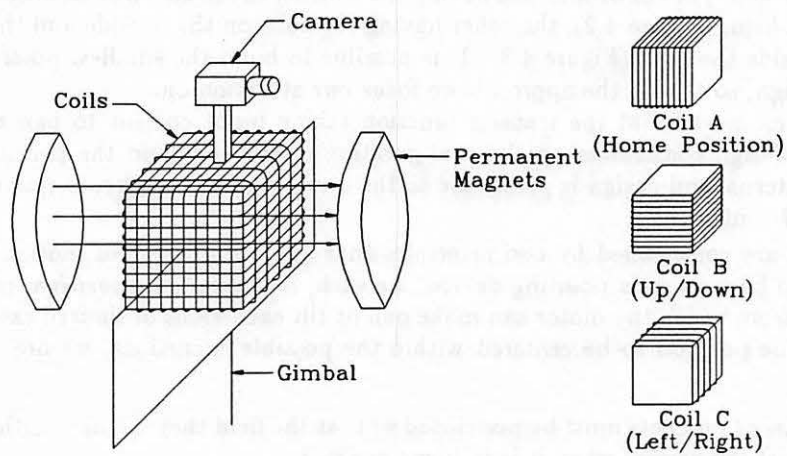


Figure 4.2: **left)** Illustration of the internal spherical pointing motor shown in its home position. **right)** Labels for the three coils.

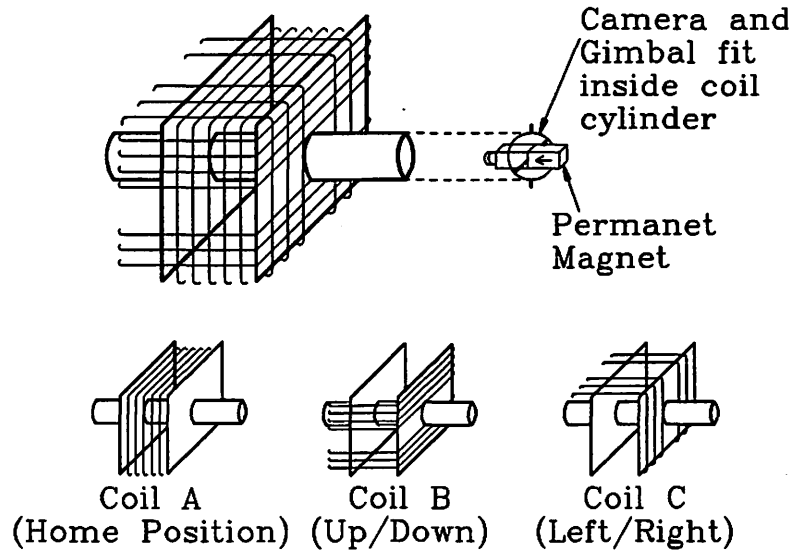


Figure 4.3: top) Illustration of the external spherical pointing motor shown in its home position. bottom) Labels for the three coils.

with the permanent magnetic field if the first design constraint is satisfied.

These design principles arise because the motor is limited to two mechanical degrees of freedom and because the motor rotation cannot be controlled about an axis aligned with the permanent magnetic field. To understand why this is so, we must examine the electromagnetic principle that provides the torque to move the motor.

The torque on the rotor comes from the basic electromagnetic principle that there is always an induced force on a current-carrying wire in a permanent magnetic field. Further, there will be a torque on a current-carrying wire loop in a permanent magnetic field in such a direction that the loop will move to make the normal to the plane of the loop align with the magnetic field. This is shown in Figure 4.4.

Given a coil of wire in a uniform magnetic field $\mathbf{B}$, the torque $\boldsymbol{\tau}$ exerted on the coil is the cross product

$$\boldsymbol{\tau} = \bar{\boldsymbol{\mu}} \times \mathbf{B} \quad (38)$$

where $\bar{\boldsymbol{\mu}}$ is the magnetic dipole moment having direction perpendicular to the plane of the coil and magnitude

$$|\bar{\boldsymbol{\mu}}| = N\iota A \quad (39)$$

where N is the number of windings in the coil, ι is the current in the wire, and A is the area enclosed by the coil [23,30]. The sign of $\bar{\boldsymbol{\mu}}$ is determined by the direction of current in the wire loop. When a nonzero current ι flows through the loop, there will be a torque on the loop so that the loop will rotate to its minimum energy configuration where $\boldsymbol{\tau} = 0$, i.e. $\bar{\boldsymbol{\mu}}$ is aligned with $\mathbf{B}$. Note that in the SPM, the permanent magnetic field is not uniform around the area of the coil. This difference is discussed in Section 4.3.

Equation 38 implies that the torque is maximum when the angle between $\bar{\boldsymbol{\mu}}$ and $\mathbf{B}$, $\Psi = 90$ and drops to nothing when $\Psi = 0$. When the coil is at position $\Psi = 0$, it is at its minimum potential energy and there is no force on it. Because of this relationship between torque and angle, the friction of the bearings must be minimized. The precision of the motor is inversely related to the amount of friction in

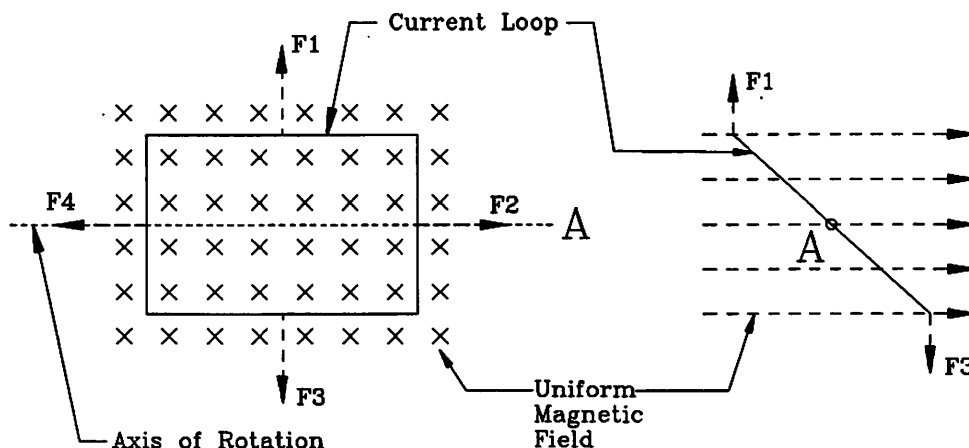


Figure 4.4: Torque on a current loop. **left)** Current loop rotates about horizontal axis with clockwise current. Uniform magnetic field normal to the page. Forces F_2 and F_4 are aligned with the axis and produce no net force. Forces F_1 and F_3 rotate the coil so that the coil encloses as many magnetic field lines as possible which occurs when the normal to the plane of the coil is aligned with the magnetic field. **right)** The same loop, but viewed from the side. (Adapted from [23, P. 541])

the bearings. This is because fine movements of the motor occur at very small angles where the torque is very small, and such fine movements must overcome the friction of the bearings.

If the first design constraint is not satisfied, then it is not possible to both pan and tilt the motor from the home position. Coils A and B control the tilting while coils A and C control the panning (Figures 4.2 and 4.3). When the motor has panned 90° , coil B has kept its original orientation with respect to the magnets, but coils A and C have swapped their relative orientation. This is depicted in Figure 4.5 for the internal coil motor and the analog for the external coil motor is in Figure 4.6. In this position, the pan angle is controlled, but there is no way to control the tilt angle in this orientation (vice versa for the external coil design). This is because the axis of rotation is aligned with the magnetic field and no torque can be exerted around this axis. This is illustrated in Section 4.2.3. Without the first design constraint, the home position could be as depicted in Figure 4.5 in which case the motor could not be tilted along the vertical meridian.

The second design constraint stems from assuming that the two degrees of freedom desired for the camera are pan and tilt. If instead roll (rotation about the optical axis) and either pan or tilt is desired, then the second design constraint is not necessary. This constraint comes directly from the mechanical degrees of freedom available. If pan and tilt are desired, the camera's optical axis must be orthogonal to the two degrees of mechanical freedom.

The design constraints result in an undesirable effect for the internal coil motor. Because the first design constraint requires that the permanent magnets must also be positioned orthogonally to both axes of rotation, the optical axis of the camera must be aligned with the permanent magnetic field. Since the camera is between the two permanent magnets, its field of view will be obscured by the magnets. Therefore, some method of looking over the magnets must be found. This could possibly be done with mirrors, or as depicted in Figures 4.3 and 4.6, the camera could be mounted on a stalk. This gives the external coil motor a clear advantage because the camera is located in the center of the motor. This not only decreases the moment of inertia, but also allows the camera to be rotated around the focal point of the lens which is usually desired for machine vision applications.

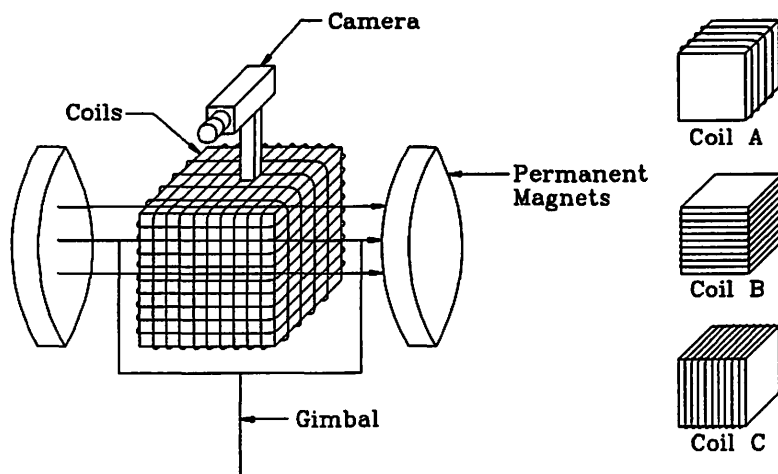


Figure 4.5: left) The internal coil motor panned 90° so that tilting is not possible. right) The coil labels for the motor in this position. Notice how coils *A* and *C* have swapped orientations.

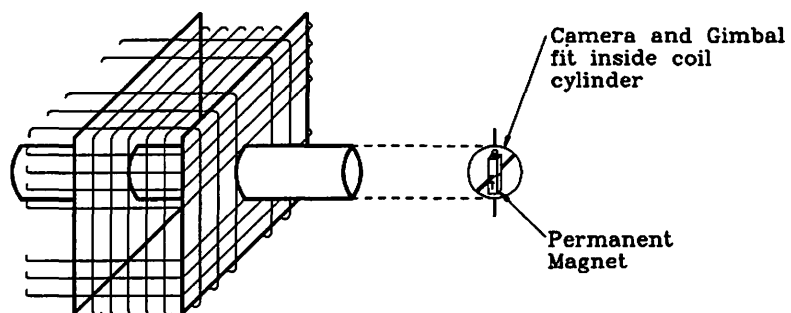


Figure 4.6: The external coil motor tilted 90° so that panning is not possible.

4.2 Theoretical Calculation of motor position

We can calculate the relation between motor orientation and coil currents by realizing that the current in each coil induces a magnetic field normal to the plane of the coil, and that the magnetic fields from the three coils add vectorially. Thus, the ratio of the coil currents determines the motor orientation. Applying a set of currents to the coils creates a torque on the rotor that will turn it to the position of lowest potential energy, so the SPM is an absolute positioning device. In this section, we assume ideal coils, i.e., they are perfectly symmetrical, have the same number of turns, and are the same size. We drop these assumptions in Section 4.3.

4.2.1 Position given currents

From Equation 38, we know that the torque on the rotor is proportional to the sine of the angle between the normal to the coil and the permanent magnetic field. If $\vec{\mu}$ is aligned with $\mathbf{B}$, pointing in either the same or the opposite direction, there is no torque. However, when they are pointing in the same direction, the potential energy is minimum and the coil is in a stable resting state. If they are pointing in opposite directions, the potential energy is maximum and the coil is in an unstable resting state. If the coil is slightly perturbed, it will swing around 180° to reach the minimum potential energy state.

We will use the external coil motor as an example, but the following derivation applies equally to the internal coil motor. We talk about torque on the magnets for the external coil motor, but an equal torque acts on the coils for the internal coil motor. For example, the torque on the magnets due to coil A for the external coil motor is equivalent to the torque on coil A for the internal coil motor.

We will use the coil labels shown in Figure 4.3. As the torque from each coil is dependent on the motor orientation, we must calculate when the sum of the torques due to all three coils vanishes. The destination position of the motor results from satisfying the equation

$$\tau_A + \tau_B + \tau_C = 0 \quad (40)$$

where τ_A , τ_B , and τ_C are the torques on the rotor due to coils A, B, and C. We define the coordinate system we will use in Figure 4.7. Here, both the permanent magnetic field and the home position are defined to lie in the direction of the positive x-axis. Θ (or tilt angle) is defined as the positive rotation about the z-axis and Φ (or pan angle) is defined as the negative rotation about the y-axis.

Let us examine the torque due to coil A. In home position, the magnetic field, $\mathbf{B} = (1, 0, 0)$, and coil A has the magnetic dipole moment, $\vec{\mu}_A = N_A A_A \iota_A (1, 0, 0)$. We use here, the constant $K_A = N_A A_A$. To calculate the torque from coil A at position (Θ, Φ) , we rotate $\vec{\mu}_A$ by Θ around the z-axis and then by Φ around the y-axis, yielding the new dipole moment, $\vec{\mu}'_A$.

$$\begin{aligned} \vec{\mu}'_A &= K_A \iota_A (1, 0, 0) \cdot [\text{rot}_Z(\Theta)] \cdot [\text{rot}_Y(\Phi)] \\ &= K_A \iota_A (\cos \Theta \cos \Phi, \sin \Theta, -\cos \Theta \sin \Phi) \end{aligned}$$

Then,

$$\begin{aligned} \tau_A &= \vec{\mu}'_A \times \mathbf{B} \\ &= K_A \iota_A (0, -\cos \Theta \sin \Phi, -\sin \Theta) \end{aligned}$$

We can calculate the torque from coils B and C similarly.

$$\begin{aligned} \vec{\mu}'_B &= K_B \iota_B (0, 1, 0) \cdot [\text{rot}_Z(\Theta)] \cdot [\text{rot}_Y(\Phi)] \\ &= K_B \iota_B (-\sin \Theta \cos \Phi, \cos \Theta, \sin \Theta \sin \Phi) \\ \tau_B &= \vec{\mu}'_B \times \mathbf{B} \\ &= K_B \iota_B (0, \sin \Theta \sin \Phi, -\cos \Theta) \end{aligned}$$

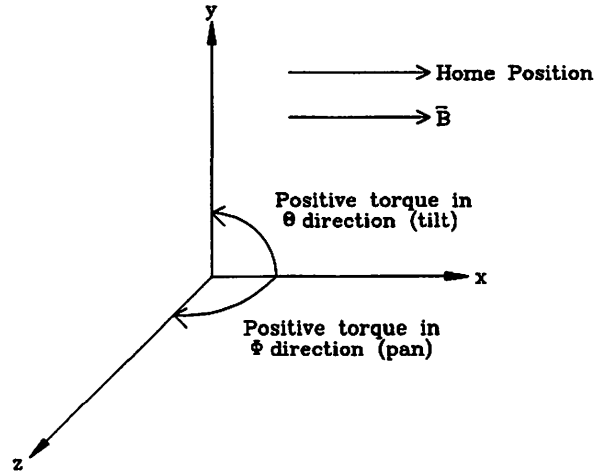


Figure 4.7: The coordinate system used for calculations. The permanent magnetic field and the home position both lie on the direction of the positive x-axis.

and

$$\begin{aligned}
 \vec{\mu}'_C &= K_C \iota_C(0, 0, 1) \cdot [\text{rot}_Z(\Theta)] \cdot [\text{rot}_Y(\Phi)] \\
 &= K_C \iota_C(\sin \Phi, 0, \cos \Phi) \\
 \tau_C &= \vec{\mu}'_C \times \mathbf{B} \\
 &= K_C \iota_C(0, \cos \Phi, 0)
 \end{aligned}$$

Finally, we calculate the position using Equation 40.

$$K_A \iota_A(0, -\cos \Theta \sin \Phi, -\sin \Theta) + K_B \iota_B(0, \sin \Theta \sin \Phi, -\cos \Theta) + K_C \iota_C(0, \cos \Phi, 0) = 0$$

This results in the solution

$$\Theta = \tan^{-1} \left(\frac{-K_B \iota_B}{K_A \iota_A} \right) \quad (41)$$

$$\Phi = \tan^{-1} \left(\frac{K_C \iota_C}{K_B \iota_B \sin \Theta - K_A \iota_A \cos \Theta} \right) \quad (42)$$

Thus, the position of the motor (Θ, Φ) can be calculated given the currents applied to the three coils.

4.2.2 Currents given position

Given a desired motor position, we would like to calculate the coil currents necessary to attain that position. As previously stated, because there are three coils and only two mechanical degrees of freedom, the motor position depends upon the ratio of the currents in the coils. Thus, we can fix one of the coil currents, and calculate the other relative currents. Then the scale of the three currents will control the torque with which the motor is moved to the specified position. Note that the three degrees of freedom corresponding to the three coil currents are (Θ, Φ) , and magnitude of the torque, $|\tau|$.

Assume ι_A constant. Then from Equations 41 and 42,

$$\iota_B = \frac{-K_A}{K_B} \iota_A \tan \Theta \quad (43)$$

$$\iota_C = \frac{-K_A \iota_A}{K_C} (\sin \Theta \tan \Theta + \cos \Theta) \tan \Phi \quad (44)$$

The three currents are then scaled so that the largest current magnitude is set to a maximum, ι_{max} . That is,

$$\max(|\iota_A|, |\iota_B|, |\iota_C|) = \iota_{max}$$

Figure 4.9 shows plots of the calculated currents vs. position for each of the three coils along with the actual currents produced from the calibration procedure described in Section 4.3.

4.2.3 Illustration of design constraint

Using these definitions for Θ and Φ , we can now explain the first design constraint more thoroughly which is that the permanent magnet must be positioned so that the field it defines is orthogonal to both axes of rotation of the gimbal when it is in home position. As previously stated, this constraint comes from the fact that when the motor is panned 90° (i.e., $\Phi = 90^\circ$), there is no control of the tilt (i.e., Θ is undefined).

Here we show that setting $\Phi = 90^\circ$ results in Θ being undefined. The case where K_A or K_B or K_C equals 0 represents a motor with one or more coils missing. This is a trivial case, so we assume that $K_A, K_B, K_C > 0$. Now plug $\Phi = 90^\circ$ into Equation 42, and we get

$$\tan^{-1} \left(\frac{K_C \iota_C}{K_B \iota_B \sin \Theta - K_A \iota_A \cos \Theta} \right) = 90^\circ$$

$$\frac{K_C \iota_C}{K_B \iota_B \sin \Theta - K_A \iota_A \cos \Theta} = \infty$$

This implies that $\iota_C \neq 0$ and that

$$K_B \iota_B \sin \Theta - K_A \iota_A \cos \Theta = 0$$

$$\tan \Theta = \frac{K_A \iota_A}{K_B \iota_B}$$

Equate this to the definition of $\tan \Theta$ in Equation 41 to get

$$\frac{K_A \iota_A}{K_B \iota_B} = \frac{-K_B \iota_B}{K_A \iota_A}$$

$$(K_A \iota_A)^2 = -(K_B \iota_B)^2$$

This is satisfied only when $\iota_A = \iota_B = 0$. But by Equation 41, $\tan \Theta$ is defined by $(-K_B \iota_B)/(K_A \iota_A)$ which is undefined when $\iota_A = \iota_B = 0$. Therefore, when $\Phi = 90^\circ$, Θ is undefined.

4.3 Calibration

The calculation of motor position for specified currents would be accurate in the ideal case where all three coils were perfectly symmetrical, had the same number of turns, and were the same size. In practice, none of these assumptions could be satisfied. Thus, the calculated currents give only an approximation to the actual position of the motor.

The motor must be calibrated to associate motor positions with the related set of currents that moves the motor to these positions. Only a small number of calibration points are necessary (depending on

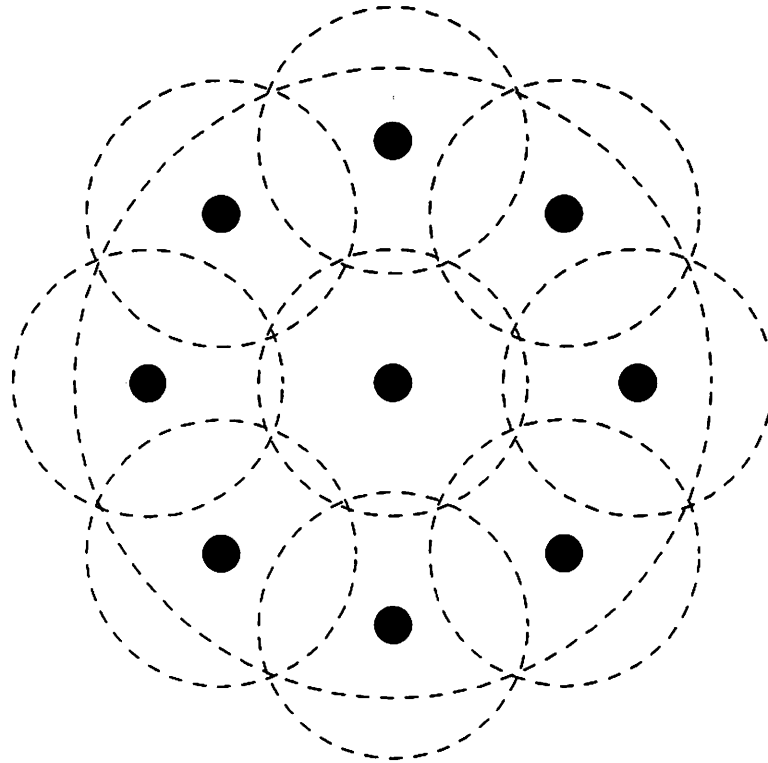


Figure 4.8: The solid dots comprise the scene used to calibrate the SPM. The nine smaller dashed circles are not part of the calibration pattern, but represent the field of view of the camera when centered on each dot. The single large dashed circle is also not in the scene, but represents the workspace of the motor. The entire workspace of the motor is covered by the small dashed circles.

the desired precision of the motor) since points between the calibrated positions can be approximated with linear interpolation.

We developed a procedure for automatic calibration of the SPM. It is based on image feedback from a camera mounted on the rotor of the motor. It assumes that a calibrated image sensor and lens are used, i.e., that it is known how many degrees each pixel subtends, and that this is constant throughout the field-of-view. The calibration algorithm uses a scene of black dots on a white background. For each motor position that is to be calibrated, the algorithm moves the motor approximately to that position using the calculated currents of Section 4.2. The algorithm analyzes the image and uses the position of the relevant dot to calculate the actual position of the motor. It then associates this position with the coil currents and stores it in a look-up-table, *calib*.

The workspace of the motor is larger than the field of view of the camera, so the scene must consist of more than one dot. To make the calibration procedure as automatic and as easy to use as possible, we do not want to measure the scene externally. Instead, the algorithm should work with only approximate knowledge of the positions of the dots. The only requirement is that there are enough dots so that the entire workspace of the motor is covered, and that it is possible to view the dots in pairs so that the position of one dot can be determined by its position relative to that of a previously determined one. In this way, the position of each dot can be precisely measured automatically by the calibration procedure. The scene used to calibrate the prototype motor is shown in Figure 4.8.

The algorithm, *calibrate*, calibrates the SPM with a multiple dot scene and it uses the algorithms

center_dot, **find_dot** and **centroid**. It assumes the scene is arranged so that the center dot is initially approximately centered in the motor workspace, and thus in the camera field of view when the motor is in home position. A data structure is used supplying information about the arrangement of the scene. The algorithms are specified for space-variant images as defined in Chapter 1. The constants are defined in Appendix C. For each dot in the calibration pattern, the data structure contains:

dot_scene table	
Variable	Description
motor_init	(Θ, Φ) position of where to initially move the motor for this point. This position should be defined so that the new dot and a previously calibrated dot are both in the field of the view of the camera when the motor is moved to this position.
new_coord	Estimated position of new dot in scene at motor position, motor_init
old_coord	Estimated position of previously calibrated dot in scene at motor position, motor_init
old_id	Index into this data structure to identify which previously calibrated dot is used

In these algorithms, the dot is defined as a single connected component whose values are below a threshold that is assumed to have been previously computed. Three routines are called that are not explicitly defined here. **soap_bubble** uses a standard relaxation technique to fill in the undefined values of the calibration table, **calib**. **median_filter** applies a standard 5×5 median filter to **calib** to eliminate bad points. **smooth** applies a standard 3×3 smoothing filter to **calib** to smooth noise.

The prototype SPM has been calibrated using **calibrate** and the scene depicted in Figure 4.8. The results are shown in Figure 4.9 along with the ideal case where the currents for each coil are shown over a workspace of $\pm 30^\circ$ in Θ and Φ .

Algorithm: calibrate

Function: This calibrates the SPM, creating a look-up-table associating coil currents with specific motor positions. This is accomplished automatically, imaging a scene with multiple dots. For each position to be calibrated, the motor is moved to that position using the calculated values. The dot is then searched for starting at its estimated position. Its actual position is determined and then associated with the motor position and entered in the calibration table.

Input: dot_scene table

Output: Calibration table: calib[Θ , Φ] holds currents for each coil at each calibrated motor position.

```

For each calibration point, [k,l], in calib
    calib[k,l] ← undefined
End For
Move motor to (0,0), i.e., home position
For each dot in the scene, dot_index
    Move motor to motor_init[dot_index]
    Acquire image
    { Find absolute position of current dot }
    zero_pt[dot_index] ← find_dot(new_coord[dot_index])
    If not the first dot Then
        old_dot_position ← find_dot(old_coord[dot_index])
        zero_pt[dot_index] ← zero_pt[dot_index] - old_dot_position +
            zero_pt[old_id[dot_index]]
    End If
    (Center $\Theta$ , Center $\Phi$ ) ← center_dot(new_coord[dot_index])
    For each motor position, ( $\Theta$ ,  $\Phi$ ), to be calibrated with the new dot
        Move motor to estimated position, ( $\Theta$ ,  $\Phi$ ), using calculated table
        Acquire image
        { Convert motor position to image position }
        estimated_dot_position ← (NROWS/2 - (Center $\Theta$  -  $\Theta$ )/DEG_PER_PIXEL,
            NCOLS/2 - (Center $\Phi$  -  $\Phi$ )/DEG_PER_PIXEL)
        new_dot_position ← find_dot(estimated_dot_position)
        { Convert new_dot_position to calib indices }
        [k,l] ← DIM_CALIB/2 + (new_dot_position - zero_pt[dot_index]) ×
            DEG_PER_PIXEL × DIM_CALIB / DEG_OF_WORKSPACE
        If calib[k,l] = undefined Then
            calib[k,l] ← -1 × (Motor currents for each coil at this position)
        Else
            calib[k,l] ← (calib[k,l] - (Motor currents for each coil
                at this position))/2
        End If
    End For
End For
calib ← soap_bubble(calib)
calib ← median_filter(calib)
calib ← smooth(calib)

```

Algorithm: `center_dot`**Function:** This moves the motor to center the dot in the image. It is an iterative process, moving the motor to center the dot using the approximate values of the calculated motor position look up table.**Input:** Starting motor position, (Θ, Φ)
Estimated dot position, p_1 , in logmap image, L **Output:** Motor position that results in centered dot.

DoMove motor to (Θ, Φ)

Acquire image

 $(\text{dot_row}, \text{dot_col}) \leftarrow \text{find_dot}(p_1)$ $\Theta \leftarrow \Theta + (\text{dot_row} - \text{NROWS}/2) \times \text{DEG_PER_PIXEL}$ $\Phi \leftarrow \Phi + (\text{dot_col} - \text{NCOLS}/2) \times \text{DEG_PER_PIXEL}$ $p_1 \leftarrow (\text{NROWS}/2, \text{NCOLS}/2)$ **While** dot not centered**return** (Θ, Φ)

Algorithm: `find_dot`

Function: This finds the dot in the image starting at the specified location, searching outward.
When the dot is found, its centroid is computed and returned.

Input: Starting image location, p_1 , in logmap image, L

Output: Centroid of found dot in TV image coordinates.

For each pixel $p_2 \in L$

`visited`[p_2] $\leftarrow$ **FALSE**

End For

`visited`[p_1] $\leftarrow$ **TRUE**

`front` $\leftarrow$ 0

`back` $\leftarrow$ 0

While ($(L(p_1) \geq \text{threshold})$ **AND** (`front` < `NPIXELS`) **AND** ($a(p_1) > 0$)) **Do**

For each neighbor $p_2 \in \mathcal{N}(p_1)$

If (`visited`[p_2] = **FALSE**) **Then**

`visited`[p_2] $\leftarrow$ **TRUE**

`recurse`[`back`] $\leftarrow$ p_2

`back` $\leftarrow$ `back` + 1

End If

End For

$p_1 \leftarrow$ `recurse`[`front`]

`front` $\leftarrow$ `front` + 1

End While

`return`(`centroid`(p_1))

Algorithm: centroid

Function: This computes the centroid of a dot in an image given a point in the dot. It assumes the dot is a single connected component of pixels below the threshold. It computes the centroid in the logmap image, but weights the pixels by their area in TV pixels to get the true centroid. It returns the result in TV image coordinates.

Input: Starting image location, p_1 , in logmap image L

Output: Centroid of found dot in TV image coordinates.

```

If ( $a(p_1) = 0$ ) Then
    print("Error: dot searched for at invalid point")
    return(error)
End If
For each point  $p_2 \in L$ 
    visited[ $p_2$ ]  $\leftarrow$  FALSE
End For
visited[ $p_1$ ]  $\leftarrow$  TRUE
 $p_3 \leftarrow 0$ 
front  $\leftarrow 0$ 
back  $\leftarrow 0$ 
total_weight  $\leftarrow 0$ 
 $\mathcal{P}_a \leftarrow 0$                                 { TV image point }
 $\mathcal{P}_{center} \leftarrow (NROWS/2, NCOLS/2)$ 
Do
    For each neighbor  $p_2 \in \mathcal{N}(p_1)$ 
        If ((visited[ $p_2$ ] = FALSE) AND ( $L(p_2) < \text{threshold}$ )) Then
            weight  $\leftarrow L(p_2) \times a(p_2)$ 
             $\mathcal{P}_a \leftarrow \mathcal{P}_a + \text{weight} \times \mu(p_2)$     { Convert  $p_2$  to TV coordinates }
            total_weight  $\leftarrow$  total_weight + weight
            visited[ $p_2$ ]  $\leftarrow$  TRUE
            recurse[back]  $\leftarrow p_2$ 
            back  $\leftarrow$  back + 1
        End If
    End For
     $p_1 \leftarrow$  recurse[front]
    front  $\leftarrow$  front + 1
While (front  $\leq$  back)
If (total_weight > 0) Then
     $\mathcal{P}_b = \mathcal{P}_a / \text{total\_weight} - \mathcal{P}_{center}$ 
Else
     $\mathcal{P}_b = (0, 0)$ 
End If
return( $\mathcal{P}_b$ )

```

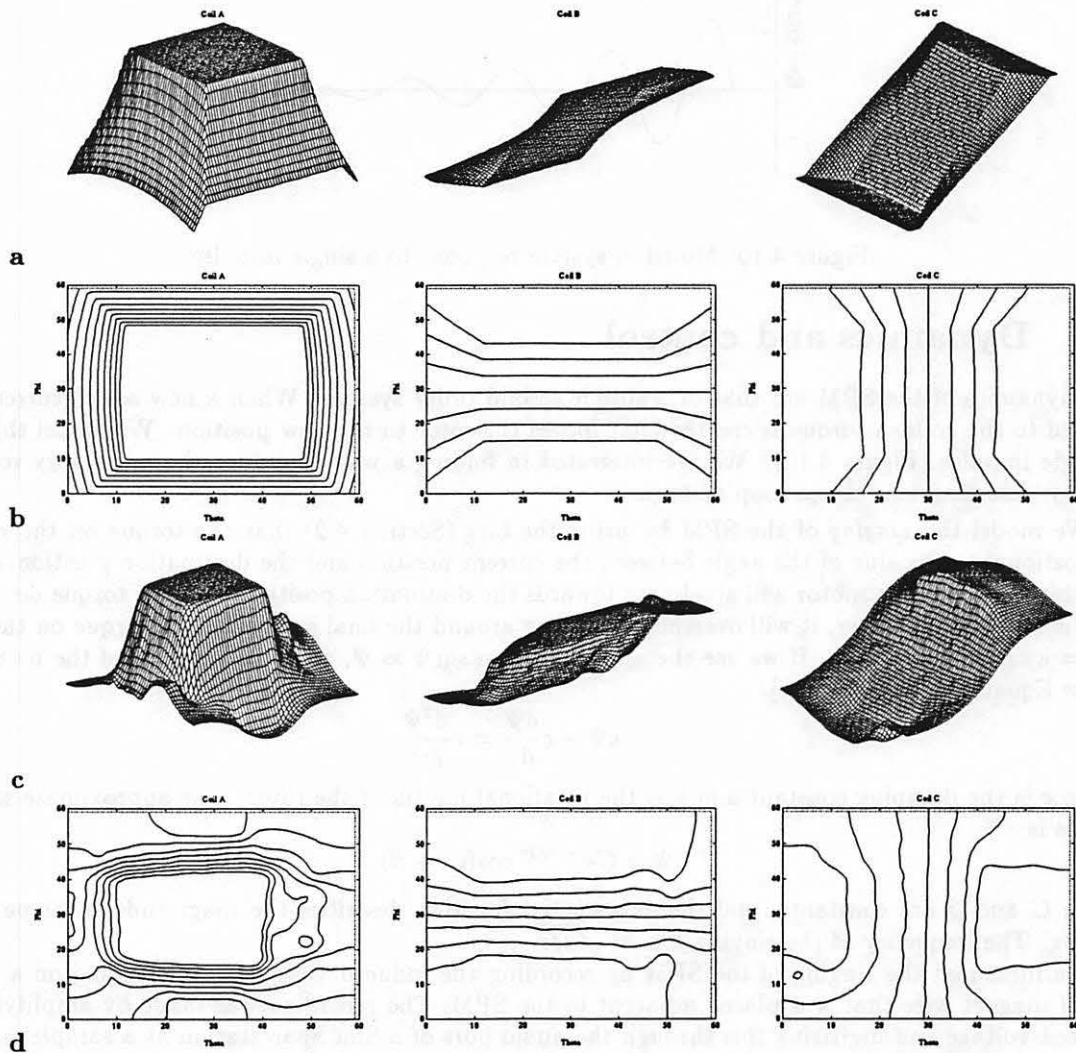


Figure 4.9: a) Theoretical calculated currents vs. position for each of the three coils shown as a mesh plot. b) The same as a) shown as a contour plot. c) Calibrated currents vs. position for each of the three coils shown as a mesh plot. d) The same as c) shown as a contour plot. The data shown here is for the $\pm 30^\circ$ workspace in (Θ, Φ) .

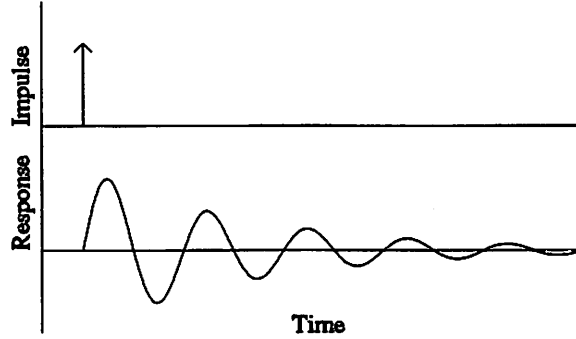


Figure 4.10: Model of system response to a single impulse.

4.4 Dynamics and control

The dynamics of the SPM are that of a simple second order system. When a new set of currents are applied to the coils, a torque is created that moves the rotor to the new position. We model this with a single impulse (Figure 4.10). We are interested in finding a way to reduce the oscillatory response without resorting to a closed-loop system.

We model the *ringing* of the SPM by using the fact (Section 4.2) that the torque on the rotor is proportional to the sine of the angle between the current position and the destination position. Let us call this angle Ψ . The motor will accelerate towards the destination position with the torque decreasing as it is reached. However, it will overshoot and ring around the final position. The torque on the rotor is $\tau = \kappa \sin \Psi$, for some κ . If we use the approximation $\sin \Psi \approx \Psi$, then the position of the motor will follow Equation 45 [23, P. 228].

$$-\kappa\Psi - c \frac{d\Psi}{dt} = r \frac{d^2\Psi}{dt^2} \quad (45)$$

where c is the damping constant and r is the rotational inertia of the rotor. The approximate solution to this is

$$\Psi = C e^{-ct/2r} \cos(\omega t + \beta) \quad (46)$$

where C and β are constants, and the exponential function describes the magnitude envelope of the ringing. The frequency of the ringing is $\omega \approx \sqrt{\kappa/r}$.

We measured the ringing of the SPM by recording the induced voltage of the ringing on a pickup coil of magnet wire that was placed adjacent to the SPM. The recording was made by amplifying the induced voltage and digitizing this through the audio port of a Sun Sparcstation at a sample rate of 8 KHz. Because the SPM is controlled with Pulse-Width Modulation (PWM), the pickup coil not only records the ringing (at approximately 60 Hz), but also records the PWM signal (at 1 KHz). The recorded signal is filtered with an FFT to isolate the ringing from the PWM in the induced signal. An example of the ringing shows the result of moving the SPM, uncontrolled, about 20° (Figure 4.11).

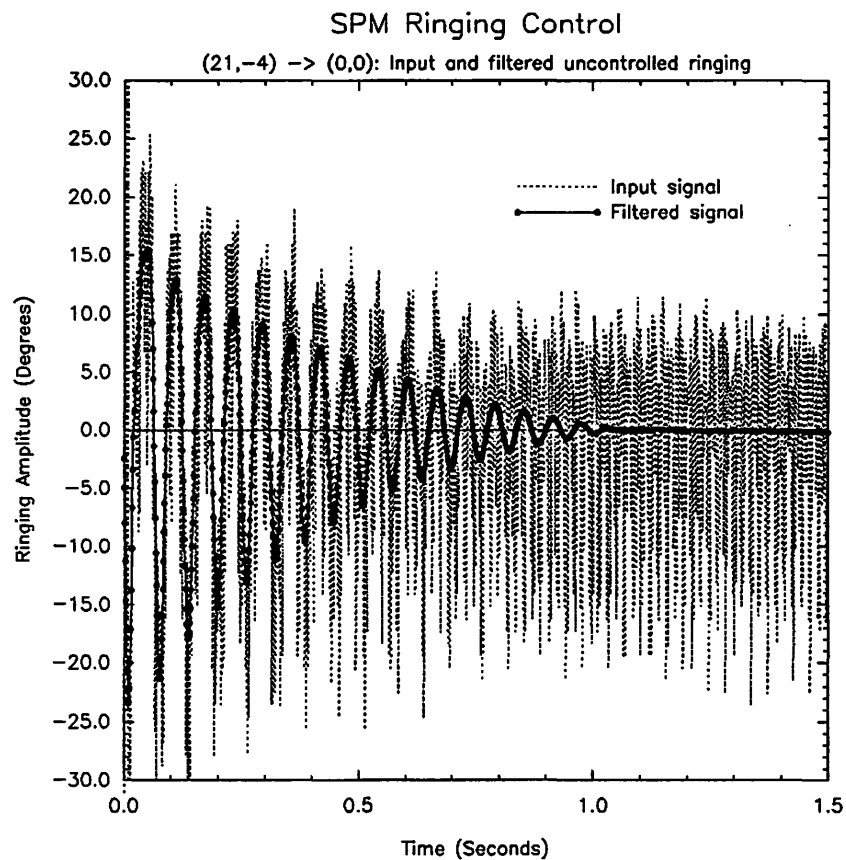


Figure 4.11: Uncontrolled ringing measurement taken by induced voltage on a separate pickup coil. The input signal (combination of ringing and PWM) is shown along with the filtered signal that isolates the ringing.

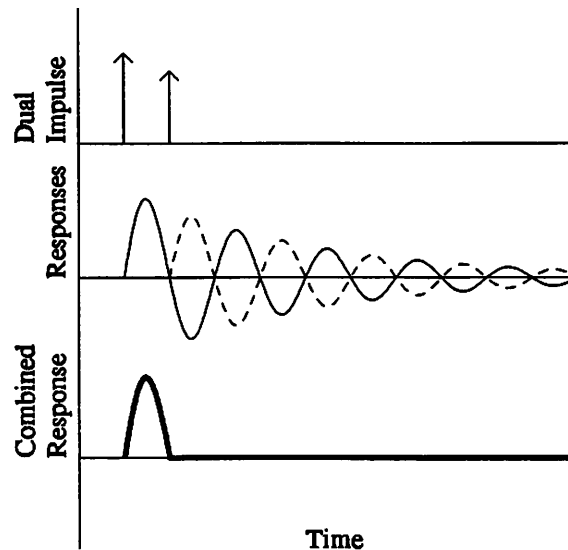


Figure 4.12: Ideal model of system response to a dual impulse.

This ringing can be greatly reduced by various open-loop control methods. Perhaps the simplest strategy is to move the motor from the initial position to the destination position in small decreasing increments at fixed time intervals. This way, maximum motor velocity is traded off for control. Including the time it takes for ringing to stop, the controlled approach yields a faster average velocity. This method is implemented by moving a fixed percentage of the distance between the current position and final position at each step. This will decrease the motor movement with each step. The motor velocity can then be controlled with either the percentage movement of each step, or the time interval between steps. We use a constant time interval, and vary the percentage movement of each step to experiment with different speeds. We call this the *fixed percentage* control method.

A second slightly more complicated open-loop control strategy based on traditional stepper motor control theory, and also examined by Singer and Seering [50], yields better results than the fixed percentage method. The idea, called the *two step* method, is based on the fact that the SPM rotor oscillates with an approximately constant period. We give two impulses (Figure 4.12) 180° out of phase with each other so as to cancel the ringing. We implement this by first stepping the motor to a point midway between the initial and destination positions (determined from the calibration results of Section 4.3). We then wait for the point where the rotor velocity is zero at which point we apply a second step to hold the rotor at this position. The energy of the ringing will be largely dissipated, and the rotor will be at the destination position. Although we don't know exactly when to apply the second step, we can estimate it by waiting half the period of oscillation. Because we model the system with a one-dimensional pendulum when it actually is a two-dimensional problem, our estimation is not exact, but is a close approximation. In practice, because neither the midpoint nor the time at which the currents are changed are exactly correct, not all of the ringing energy is dissipated, and a little ringing of the same period remains, but with vastly reduced amplitude (Figure 4.13). This approach is unusual in that it takes a constant time to move any distance, where the time is half the period of oscillation. A sample result of this approach is shown in Figure 4.14.

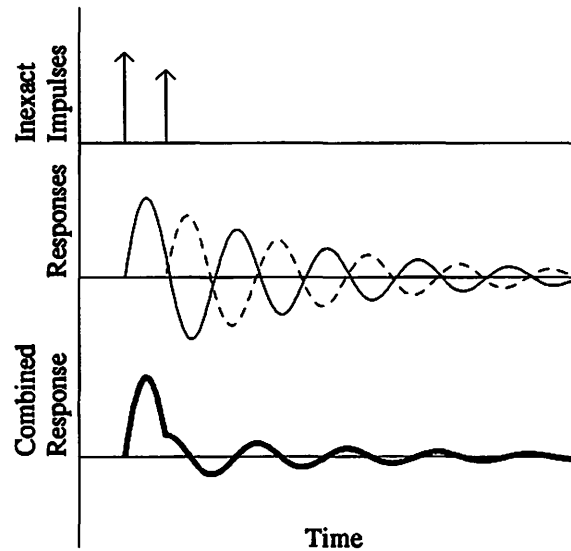


Figure 4.13: Model of system response with inexact timing to a dual impulse.

A problem with the two-step control strategy is that the system response is not very robust, i.e., a small uncertainty in the midpoint or delay timing results in relatively large uncontrolled ringing. We compensated for this imprecision by various combinations of the previous two control strategies. The first approach is to apply the two-step method twice, first moving to the midpoint, and then to the destination point. This method theoretically also takes constant time, which is twice as long as the two-step method. We call this approach the *dual two-step* method.

A final variation, called the *multi two-step* method, is to apply the two-step method multiple times with logarithmically decreasing steps. The motor is always moved half way between its current position and the destination position with the two-step method. This takes time $O(\log(\Psi))$ where Ψ is the total angular distance travelled.

We are currently investigating other control approaches advocated by Singer and Seering involving three impulses to make a more robust strategy. Results from the open-loop control strategies we tested are presented in Section 4.7.2.

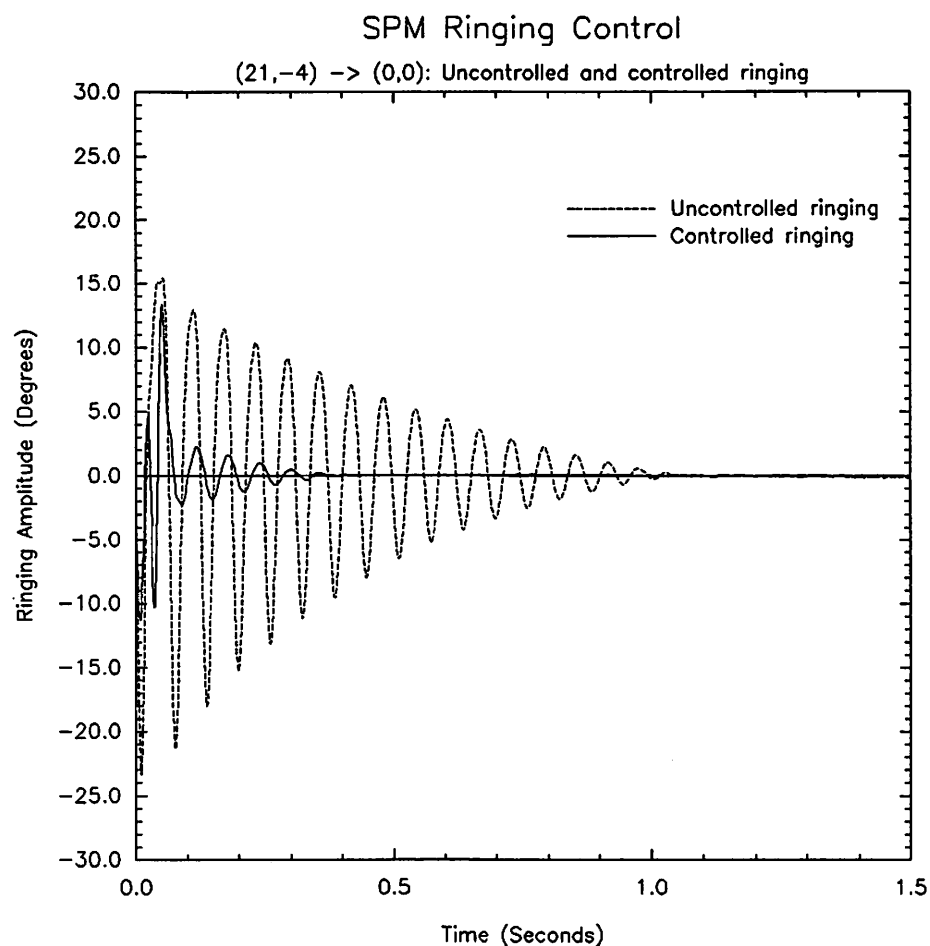


Figure 4.14: Measurement of two-step open-loop control strategy to reduce SPM ringing. Results are shown for the same movement of 21.4° with and without control. The motor currents are changed to hold the motor at the final position when the motor is expected to have zero velocity. Because the midpoint and timing are not exactly correct, the ringing is not completely eliminated.

One of the motivating factors behind the development of the SPM was to make a motor that could be run by an open-loop controller, and we have accomplished this task. However, if higher performance is needed, a velocity or position sensor could be added to run the motor with a closed-loop control strategy. This would allow the two-step control strategy to be applied more accurately, as the zero-velocity point would be known exactly. Not only would this increase the speed of the motor, but it would also increase its accuracy. The motor's supply currents could be 90° out of phase with the position, like in traditional DC motors. This would result in always driving the motor at its maximum torque and thus the torque would no longer be dependent on the difference between the current and destination position. No closed-loop control strategies have been implemented yet.

4.5 Design Issues

We want to design the SPM to maximize its torque while minimizing its size and power usage. There are several parameters that we have control over in this design. The diameter of the wire and the area, number of turns, and voltage across the coil as well as the magnetic return path are the parameters that most affect the design, and are the ones that we will discuss here.

The first issue is the magnetic return path. Our magnetic circuit consists of a permanent magnet that generates magnetic field lines with a geometry dependent on the type of material in its immediate vicinity. Magnetic circuits can be described analogously to electric circuits. With electric circuits, different materials have different resistances and the current will follow the path of least resistance. In magnetic circuits, different materials have different permeability and the magnetic field will follow the path of greatest permeability. Because there is only torque on a current loop if the magnetic field goes through the loop and returns outside the loop, it is critical that the geometry of the motor is designed so that the permanent magnetic field lines return outside the coil. Air has very low permeability while iron has very high permeability. By adding an iron return path outside the motor, more magnetic field lines will enclose the coil because the magnetic field lines will follow the path of highest permeability, resulting in more torque. A simulation of the magnetic circuit for the external coil SPM with and without an iron return path is shown in Figure 4.15.

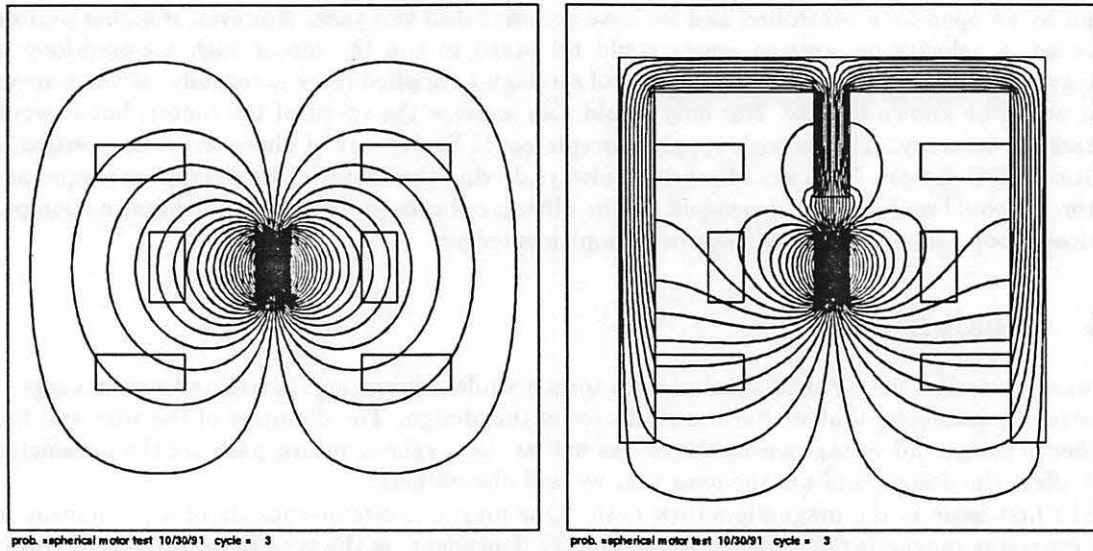


Figure 4.15: Simulation of magnetic circuit for the SPM (left) without and (right) with an iron magnetic return path. The permanent magnet is in the center. A camera mounted on the magnet would be looking towards the bottom of the page. The rectangles represent the coils of the motor. It can be seen that without the magnetic return, only a small percentage of the magnetic field returns outside the coils, but with the magnetic return, an efficiency of nearly 50% is achieved.

Now let us investigate the parameters of voltage (V), the number of turns in the coil (N), the area of the coil (A), and the diameter of the wire (D). We know that $\tau \propto N\iota AB$, and power, $P = V\iota$. We want to maximize the ratio τ/P while maintaining a minimum acceptable torque and maximum acceptable power. The resistance of the wire is proportional to the number of turns of the coil and the area of the coil, and it is inversely proportional to the square of the diameter of the wire, so we can write

$$\mathcal{R} \propto \frac{NA}{D^2}$$

Because $\iota = V/\mathcal{R}$, we can rewrite the equation for current as $\iota \propto (VD^2)/(NA)$. Finally we can rewrite the equations for torque, power and their ratio using only the input parameters:

$$\tau \propto VD^2 \cdot B \quad (47)$$

$$P \propto VD^2 \cdot \frac{V}{NA} \quad (48)$$

$$\frac{\tau}{P} \propto \frac{BNA}{V} \quad (49)$$

From these relationships, we can see that we want to maximize B and N while minimizing V . In doing so, we must also maintain our torque and power requirements and adjust V , D , and A accordingly. The values of these parameters for our prototype SPM are reported in Section 4.7.

4.6 Controller description

The SPM is controlled by limiting the amount of current in each coil. This can be done either by regulating the voltage on the coils, by using a constant voltage with a Pulse-Width Modulation (PWM) technique, or by using a variable current

source. The PWM method is preferred for ease of use, although the variable current source technique has the advantage that the currents, and thus the motor position, are independent of the temperature of the motor.

The SPM controller we use is based on a Motorola M68332 microcontroller that is capable of creating PWM signals. A Unitrode L293N power driver supplies the current for the motor. PWM means driving the motor with a square wave of constant amplitude and period with a variable duty cycle. The percentage of the period where the supply is high is called the duty cycle. A duty cycle of 0 gives no current to the motor while a duty cycle of 100 gives maximum current. The inertia and inductance of the coil effectively smooths the PWM supply. The polarity of the voltage on the coil also needs to be controlled in order to position the motor everywhere on the unit sphere. The power driver allows us to control the voltage polarity.

4.7 Prototype spherical pointing motor

The prototype SPM (Figure 4.1) is $4 \times 5 \times 6$ cm, weighs 160 grams and is capable of actuating a 6 gram load. Its total workspace is approximately 60° in both the pan and tilt directions. The permanent magnetic field is estimated at 1 Tesla with an efficiency of about 5%. The number of turns in the coil is estimated. The maximum torque is computed here, where $\Psi = 90^\circ$. This table shows the parameters of the motor along with its calculated torque and power usage. The actual torque and power usage have not been measured.

Coil	Parameter	Value
Inside	N	630
	D	30 gauge (= .000254 m)
	A	0.000730 m^2
	V	15 Volts
	R	21Ω
	τ	$.016\text{N} \cdot \text{m}$ or $2.3\text{oz} \cdot \text{in}$
	P	11 Watts
Middle	N	500
	D	30 gauge
	A	0.000993 m^2
	V	15 Volts
	R	26.0Ω
	τ	$.014\text{N} \cdot \text{m}$ or $2.0\text{oz} \cdot \text{in}$
	P	9 Watts
Outside	N	420
	D	30 gauge
	A	0.00130 m^2
	V	15 Volts
	R	22Ω
	τ	$.019\text{N} \cdot \text{m}$ or $2.6\text{oz} \cdot \text{in}$
	P	10 Watts

4.7.1 Accuracy and precision measurements

The precision and accuracy of the SPM were measured by reflecting a laser diode off a reflective surface attached to the rotor. The reflected beam's movement is measured on a wall several feet away as described below.

The precision of the motor is defined as the angular distance between adjacent positions. If the controller consisted of an analog variable current source, the motor's precision would be infinite. As we are using a digital PWM controller, the precision with which we can control the motor is limited by the step size of the PWM duty cycle. The step size is dependent on the frequency of the PWM. Lower frequencies have more precise control of the step. However, lower frequencies also introduce a choppiness that vibrates the motor. We use a 1KHz frequency and are able to specify 4,000 steps in the duty cycle (i.e., 4,000 different duty cycles). As the duty cycle of the PWM on a single coil with a specified polarity changes between 0 and 100, the motor will turn at most 45° . Thus, we can position the motor to steps no smaller than $45^\circ/4000 = 0.011^\circ$ or 0.68 minutes of arc.

The accuracy of the motor is defined as the repeatability of the motor. If the motor were balanced so that gravity was not a factor, then its accuracy would be dependent only upon the friction of the bearings. As there is no iron core, there is no hysteresis, and thus it is absolutely repeatable. Because the motor, in fact, is not perfectly balanced, the motor position is not constant at different orientations to gravity, but at a fixed orientation, the accuracy is dependent only upon the friction of the bearings.

We measured the motor's accuracy with a laser setup (Figure 4.16) and found the motor to be accurate to 0.15° . The camera mounted in our motor returns 192×165 pixels with a 33.6° horizontal field of view. This is equivalent to 0.175° per pixel. For our camera and lens, the SPM is thus accurate to about one pixel.

The procedure for measuring motor accuracy is as follows. The motor is set at a fixed position with the rotor positioned at the place to be measured. The laser is then oriented so that the reflected beam hits the wall at a right angle. This point on the wall is recorded. The rotor is moved away and then back to the original position. The distance between the new reflected laser position and the original position is measured, and the accuracy of the motor is calculated according to Equation 50.

The measurement setup is illustrated in Figure 4.16. The motor is moved Θ degrees. The distance from the motor to the wall is f , and e is the distance between the first and second laser position on the wall. Φ is the angular difference between the first and second motor positions. The solid lines represent the first motor position, and the dashed lines represent the second. The relation between the laser point movement and the accuracy of the motor is

$$\Phi = \frac{1}{2} \tan^{-1}\left(\frac{e}{f}\right) \quad (50)$$

4.7.2 Velocity measurements

We measured the average velocity of the motor for point to point motions since we have no ability to measure instantaneous velocity. We recorded the velocity using the technique described in Section 4.4 for measuring the ringing. The motor controller was programmed to accept motion commands with a specified control strategy over an RS-232 serial port. A program running on a Sun Sparcstation controlled the SPM and automatically recorded and analyzed the results. For each control strategy, ten measurements were made at each of ten different angular movements. The averaged results with error bars showing the variance of each experiment are shown in Figures 4.17, 4.18, and 4.19.

A summary of the control strategy measurements follows:

No control Base line for comparing control strategies.

Fixed percentage The motor is moved a fixed percentage of the distance between its current position and the destination position with a fixed delay between each movement.

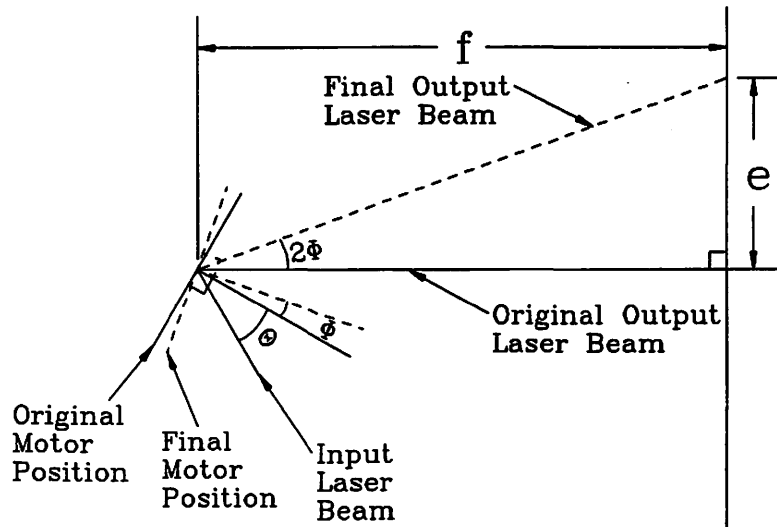


Figure 4.16: Illustration of our experimental setup for measuring motor accuracy. The laser beam is reflected off the rotor onto a flat surface. The motor is then moved to a new position and back to the original position. The new laser beam position is compared with the original position to see how accurately the motor returned to the original position.

By making the percentage of movement per step small enough, the ringing can be completely eliminated at the price of velocity. Results shown here are for a percentage selected such that the total time to move was minimized.

Two-step The motor is moved to the midpoint between the initial and destination positions. After a delay equal to half the period of the natural motor oscillation, the motor is moved to the destination position.

The midpoint and delay used are an approximation to the ideal values. These result in good results with a fixed time to move any distance, and a small amount of ringing at the end of the movement. The ringing at the end of the motion does not depend on the amplitude of the motor movement. Rather, it depends on the inaccuracy of the midpoint and delay calculations. A calibrated look-up-table approach would be necessary to eliminate the ringing completely.

Dual two-step The two-step method is used to move first to the midpoint, and then to the endpoint.

This theoretically takes twice as long as the single two-step method, but actually results in slightly better results because the resultant ringing is somewhat reduced.

Multiple two-step The two-step method is used multiple times, always moving halfway between the current position and the destination position.

This theoretically takes time $O(\log(\Psi))$ where Ψ is the angular distance between the initial and destination positions. In practice, it produced very similar results to the dual two-step method.

Our measuring apparatus was fairly noisy, resulting in the high variance reported. It is clear that all four control strategies provided substantial improvement over the uncontrolled performance. Of these, the two-step methods are a little better than the fixed percentage approach. Because of the noise in our measurements, it is not clear which two-step method is the best.

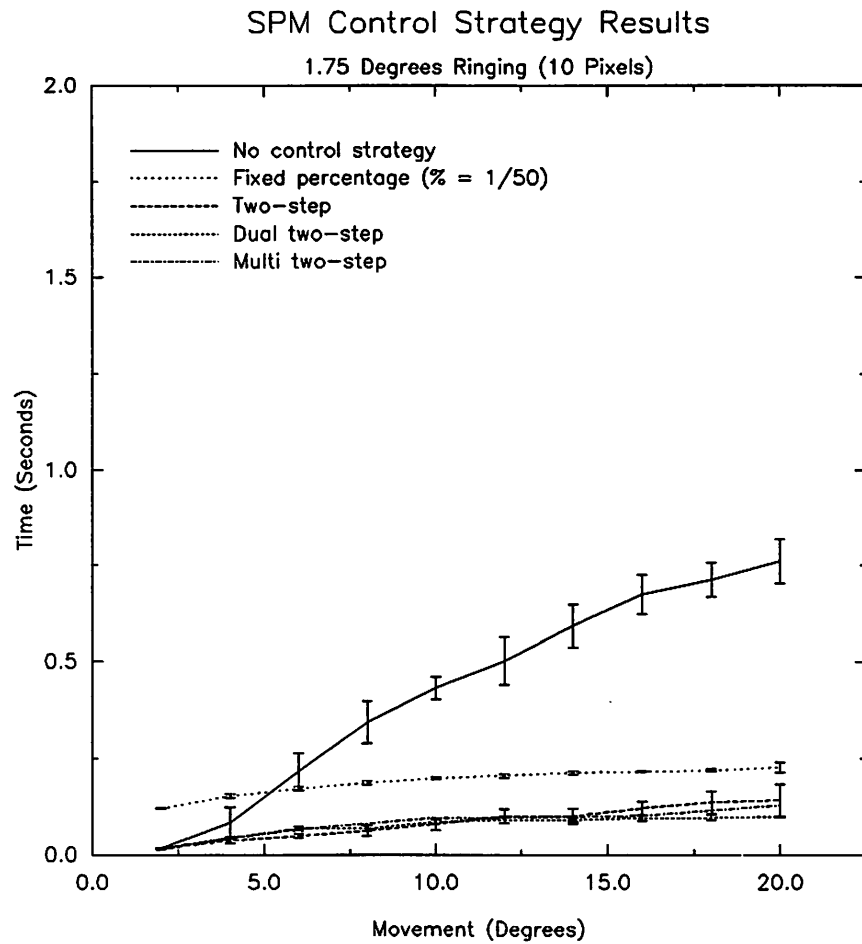


Figure 4.17: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 1.75° (10 pixels in our camera). Control strategies are discussed in the text.

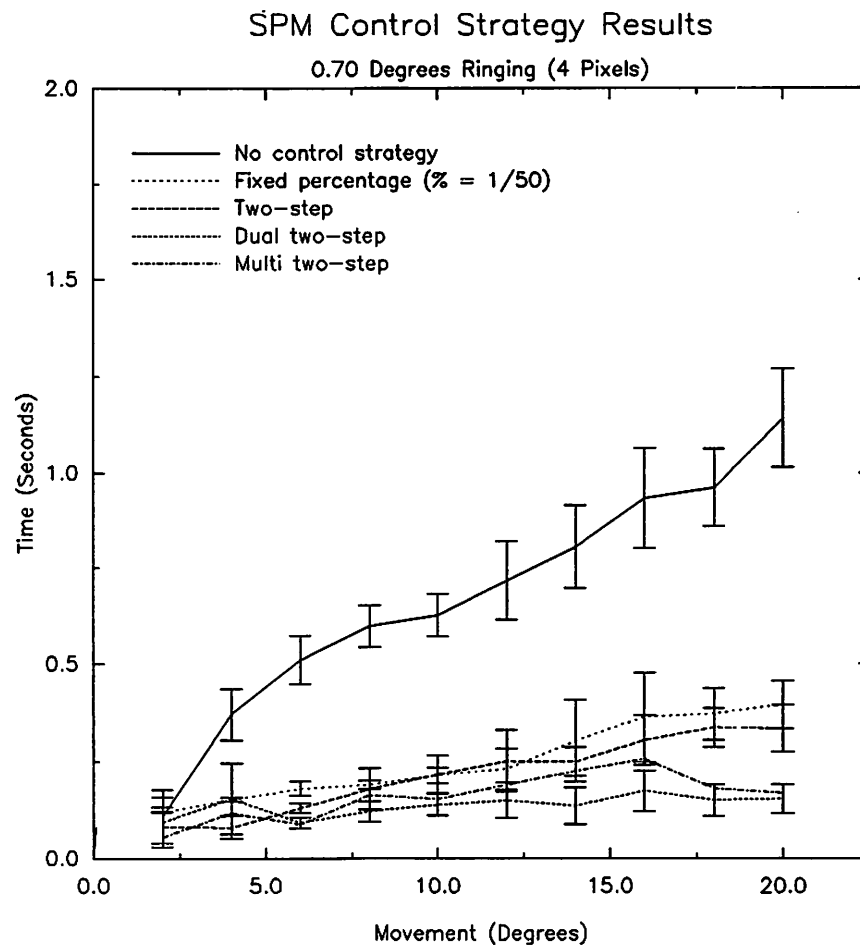


Figure 4.18: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 0.70° (4 pixels in our camera).

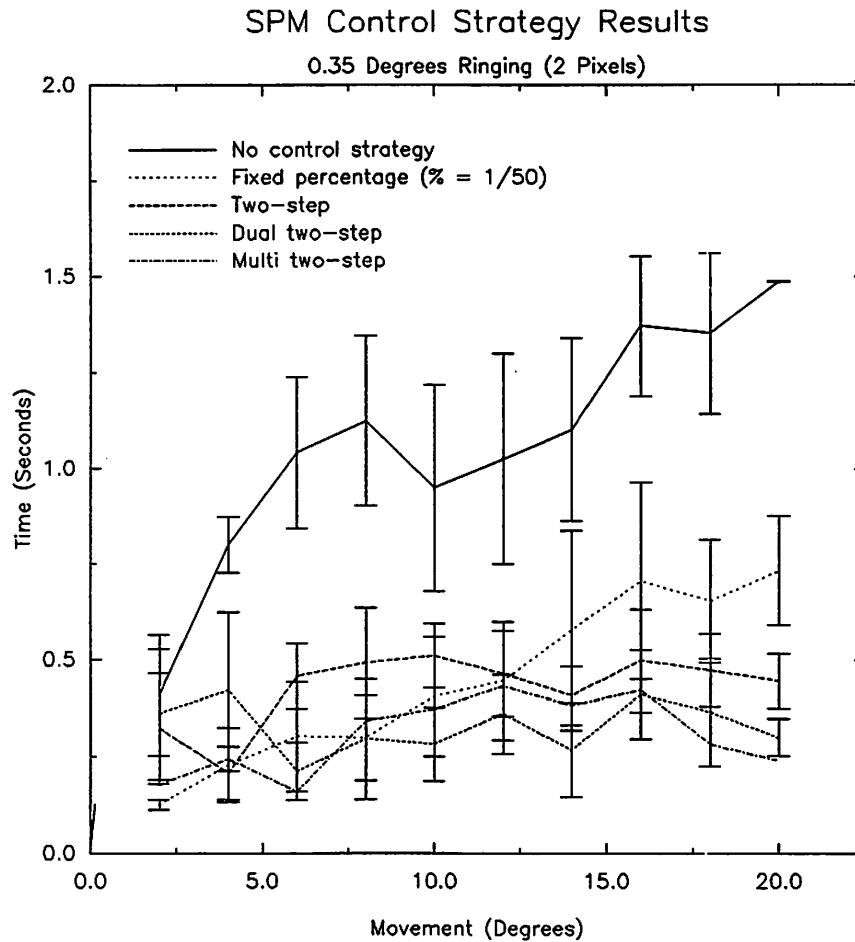


Figure 4.19: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 0.35° (2 pixels in our camera).

The results are summarized in the following table. For each control strategy, the time to move 2 degrees and to move 20 degrees are given for each of three ringing requirements. The times are specified in milliseconds.

Ctrl Method	.35° ringing	.70° ringing	1.75° ringing
No control	408/1480	110/1140	18/760
Fixed percentage	127/734	120/395	120/227
Two-step	332/445	83/334	16/143
Dual two-step	359/298	96/155	18/100
Multi two-step	179/240	56/171	17/128

It should be stressed that the results from these two-step open-loop control strategies were obtained with approximate calculated values for the mid-point and timing values. We fully expect that with a calibrated look-up-table approach, as with the position calibration of Section 4.3, we will obtain substantially better results.

Chapter 5

Results

5.1 System architecture

Cortex-I consists of a camera, actuator, camera driver, motor driver, video interface, telephone interface, and processors. The three processors communicate via high-speed serial interface. The flow of information between the modules (Figure 5.1) starts with first DSP board that controls the camera driver, acquires the uniform sensor image, and maps it to a logmap image. It sends the image to the M68332-based microcontroller board which performs image processing, and controls the motor. The microcontroller board sends the resulting image to the second DSP board that maps the logmap image to the inverse map image and creates an RS-170 video signal for display on a TV monitor. The microcontroller board also sends the image to an optional third DSP board that is used as a video transmitter that can send 4 frames per second of logmap images over a standard voice-grade telephone line [56].¹

Each of the processors in our prototype is a general purpose computer, so the programming model bears great resemblance to that for embedded control systems on general purpose processors. One of the key elements in this model is a distinction between a development system and a target system. We initially developed our applications on a Sun Sparcstation based system. This development system simulates the target system to a high degree: it combines a robotic pan-tilt actuator made by Klinger Scientific, a Sony XC-77RR camera and an Analogic video digitizer, with software tools to simulate a variety of sensor geometries.

There are four stages to developing an application.

1. We design a prototype program on the development host. At the first stage there are few memory, timing or numeric precision restrictions. The commercial camera and actuator provide a high degree of reliability.
2. Development continues on the host, but we use the camera and actuator of Cortex-I. The host controls the SPM by sending commands over an RS-232 serial line, and receives the logmap image from the miniature camera, digitizing the video signal output. This allows the the same software to be tested using the new hardware.
3. The application is debugged on the target hardware using the host as a terminal. Since *C* language cross-development environments are available for both the DSPs and the microcontroller used in this system, we develop *C* code on the Sun host, then download it to the target system. The code must account for the more stringent hardware restrictions such as integer arithmetic, less memory, and less processing power. Such restrictions are a result of the inherent tradeoff between a highly flexible, programmable development system and an inexpensive target system.

¹Patent application #07/802,320, entitled "Telephone Line Picture Transmission" was submitted to the U.S. Patent Office in 1991.

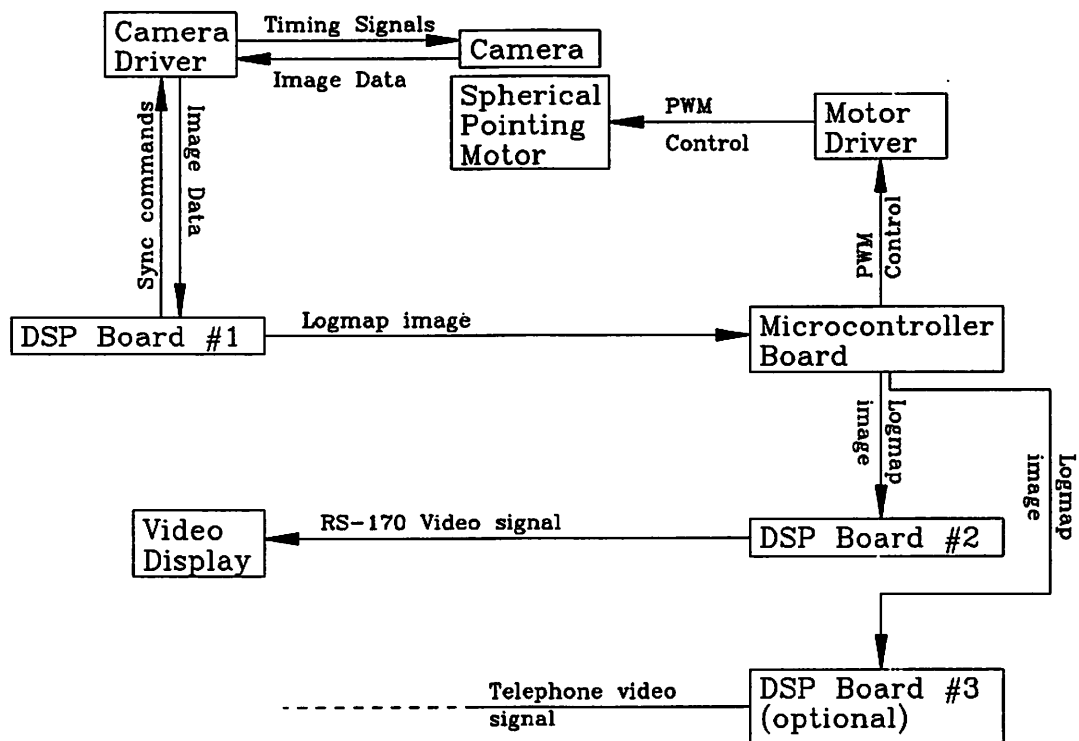


Figure 5.1: System architecture overview.

4. We burn the application into the ROM of the target system at which point we can take the target system “into the field”. Some applications, like an attentional OCR program for reading license plates [38] are at stage two, while many simpler image processing algorithms, such as motion tracking are at stage four.

5.2 Image processing

We have developed several algorithms that are burned into ROM in Cortex-I, and are transportable. Some of them use the connectivity graph defined in Section 1.4 of Chapter 1.

Image binarization using the mean pixel value for a threshold runs at 22 frames per second (fps). The mean pixel value is given by

$$\bar{p} \leftarrow \frac{1}{N_{\text{PIXELS}}} \sum_{p \in L} L(p)$$

Then the binarized image, L_B , is defined by

$$L_B(p) \leftarrow \begin{cases} 0 & \text{if } L(p) \leq \bar{p} \\ 255 & \text{otherwise} \end{cases}$$

Image smoothing runs at 3 fps. The smoothed image, L_S , is defined by

$$L_S(p) \leftarrow \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} L(q)$$

We defined a simple edge operator that runs at 2 fps. The edge image, L_E , is defined by

$$L_E(p) \leftarrow \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} (L(p) - L(q))^2.$$

Finally, a connected components algorithm runs at only 1 fps, illustrating both the generality of the programming model and the limitations of the microcontroller in the current prototype. The connected components problem begins with a given predicate F , for example the foreground-background function

$$F(p, q) = \begin{cases} \text{true} & \text{if } L_B(p) = L_B(q) \\ \text{false} & \text{otherwise} \end{cases} \quad (51)$$

$F(p, q)$ is true in a binary image when the two pixels p and q are either both black or both white. The algorithm to find all connected components of pixels satisfying F is well-known [7,31,44]. We use the familiar graph theoretic algorithm [19]. First we break the graph into subgraphs by removing those edges between pixels where $F(p, q)$ is false. Then, by the connectedness and components algorithm for graphs, our program extracts the connected regions. Figure 5.2 shows the result.

5.3 Motion tracking

We have implemented a motion-tracking application on Cortex-I in ROM that integrates all the capabilities of the system. It is based on a simple frame-differencing algorithm and is able to track moving objects at approximately 5 frames per second.

The algorithm, `motion_track`, subtracts successive pairs of frames, computes the centroid of motion, and moves the motor to point the camera at the computed centroid. Because this algorithm needs two frames to compute the centroid of motion, and then it misses a frame while the motor is moving, it can only make one motor movement every three frames. Therefore, although it can compute the frame difference and centroid at 16 Hz, the motor is only moved about 5 times per second.

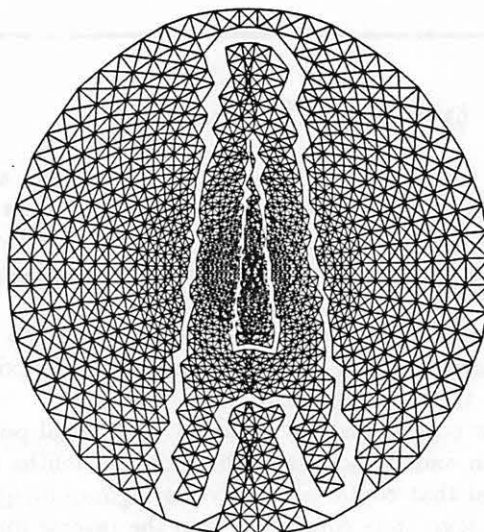


Figure 5.2: The connectivity graph reduces the image connected components problem to its graph theory counterpart. Here we show the foreground-background segmentation of an image containing the letter "A".

Algorithm: **motion_track**

Function: Moves the motor to keep the camera foveated on the centroid of motion in the scene.

Input: None

Output: None

$\Theta \leftarrow 0$ { Initial motor position }

$\Phi \leftarrow 0$

Repeat

Move motor to (Θ, Φ)

wait 50ms

{ Wait a little for motor ringing to decay }

$L_{Null} \leftarrow$ Acquire image

{ Ignore image with motor movement }

$L_1 \leftarrow$ Acquire image

{ Acquire two successive images }

$L_2 \leftarrow$ Acquire image

$\mathcal{P} \leftarrow 0$

{ $\mathcal{P}$ in TV image coordinates }

total_weight $\leftarrow 0$

For each point, $p \in L_1$

{ Compute centroid of motion }

$L_M(p) \leftarrow |L_1(p) - L_2(p)|$

If $L_M(p) > \text{MOTION_THRESH}$ Then

weight $\leftarrow L_M(p) \times a(p)$

$\mathcal{P} \leftarrow \mathcal{P} + \text{weight} \times \mu(p)$

total_weight $\leftarrow \text{total_weight} + \text{weight}$

End If

End For

$(x, y) \leftarrow \mathcal{P} / \text{total_weight}$

$\Theta \leftarrow \Theta + \text{DEG_PER_PIXEL} \times x$

$\Phi \leftarrow \Phi + \text{DEG_PER_PIXEL} \times x$
 Forever

5.4 License plate tracking & reading

We solved a difficult tracking and pattern recognition problem using a combination of **Cortex-I** and a Sun Sparcstation, bringing it to the second stage of development as discussed in Section 5.1. This application is described in detail in [37,38], but is summarized here. The task is to find the license plate on a moving car, track it as the car moves towards the camera, and finally read the characters on the license plate when it gets close enough to the camera.

The experimental setup consists of a real license plate attached to a toy truck that is pulled by a string wound around a motor-driven pulley. The truck moves at approximately two inches/second. It is tracked for about 6 feet before the license plate is read.

Cortex-I is set up to allow commands sent via an RS-232 serial port to control the SPM position. The commands specify the pan and tilt angles of the motor in tenths of a degree. **Cortex-I** outputs a standard RS-170 video signal that contains the inverse logmap image. This signal is then digitized into the Sun, and the forward logmap is computed from the inverse image just as if it were a uniform TV image. This method is inefficient as it wastes communication bandwidth in transmitting an inverse image, and computational resources in computing the forward map from the inverse map. We chose this approach because it allows the existing application running on the host to use the **Cortex-I** hardware without modification.

The license plate is tracked by using a combination of three techniques which are a modified hough transform [43], a foreground-background segmentation, and a corner detector. The algorithm assumes that the car moves at a constant velocity. It estimates the license plate's position each frame using a Kalman filter. When the car is close enough, and thus the characters on the license plate are big enough, the license plate is scanned from left to right with 5 frames. Each frame is analyzed to segment the characters and perform OCR. The results from the 5 frames are then merged together to yield the string of characters on the license plate. A sequence of images showing the tracking of the license plate is shown in Figure 5.3. The OCR results are shown in Figure 5.4, and the actual and estimated motor position during the tracking phase is shown in Figure 5.5.

This application also runs on the Sun development system using a commercial camera and actuator where it correctly reads the license plate [37]. The present prototype's limitations are described in the next section. We are currently building a new version which addresses these limitations - adding a zoom lens, more processing power, and more memory.

5.5 Cortex-I limitations

The development of the application described in the previous section allows for a better understanding of some of the limitations of using **Cortex-I** in a real situation. We found that although **Cortex-I** works well, it is limited in several ways which are:

- Not enough processing power** The Motorola M68332, which is capable of about 2 MIPS, performs all of the image processing in the system as well as controlling the motor. This is not enough to track the license plate even at these low speeds.
- Not enough memory** The M68332 board has only 192 Kbytes of RAM. This is not enough to hold the OCR tables, and thus the application could not be run on **Cortex-I** directly.
- Not enough sensor resolution** By the time the license plate is near enough to the camera so that the characters are reliably recognizable, the truck is so close to **Cortex-I**, that there is not enough time to scan the license plate and read the characters. Because of this, the characters on the

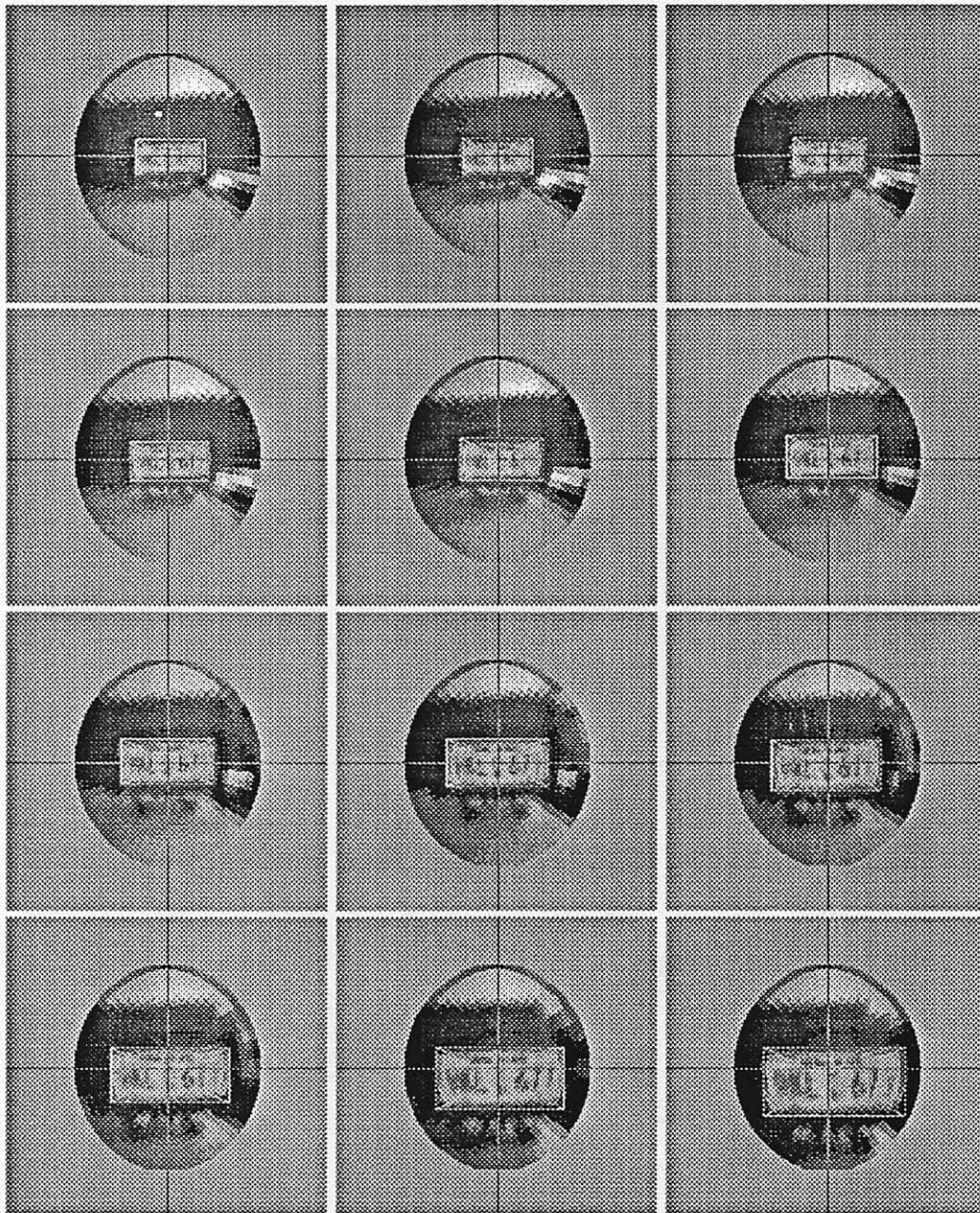


Figure 5.3: Sequence of images showing the results of tracking a moving license plate using a Kalman filter estimation procedure. A rectangle is drawn around the license plate showing that the license plate has been found correctly.

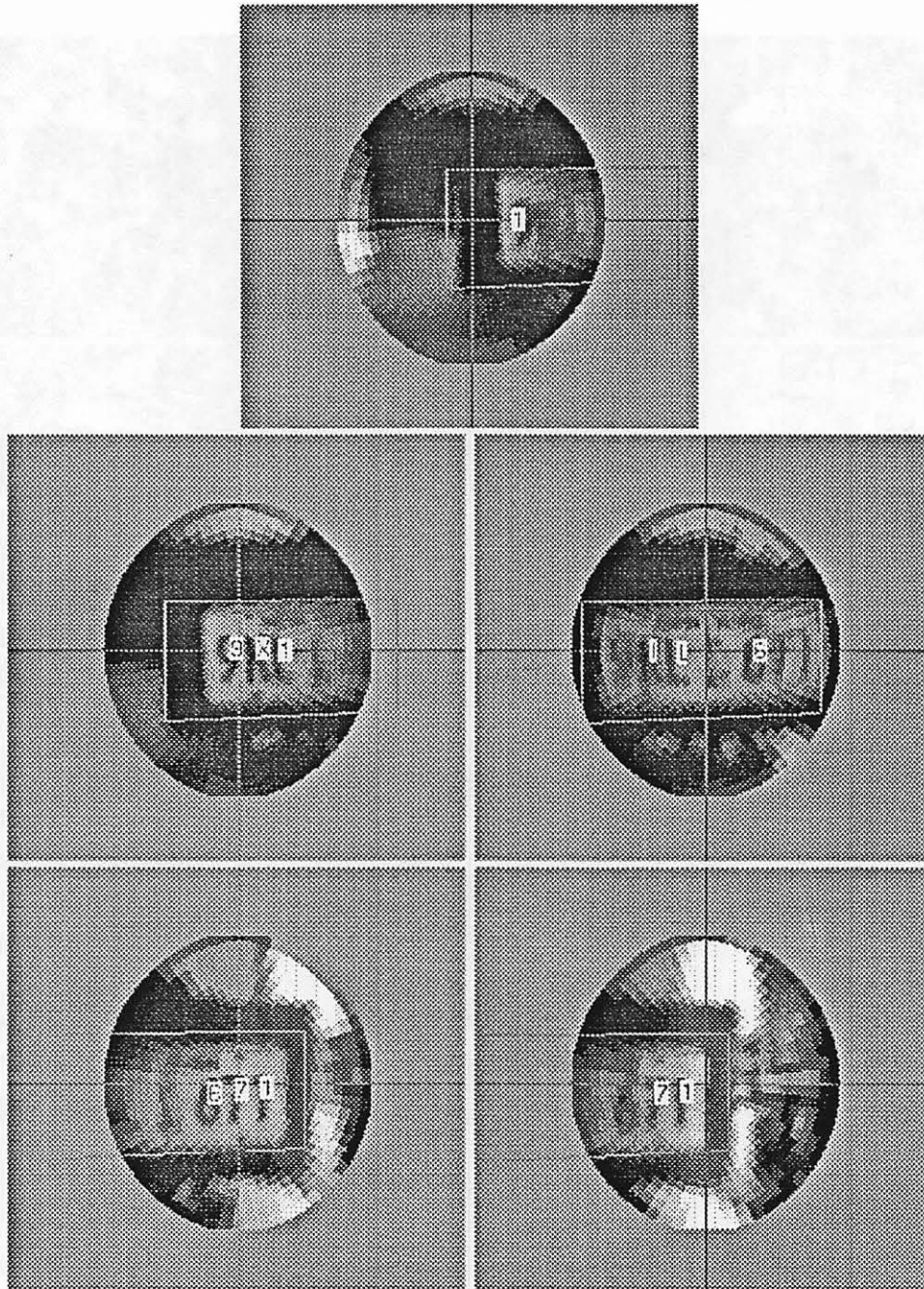


Figure 5.4: The OCR results obtained with several fixation points while the license plate was still moving. The OCR results are superimposed on the images. The actual text on the license plate is "9KL 671". The program output was "9X1L 6711".

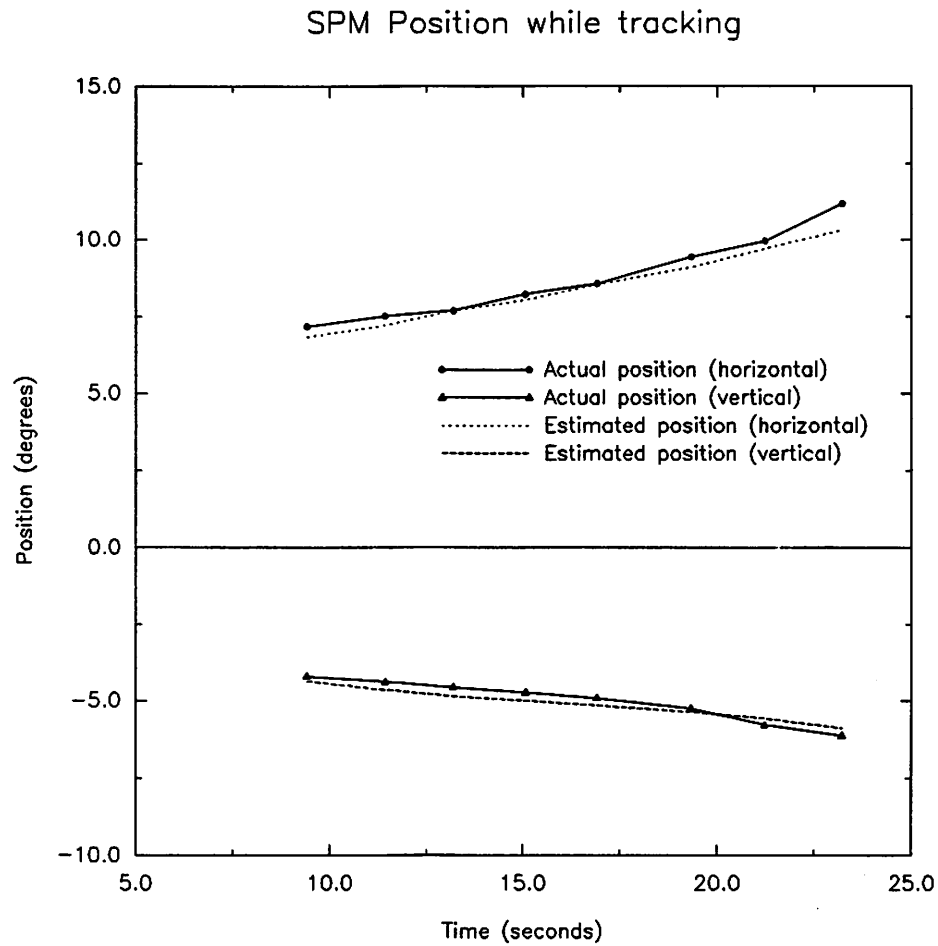


Figure 5.5: Horizontal and vertical actual and estimated position of the motor while tracking a license plate on the moving toy car.

license plate must be read while the license plate is still too far away, resulting in the imperfect recognition results.

Camera doesn't rotate about its focal point The application software assumes the camera rotates about its focal point when it scans the license plate to read the characters. Because the camera doesn't rotate about its focal point, the rotation results in an effective translation which throws off the expected position of the camera. Because of this, the OCR results from the different frames are not merged accurately, yielding an incorrect resultant string.

Appendix A

Glossary

active vision A computer vision system that purposely changes the sensor's parameters in response to the sensor's environment. The parameters that can be changed very often include, but are not limited to, position, focus, zoom, and vergence.

actuator A device to put an object into action or motion.

attention Where to look next. In a space-variant active vision system, only a small area of the scene (i.e., the fovea) is viewed in high resolution, so the problem of where to aim the camera is the problem of attention.

CCD A silicon based technology, the Charged Coupled Device (CCD) is used almost exclusively for image sensors.

CMOS A silicon based technology, the Complementary Metal Oxide Semiconductor (CMOS) circuit allows many types of electronic circuits to be built, including image sensors.

connectivity graph The graph representation of a space-variant image that allows general-purpose image processing on an image of any geometry. The graph has one vertex for each pixel in the image, and one edge between every pair of pixels that are adjacent to each other in the inverse map.

dual two-step control strategy A control strategy for moving the SPM. This applies the two-step method twice.

field width/field of view A description of how much of an external scene is viewed by a camera, typically measured in degrees.

fixed percentage control strategy A control strategy for moving the SPM. This moves the motor in small decreasing steps to the final position.

forward map image The computer representation of a space-variant image. Each pixel is stored in one word of memory and represents differing amounts of the external scene resulting in a warped image.

fovea The central, high-resolution portion of a logmap image.

home position The initial resting position of the Spherical Pointing Motor from which it can make pan or tilt excursions of limited extent.

inverse map image A space-variant image mapped to scene coordinates so that the image is no longer warped, and represents the external scene. Its main purpose is for a human to view a space-variant image.

logmap image An example of a space-variant image defined by a complex logarithm. It is typically composed of concentric rings of pixels, so that the center of the image has very small pixels and high resolution while the periphery has large pixels and thus, low resolution.

- multi two-step control strategy** A control strategy for moving the SPM. This applies the two-step method multiple times.
- multi-resolution** Any of several techniques of image representation and processing in which a scene is represented by an image or images of varying resolution. Both pyramids and space-variant images are examples of multi-resolution techniques.
- pan** Movement of a object (such as a camera) horizontally.
- pan-tilt actuator** A computer-controlled actuator designed to point an object such as a camera.
- Platform Pantilt** A small two degree-of-freedom pan-tilt actuator that is used to point a small camera. It consists of a platform connected to two linear stepper motors by several universal joints. It is run open-loop, and attains rotational velocities of up to about 100 degrees per second.
- progressive video coding** A method of image compression in which a image is represented by a sequence of logmap images foveated at different points in the scene. A high-resolution version of the scene is built up over time from the multiple images.
- pyramid** A multi-resolution method of image representation where an image is stored at several different resolutions. Typically the lowest level contains the full image, the next level contains the image at half-resolution, the next level contains the image a quarter-resolution, and so on until the top level contains a single pixel.
- random map** An example of a space-variant image defined by a stochastic process that places pixels randomly in the map with more pixels in the center than the periphery.
- resolution** A measure of the information content of an image. Higher resolution means more information.
- spoke and ring lookup tables** The look-up-tables used to map a TV image to a space-variant image. The tables contain the address that each TV pixel maps to. Because it is a many-to-one mapping, all TV pixels that are mapped to the same forward map pixel must be averaged together.
- ringing** The oscillatory motion that the Spherical Pointing Motor makes when moved to a new position without any control.
- runs** Typically used in the description of "run-length-encoding", a run is a sequence of pixels in a row of an image that have the same value. We use it to specify a sequence of pixels in a row of a TV image that map to the same logmap pixel.
- saccade** Very fast point to point motions of the human eye that are used to attend to a new area, not for tracking a moving object. "Saccade" is often applied to a computerized pan-tilt actuator to describe any point to point motion.
- solenoid** A coil of wire that when carrying current, acts like a bar magnet. Typically used with a permanent magnet to make a one degree-of-freedom linear actuator.
- space-variant image** An image whose resolution is spatially non-uniform, an example of which is a logmap image.
- spatial quality measure, Q** A figure of merit for an image sensor describing the ratio of field-width to resolution.
- Spherical Pointing Motor** A miniature two degree-of-freedom pan-tilt actuator that is used to point a miniature camera. It is a single direct drive motor that is run open-loop and attains rotational velocities of several hundred degrees per second.
- temporal continuity** The concept that in real scenes, things change slowly. Thus, in a rapid sequence of images of a real scene, there is not much change between successive frames, and the position of an object in one frame is a good prediction of the position of that object in the next frame.
- tilt** Movement of an object (such as a camera) vertically.

- truncated pyramid** A multi-resolution method of image representation based on a pyramid where only a fixed-size window of each pyramid is stored. The result is effectively a digital approximation of a logmap image. The bottom layer contains a high-resolution representation of a small field in the image. The next layer contains half the resolution of twice as much area, and so on. The truncated pyramid is essentially a window into a regular pyramid where the window represents the foveation point.
- TV/uniform image** A "standard" image consisting of a rectangular array of pixels that are all the same size and represent equal areas of scene space. Sometimes referred to as a space-invariant image.
- two-step control strategy** A control strategy for moving the SPM. This moves the motor halfway to the desired position and when the motor overshoots and reaches its zero-velocity point, it is held at this final position.
- vertical smearing** The resulting image degradation of certain CCD image sensors produced because the image is read from the sensor by shifting it vertically through the sensor. Thus, as the image is read out from the sensor, it accumulates light intensity information from every other pixel in the same column.

Appendix B

Variable List

$a(u, v)$	Area of a logmap pixel in TV pixels
α	Constant used to offset logmap calculation to replace the singularity at the center with a discontinuity along the vertical meridian
A	Enclosed area of a coil of the SPM
$\mathcal{A}$	Matrix of coefficients used to desmear a logmap image
b	Vectorized version of $L'(i, j)$ used for desmearing
β	Constant used in the equation for the ringing of the SPM
B	Permanent magnetic field in the SPM
c	Damping constant used in the equation for the ringing of the SPM
C	The magnitude of the exponential envelope describing the decay of the ringing of the SPM
calib	Calibration table for the SPM to map coil currents to motor position
d	Distance between the center of the universal joint connected to the end of the second drive shaft in the Platform Pantilt
D	Diameter of the wire in a coil of the SPM
e	Distance that the laser beam moved when measuring SPM accuracy
E	All of the edges in the connectivity graph
E_I	Edges in the connectivity graph from the inverse map
E_F	Edges in the connectivity graph from the forward map
f	Distance of SPM from wall when measuring accuracy
$G(V, E)$	Connectivity graph of a logmap image
h_1	Height of the first shaft in the Platform Pantilt
h_2	Height of the second shaft in the Platform Pantilt
$I(i, j)$	TV image
$I'(i, j)$	TV image as it read out from a CCD image sensor that has vertical smearing
i	Current in a coil of the SPM
i, j	General purpose indices, most often used for TV images
k, l	General purpose indices
κ	Constant used in the equation for torque of the SPM
K	Constant equal to NA used in describing the SPM
$L(u, v)$	Logmap image
$L'(i, j)$	Logmap image as it read out from a CCD image sensor that has vertical smearing

m	Object distance in a lens system
m'	Image distance in a lens system
N	Number of turns of wire in a coil of the SPM
$\mathcal{N}(p)$	List of neighbors of a logmap pixel, p
Θ	Tilt (vertical) angle of the SPM
p	Individual vertex in the connectivity graph, i.e., point in a logmap image
$\bar{p}$	Mean pixel value in a logmap image
P	Power dissipated by a coil of the SPM
P_1, P_2	Points in retinal and log space, respectively, used to illustrate the logmap calculation
$\mathcal{P}$	Point in TV image space
$\phi(u, v)$	Vertex identified by its pixel coordinates
Φ	Pan (horizontal) angle of the SPM
r	Rotational inertia of the SPM
$\mathcal{R}$	Resistance of a coil of the SPM
$R(i, j)$	Ring look-up-table for computing $L(u, v)$ from $I(i, j)$
<code>run_addr</code>	Run address table used by <code>run_length_encode</code> to map a TV image to a logmap image.
<code>run_len</code>	Run-length table used by <code>run_length_encode</code> to map a TV image to a logmap image.
s	Distance between the center of the universal joint connected to the end of the first drive shaft in the Platform Pantilt
$S(i, j)$	Spoke look-up-table for computing $L(u, v)$ from $I(i, j)$
Ψ	Difference between the initial position and the destination position of the SPM
τ	Torque on the SPM due to a coil of wire
<code>threshold</code>	Threshold used for binarizing an image used in calibrating the SPM.
u, v	General purpose indices, most often used for logmap images
$u(p), v(p)$	Array coordinates in logmap image L of a point p
$\bar{\mu}$	Magnetic dipole moment induced by a coil of wire in the SPM
$\mu(p)$	Centroid of a logmap pixel, p , in TV image coordinates
V	Vertices in the connectivity graph
$\mathcal{V}$	Voltage across a coil of the SPM
ω	Frequency of the ringing of the SPM
$W(u, v, k, l)$	Weights used in computing logmap desmearing
$W_{TOT}(u, v)$	Total weights used in computing logmap desmearing
x, y	Coordinates of a point in log space for computing a logmap image
$\mathcal{X}$	Desmeared logmap image
z	Point in retinal space used to illustrate logmap calculation

Appendix C

Constant List

DEG_OF_WORKSPACE	Size of workspace of SPM in degrees
DEG_PER_PIXEL	Number of degrees that each pixel subtends for our image sensor and lens combination
DIM_CALIB	Number of calibration points per axis for the SPM
INT_SCAN	Integration to scan-time ratio for reading out the CCD image sensor
MOTION_THRESH	Threshold used by <code>motion_track</code> algorithm to determine when a motion pixel should contribute to the motion centroid.
NCOLS	Number of columns in a TV image, $I(i, j)$, i.e., $j \in \{0, \dots, \text{NCOLS} - 1\}$
NPIXELS	Total number of pixels in a logmap image, $L(u, v)$ that map to TV pixels, i.e., the count of pixels, (u, v) such that $a(u, v) > 0$
NRINGS	Number of columns in a logmap image, $L(u, v)$, i.e., $v \in \{0, \dots, \text{NRINGS} - 1\}$
NROWS	Number of rows in a TV image, $I(i, j)$, i.e., $i \in \{0, \dots, \text{NROWS} - 1\}$
NRUNS	Number of runs in the mapping from TV to logmap coordinates.
NSPOKES	Number of rows in a logmap image, $L(u, v)$, i.e., $u \in \{0, \dots, \text{NSPOKES} - 1\}$

Bibliography

- [1] A. Lynn Abbott. Selective fixation control for machine vision: A survey. In *Proceedings of the IEEE International conference on systems, man, and cybernetics*, pages 1–6, October 1991.
- [2] A. Lynn Abbott and Narendra Ahuja. Surface reconstruction by dynamic integration of focus, camera vergence, and stereo. *International Conference on Computer Vision*, pages 532–543, 1988.
- [3] A. Lynn Abbott and Narendra Ahuja. The university of illinois active vision system. Technical Report CV-91-8-2, University of Illinois, Beckman Institute, 1991.
- [4] Peter K. Allen, Billibon Yoshimi, and Aleksandar Timcenko. Real-time visual servoing. In *Proceedings of the 1991 IEEE International Conf. on Robotics and Automation*, pages 851–856, April 1991.
- [5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Intl. J. Computer Vision*, 2:333–356, 1988.
- [6] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996–1005, 1988.
- [7] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [8] Aijza A. Baloch, Allen M. Waxman, Aijza A. Baloch, and Allen M. Waxman. Visual learning: adaptive expectations, and behavioural conditioning of the mobile robot mavin. *Neural Networks*, 4:271–302, 1991.
- [9] Leonard Bergstein. General theory of optically compensated varifocal system. *Journal of the Optical Society of America*, 48(3):154–171, 1958.
- [10] Leonard Bergstein and Lloyd Motz. Two-component optically compensated varifocal system. *Journal of the Optical Society of America*, 52(4):353–362, 1962.
- [11] P. J. Burt. Algorithms and architectures for smart sensing. *Proc. DARPA Image Understanding Workshop*, pages 139–153, 1988.
- [12] Peter. J. Burt. Attention mechanisms for vision in a dynamic world. *9th IEEE Intl. Conf. on Pattern Recognition*, pages 977–987, 1988.
- [13] ed. C. Brown. The rochester robot. Technical Report 257, University of Rochester, 1988.
- [14] A. Califano, R. Kjeldsen, and R. M. Bolle. Data and model driven foveation. *10th International Conference on Pattern Recognition, Vol. 1*, pages 1–7, 1990.
- [15] G. M. Chaikin and C. F. R. Weiman. Image processing system. U.S. Patent No. 4,267,573, May 1981.
- [16] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. *Second Intl. Conf. Computer Vision*, page 514, 1988.
- [17] James L. Crowley. *A Representation for Visual Information*. PhD thesis, Carnegie-Mellon University, November 1981.
- [18] James L. Crowley and Amy L. Lowrie. Multiple resolution representation and matching of stereo scan lines, January 1985.

- [19] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, 1974.
- [20] E. D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241-261, 1988.
- [21] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223-240, 1988.
- [22] E. F. Fichter. A stewart platform-based manipulator: General theory and practical construction. *The International Journal of Robotics Research*, 5(2):157-182, 1986.
- [23] David Halliday and Robert Resnick. *Fundamentals of Physics, Second Edition, Extended Version*. John Wiley & Sons, New York, 1981.
- [24] Vincent Hayward and Ronald Kurtz. Preliminary study of serial-parallel redundant manipulator. In *NASA Conference on Telerobotics*, 1989.
- [25] Paul Horowitz and Winfield Hill. *The Art of Electronics, 2nd Edition*. Cambridge University Press, 1989.
- [26] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol. (Lond.)*, 195:215-243, 1968.
- [27] D. H. Hubel and T. N. Wiesel. Sequence regularity and geometry of orientation columns in the monkey striate cortex. *J. Comp. Neurol.*, 158:267-293, 1974.
- [28] Francis A. Jenkins and Harvey E. White. *Fundamentals of Optics*. McGraw-Hill, New York, 1957.
- [29] Hideharu Kawarabayashi, Makoto Watanabe, Yoshiaki Shirai, Minoru Asade, and Jun Miura. Tracking of a moving object using an active vision system. *Japan Mechanical Society, Robotics Mechatronics Conference Proceedings*, pages 207-212, June 1991.
- [30] T. Kenjo and S. Nagemori. *Permanent-Magnet and Brushless DC Motors*. Oxford Science Publications, 1985.
- [31] Berthold Klaus and Paul Horn. *Robot Vision*. The MIT Press, Massachusetts, 1986.
- [32] G. Kreider, J. Van der Spiegel, I. Born, C. Claeys, I. Debusschere, G. Sandini, P. Dario, and F. Fantini. A retina like space variant ccd sensor. *SPIE/Charge-Coupled Devices and Solid State Optical Sensors*, 1242, 1990.
- [33] E. Krotkov, F. Fuma, and J. Summers. An agile stereo camera system for flexible image acquisition. *IEEE J. Robotics and Automation*, 4:108-113, 1988.
- [34] Larry Matthies, Richard Szeliski, and Takeo Kanade. Incremental estimation of dense depth maps from image sequences. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pages 366-374, 1986.
- [35] C. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.
- [36] Thomas J. Olson and Robert D. Potter. Real-time vergence control. Technical Report 264, University of Rochester, 1988.
- [37] Ping-Wen Ong. *Image Processing, Pattern Recognition and Attentional Algorithms in a Space-Variant Active Vision System*. PhD thesis, GSAS, New York University, 1992, expected.
- [38] Ping-Wen Ong, Richard S. Wallace, and Eric L. Schwartz. Space-variant optical character recognition. In *11th International Conference on Pattern Recognition*, 1992.
- [39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, NY, 1988.
- [40] Daniel Raviv and Avraham Shapira. Miniature vision-based flight simulator. In *4th Annual Conference on Recent Advances in Robotics*, pages 81-86, 1991.

- [41] David A. Robinson. The oculomotor control system: A review. *Proceedings of the IEEE*, 56(6):1032-1049, 1968.
- [42] A. S. Rojer and E. L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. *10th ICPR, Vol. 2*, pages 278-285, 1990.
- [43] Alan S. Rojer. *Space-variant computer vision with a complex-logarithmic sensor geometry*. PhD thesis, GSAS, New York University, 1989.
- [44] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing, 2nd Edition*. Academic Press, 1982.
- [45] G. Sandini, F. Bosero, F. Bottino, and A. Ceccherini. The use of an anthropomorphic visual sensor for motion estimation and object tracking. In *Image Understanding and Machine Vision Workshop*, pages 1-5, 1990.
- [46] Giulio Sandini, Paolo Dario, and I. Debusschere. Active vision based on space-variant sensing. In *5th International Symposium on Robotics Research*, August 1989.
- [47] E. L. Schwartz. The development of specific visual projections in the monkey and the goldfish: Outline of a geometric theory of receptotopic structure. *J. Theoretical Biology*, 69:655-685, 1977.
- [48] E. L. Schwartz. Spatial mapping in primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25:181-194, 1977.
- [49] E. L. Schwartz, B. Merker, E. Wolfson, and A. Shaw. Computational neuroscience: Applications of computer graphics and image processing to two and three dimensional modeling of the functional architecture of visual cortex. *IEEE Computer Graphics and Applications*, 8(4):13-28 (July), 1988.
- [50] Neil C. Singer and Warren P. Seering. An extension of command shaping methods for controlling residual vibration using frequency sampling. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 800-805, Nice, France, May 1992.
- [51] Warren J. Smith. *Modern Optical Engineering*. McGraw-Hill, New York, 1990.
- [52] N. Stelman. Design and control of a six-degree-of-freedom platform with variable admittance. Master's thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1988.
- [53] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 1965.
- [54] John K. Tsotsos. On the relative complexity of active vs. passive visual search. *International journal of computer vision*, 7(2):127-141, 1992.
- [55] J. van der Spiegel, F. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Belluti, and G. Soncini. A foveated retina-like sensor using ccd technology. In C. Mead and M. Ismail, editors, *Analog VLSI Implementations of Neural Networks*. Kluwer, Boston, 1989.
- [56] Richard S. Wallace, Benjamin B. Bederson, and Eric L. Schwartz. Telecortex-i logmap video telephone. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (submitted)*, November 1992.
- [57] Richard S. Wallace, Ping-Wen Ong, Benjamin B. Bederson, and Eric L. Schwartz. Space-variant image processing. Technical Report 256, New York University, Computer Science, Robotics Research, October 1991.
- [58] A. L. Yarbus. *Eye Movements and Vision*. Plenum, NY, 1967.
- [59] Y. Yeshurun and E. L. Schwartz. Cepstral filtering on a columnar image architecture: a fast algorithm for binocular stereo segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):759-767, 1989.



Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics

BENJAMIN B. BEDERSON,* JAMES D. HOLLAN,* KEN PERLIN,† JONATHAN MEYER,† DAVID BACON† AND GEORGE FURNAS‡

**Computer Science Department, University of New Mexico, Albuquerque, NM 87131, U.S.A.,*

†Media Research Laboratory, Computer Science Department, New York University, NY 10003, U.S.A., ‡Bell Communications Research, 445 South Street, Morristown, NJ 07960, U.S.A.

Received 7 March 1995 and accepted 29 September 1995

We describe Pad++, a zoomable graphical sketchpad that we are exploring as an alternative to traditional window and icon-based interfaces. We discuss the motivation for Pad++, describe the implementation and present prototype applications. In addition, we introduce an informational physics strategy for interface design and briefly contrast it with current design strategies. We envision a rich world of dynamic persistent informational entities that operate according to multiple physics specifically designed to provide cognitively facile access and serve as the basis for the design of new computationally-based work materials.

©1996 Academic Press Limited

1. Introduction

Imagine a computer screen made of a sheet of a miraculous new material that is stretchable like rubber but continues to display a crisp computer image, no matter what the sheet's size. Imagine that this sheet is very elastic and can stretch orders of magnitude more than rubber. Further, imagine that vast quantities of information are represented on the sheet, organized at different places and sizes. Everything you do on the computer is on this sheet. To access a piece of information you just stretch to the right part and there it is.

Imagine further that special lenses come with this sheet that let you look onto one part of the sheet while you have stretched another part. With these lenses, you can see and interact with many different pieces of data at the same time that would ordinarily be quite far apart. In addition, these lenses can filter the data in any way you would like, showing different representations of the same underlying data. The lenses can even filter out some of the data so that only relevant portions of the data appear.

Imagine also new stretching mechanisms that provide alternatives to scaling objects purely geometrically. For example, instead of representing a page of text so small that it is unreadable, it might make more sense to present an abstraction of the text, perhaps so that just a title that is readable. Similarly, when stretching out a spreadsheet,

** (bederson, hollan)@cs.unm.edu, † (perlin, meyer bacon)@play.cs.nyu.edu, ‡ gwf@bellcore.com*
URL: <http://www.cs.unm.edu/pad++>

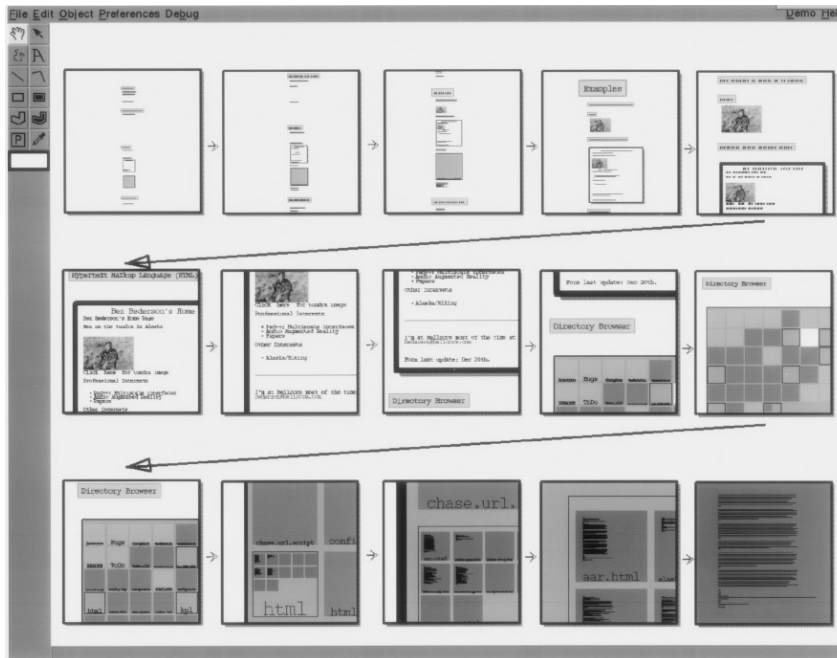


Figure 1. A sequence of views as we zoom into some data

instead of showing huge numbers it might make more sense to show the computations from which the numbers were derived or a history of interaction with them.

The beginnings of an interface like this sheet exists today in a program we call Pad++. We don't really stretch a huge rubber-like sheet, but we simulate it by *zooming* into the data. We use what we call *portals* to simulate lenses, and a notion we call *semantic zooming* to scale data in non-geometric ways. The user controls where they look on this vast data surface by panning and zooming. Portals are objects on the Pad++ data surface that can see anywhere on the surface, as well as filter data to represent it differently than it normally appears.

Panning and zooming allow navigation through a large information space via direct manipulation. By tapping into people's natural spatial abilities, we hope to increase users' intuitive access to information. Conventional computer search techniques are also provided in Pad++, bridging traditional and new interface metaphors. Figure 1 depicts a sequence of views as we pan and zoom into some data.

1.1. Motivation

If interface designers are to move beyond windows, icons, menus and pointers to explore a larger space of interface possibilities, additional ways of thinking about interfaces that go beyond the desktop metaphor are required.

There are myriad benefits associated with metaphor-based approaches, but they also orient designers to employ computation primarily to mimic mechanisms of older media. While there are important cognitive, cultural and engineering reasons to exploit earlier successful representations, this approach has the potential of under-utilizing the mechanisms of new media.

For the last few years we have been exploring a different strategy [21] for interface design to help focus on novel mechanisms enabled by computation rather than on mimicking mechanisms of older media. Informally, the strategy consists of viewing interface design as the development of a physics of appearance and behavior for collections of informational objects.

For example, an effective informational physics might arrange for an object's representation to be a natural by-product of normal activity. This is similar to the physics of certain materials that evidence the wear associated with use. Such wear records a history of use and at times this can influence future use in positive ways. Used books crack open at frequently referenced places. It is common for recently consulted papers to be at the tops of piles on our desks. Usage dog-ears the corners and stains the surface of index cards and catalogs. All these wear marks provide representational cues as a natural product of doing, but the physics of materials limit what can be recorded and the ways it can influence future use.

Following an informational physics strategy has led us to explore history-enriched digital objects [18, 19]. Recording on objects (e.g. reports, forms, source-code, manual pages, email, spreadsheets) the interaction events that comprise their use makes it possible on future occasions, when the objects are used again, to display graphical abstractions of the accrued histories as parts of the objects themselves. For example, we depict the copy history on source code. This allows a developer to see that a particular section of code has been copied and perhaps be led to correct a bug not only in the piece of code being viewed but also in the code from which it was derived.

This informational physics strategy has also lead us to explore new physics for interacting with graphical data. As part of that exploration we have formed a research consortium to design a successor to Pad [25]. This new system, Pad++, serves as a substrate for exploration of novel interfaces for information visualization and browsing in complex, information-intensive domains. The system is being designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants) and interactive set-top cable boxes. Here we describe the motivation behind the Pad++ development, report the status of the current implementation and present initial prototype applications.

Today, there is much more information available than we can access readily and effectively. The situation is further complicated by the fact that we are on the threshold of a vast increase in the availability of information because of new network and computational technologies. Paradoxically, while we continuously process massive amounts of perceptual data as we experience the world, we have perceptual access to very little of the information that resides within our computing systems or that is reachable via network connections. In addition, this information, unlike the world around is, is rarely presented in ways that reflect either its rich structure or dynamic character.

We envision a much richer world of dynamic persistent informational entities that operate according to multiple physics specifically designed to provide cognitively facile access. These physics need to be designed to exploit semantic relationships explicit and implicit in information-intensive tasks and in our interaction with these new kinds of computationally-based work materials.

One physics central to Pad++ supports viewing information at multiple scales and attempts to tap into our natural spatial ways of thinking. We address the information

presentation problem of how to provide effective access to a large structure of information on a much smaller display. Furnas [15] explored degree of interest functions to determine the information visible at various distances from a central focal area. There is much to recommend the general approach of providing a central focus area of detail surrounded by a periphery that places the detail in a larger context.

With Pad++ we have moved beyond the simple binary choice of presenting or eliding particular information. We can also determine the scale of the information and, perhaps most importantly, the details of how it is rendered can be based on various semantic and task considerations that we describe below. This provides semantic task-based filtering of information that is similar to the early work at MCC on lens-based filtering of a knowledge base using HITS [20] and the recent work of moveable filters at Xerox [4] [30].

The ability to make it easier and more intuitive to find specific information in large dataspace is one of the central motivations behind Pad++. The traditional approach is to filter or recommend a subset of the data, hopefully producing a small enough dataset for the user to navigate effectively. Pad++ is complementary to these filtering approaches in that it promises to provide a useful substrate to *structure* information.

2. Description

Pad++ is a general-purpose substrate for creating and interacting with structured information based on a zoomable interface. It adds scale as a first class parameter to all items, as well as various mechanisms for navigating through a multiscale space. It has several efficiency mechanisms which help maintain interactive frame-rates with large and complicated graphical scenes.

While Pad++ is not an application itself, it directly supports creation and manipulation of multiscale graphical objects, and navigation through spaces of these objects. It is implemented as a widget in Tcl/Tk [24] (described in a later section) which provides an interpreted scripting language for creating zoomable applications. The standard objects that pad++ supports are colored text, graphics, images, portals and hypertext markup language (HTML). Standard input widgets (buttons, sliders, etc.) are supplied as extensions.

One focus in the current implementation has been to provide smooth zooming within very large graphical datasets. The nature of the Pad++ interface requires consistent high frame-rate interactions, even as the dataspace becomes large and the scene gets complicated. In many applications, speed is important, but not critical to functionality. In Pad++, however, the interface paradigm is inherently interactive. One important searching strategy is to visually explore the dataspace while zooming through it, so it is essential that interactive frame rates be maintained.

A second focus has been to design Pad++ to make it relatively easy for third parties to build applications using it. To that end, we have made a clear division between what we call the 'substrate' and applications. The substrate, written in C++, is part of every release and has a well-defined API. It has been written with care to ensure efficiency and generality. It is connected to a scripting language (currently Tcl, but we are exploring alternatives) that provides a fairly high-level interface to the complex graphics and interactions available. While the scripting language runs quite slowly, it is used as a glue language for creating interfaces and

putting them together. The actual interaction and rendering is performed by the C++ substrate. This approach allows people to develop applications for Pad++ while avoiding the complexities inherent in this type of system. (See the Implementation section for more information on this.)

2.1. PadDraw: A Sample Application

PadDraw is a sample drawing application built on top of Pad++. It supports interactive drawing and manipulation of objects as well as loading of predefined or programmatically created objects. This application is written entirely in Tcl (the scripting language) and was used to produce all the figures depicted in this paper. The tools, such as navigation aids, hyperlinks and the outline browser, that we discuss later, are part of this application.

The basic user interface for navigating in PadDraw uses a three button mouse. The left button is mode dependent and lets users select and move objects, draw graphical objects, follow hyperlinks, etc. The middle button zooms in and the right button zooms out. Zooming is always centered on the cursor, so moving the mouse while zooming lets the user dynamically control which point they are zooming around.

PadDraw has a primitive Graphical User Interface (GUI) builder that is in progress. Among other things, it allows the creation of active objects. Active objects can animate the view to other locations (a kind of hyperlink) or move other objects around on the surface.

2.1.1. Navigation

Easily finding information on the Pad++ surface is obviously very important since intuitive navigation through large dataspace is one of its primary motivations. Pad++ supports visual searching with direct manipulation panning and zooming in addition to traditional mechanisms, such as content-based search.

Some applications animate the view to a certain piece of data. These animations interpolate in pan and zoom to bring the view to the specified location. If the end point is further than one screen width away from the starting point, the animation zooms out to a point midway between the starting and ending points, far enough out so that both points are visible. The animation then smoothly zooms in to the destination. This gives both a sense of context to the viewer as well as speeding up the animation since most of the panning is performed when zoomed out which covers much more ground than panning while zoomed in. See the section on Space-Scale Diagrams for more detail on the surprisingly complex topic of multiscale navigation.

Content-based search mechanisms support search for text and object names. Entering text in a search menu results in a list of all of the objects that contain that text. Clicking on an element of this list produces an automatic animation to that object. The search also highlights objects on the data surface that match the search specification with special markers (currently a bright yellow outline) that remain visible no matter how far you zoom out. Even though the object may be so small as to be invisible, its marker will still be visible. This is a simple example of task-based semantic zooming. See Figure 2 for a depiction of the content-based search mechanism.

We have also implemented visual bookmarks as another navigational aid. Users can

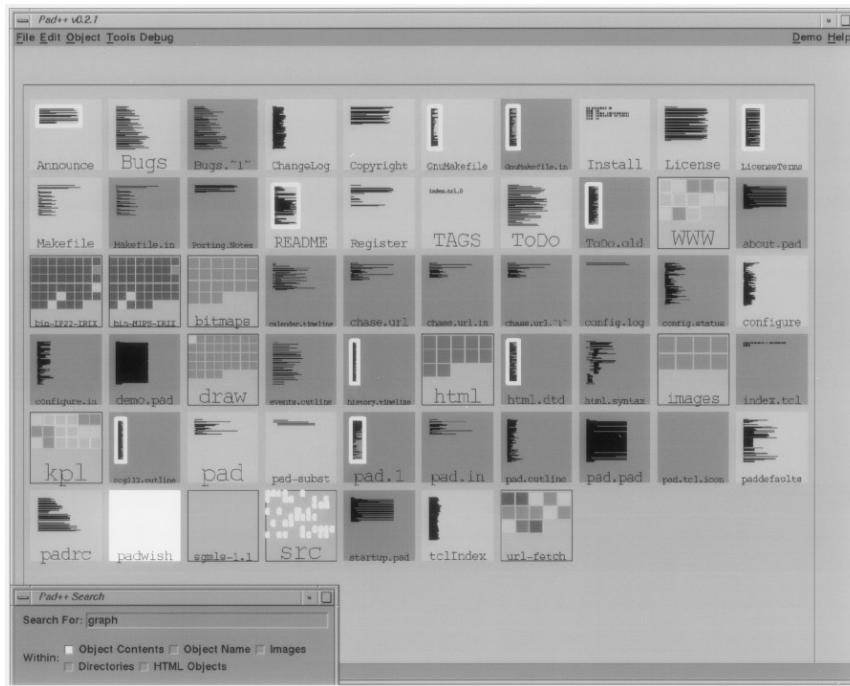


Figure 2. The content-based search window lets users search for text and names, and then animate to any of those objects by clicking on the search entry

remember places they have been, and maintain miniature views onto those places. Moving the mouse over one of these bookmark views places a marker in the main view to identify where it will take you (although the marker may be off to the side and hence not visible). Clicking on a view animates the main view to that place (Figure 3).

2.2. Portals

Portals are special items that provide views onto other areas of the Pad++ surface, or even other surfaces. Each portal passes interaction events that occur within it to the place it is looking. Thus, you can pan and zoom within a portal. In fact, you can perform any kind of interaction through a portal. Portals can filter input events, providing a mechanism for changing behavior of objects when viewed through a portal. Portals can also change the way objects are presented. When used in this fashion, we call them *lenses* (see below).

Portals can be used to replicate information efficiently, and also provide a method to bring physically separate data near each other. Figure 1 was created using several portals, each looking at approximately the same place at different magnifications.

Portals can also be used to create indices. For example, creating a portal that looks onto a hyperlink allows the hyperlink to be followed by clicking on it within the portal, changing the main view. This however, may move the hyperlink off the screen.

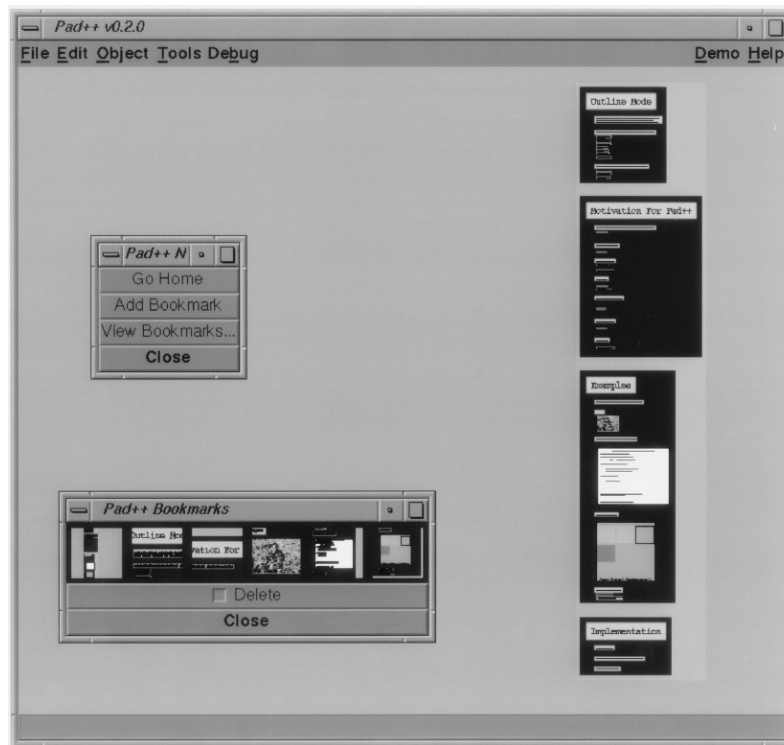


Figure 3. Visual bookmarks let users remember interesting places they have been by showing miniature views of those places. Clicking on one of the views animates the main view to the location

We can solve this by making the portal (or any other object for that matter) *sticky*, which is a method of keeping the portal from moving around as the user pans and zooms. Making an object sticky effectively lifts it off the Pad++ surface and sticks it to the monitor glass. Thus, clicking on a hyperlink through a sticky portal brings you to the link destination, but the portal index is not lost and can continue to be used.

2.3. Lenses

Designing user interfaces is typically done at a low level, focusing on user interface components rather than on the task at hand. If the task is to enter a number, we should be able to place a generic number entry mechanism in the interface. However, typically, once the specific number entry widget, such as a slider or dial, is decided on, it is fixed in the interface.

We can use lenses to design interfaces at the level of specific tasks. For example, we have designed a pair of number entry lenses for Pad++ that can change a generic number entry mechanism into a slider or dial, as the user prefers. By default the generic number entry mechanism allows entering a number by typing. However, dragging the *slider lens* over it changes the representation of the number from text to a slider, and now the mouse can be used to change the number. Another lens shows the data as a dial and lets you modify that with a mouse as well.

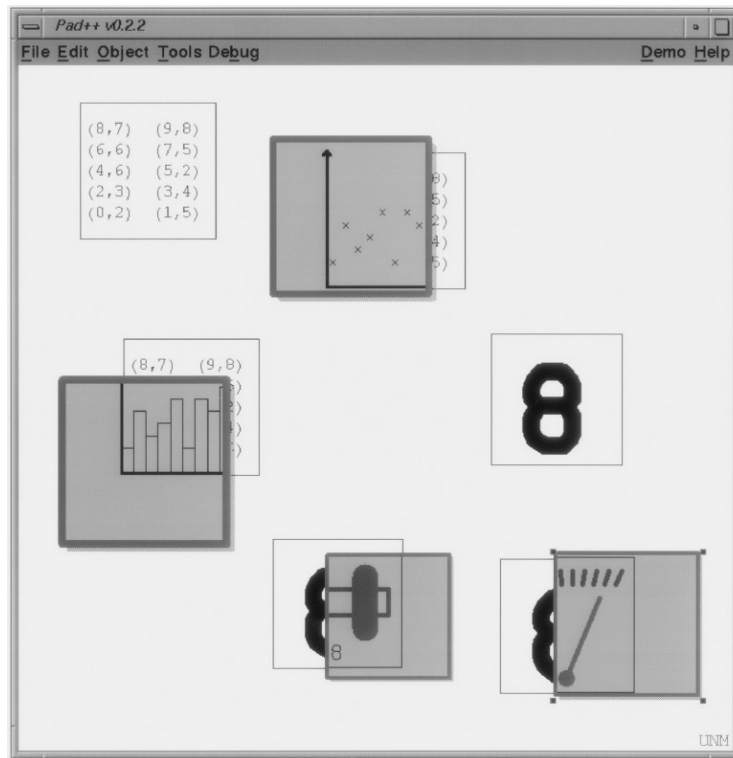


Figure 4. Lenses that show textual data as scatter plots and bar charts

More generally, lenses are objects that alter appearance and behavior of components seen through them. They can be dragged around the Pad++ surface examining existing data. For example, data might normally be depicted by columns of numbers. However, looking at the same data through a lens could show that data as a scatter plot, or a bar chart (see Figure 4).

Lenses such as these support multiple representations so that information can be displayed in ways most effective for the task at hand. They make the notion of multiple representations of the same underlying data more intuitive and can be used to show linkages between the representations. For example, if the slider lens only partially covers the text number entry wide, then modifying the underlying number with either mechanism (text or mouse) modifies both. So typing in the text entry moves the slider, and vice versa.

2.4. Semantic Zooming

Once we make zooming a standard part of the interface, many parts of the interface need to be reevaluated. For example, we can use semantic zooming to change the way things look depending on their size. As we mentioned, zooming provides a natural mechanism for representing abstractions of objects. It is natural to see extra details of an object when zoomed in and viewing it up close. When zoomed out, instead of simply seeing a scaled down version of the object, it is potentially more effective to see a different representation of it.

For example, we implemented a digital clock that at normal size shows the hours and minutes. When zooming in, instead of making the text very large, it shows the seconds, and then eventually the date as well. Similarly, zooming out shows just the hour. An analog clock (implemented as a lens that can be positioned over a digital clock) is similar—it does not show the second hand or the minute markings when zoomed out.

Semantic zooming can take an even more active role in the interface. It can be used as a primary mechanism for retrieving data. We have built prototype tools for accessing system usage including information about the print queue, the system load and the users on the machine. They are depicted as small objects with labels. Zooming into each of them starts a process which gathers the appropriate information and shows it in the now larger object. Zooming out makes the information disappear and the data-gathering process inactive.

3. Visualizations

We are exploring several different types of interactive visualizations within Pad++, some of which are described briefly here. Each takes advantage of the variable resolution available for both representation and interaction.

Layout of graphical objects within a multi-resolution space is an interesting problem, and is quite different than traditional fixed-resolution layout. Deciding how to visually represent an arbitrary graph on a non-zoomable surface is extremely difficult. Often it is impossible to position all objects near logically related objects. In addition, representing the links between objects often requires overlapping or crossing edges. Even laying out a tree is difficult because, generally speaking, there are an exponential number of children that will not fit in a fixed size space.

Traditional layout techniques use sophisticated iterative, adaptive algorithms for laying out general graphs, and still result in graphs that are hard to understand. Large trees are often represented hierarchically with one sub-tree depicted by a single box that references another tree.

Using an interactive zoomable surface, however, allows very different methods of visually representing large data structures. The fact that there is always more room to put information ‘between the cracks’ gives many more options. Pad++ is particularly well suited to visualizing hierarchical data because information that is deeper in the hierarchy can be made smaller. Accessing this information is accomplished by zooming.

3.1. Hypertext Markup Language (HTML)

In traditional window-based systems, there is no graphical depiction of the relationship among windows even when there is a strong semantic relationship. For example, in many hypertext systems, clicking on a hyperlink brings up a new window with the linked text (or alternatively replaces the contents of the existing window). While there is an important relationship between these windows (parent and child), this relationship is not represented.

We are experimenting with multiscale layouts of hypertext document traversals where the parent–child relationships between links is represented visually. The layout

represents a tree that is distorted so that the page that has the focus (i.e. the one being looked at) is quite large. As nodes get further away from the focus, they get smaller. The distortion is controllable with a pop-up window. This is an example of a graphical fisheye view [15]. As links are followed, they are added to the tree and become the current focus. The view is animated so that the new node is centered and large enough to read.

Pad++ reads hypertext written in the Hypertext Markup Language (HTML), the language used to describe hypertext documents used by WWW browsers such as Mosaic and Netscape. Pad++ also can follow links across the internet. Figure 5 shows a snapshot where several hypertext links have been followed. Two views show the same tree focused on different nodes. The Pad++ user interface for accessing hypertext is similar to traditional systems, but zooming mechanisms are employed. There are also special mechanisms to return to an object's parent.

An alternative layout technique (not shown here) uses a camera with a special zoomed in view of the tree. The idea is to give an overview of the tree in one view while allowing individual pages to be read in another view. This gives both a global context and local detail simultaneously. The camera can be dragged around the overview, and the detail view is updated to see what the camera is pointing at. Clicking on a page causes the camera to animate to that page and, when a new page is brought in, the camera centers its view on it.

This layout problem is challenging because the underlying data can be an arbitrary cyclic graph. Any graph can be viewed as a hierarchy by taking a single node and calling it the root node. Imagine taking that node and shaking the graph out. Its neighbors become children, and the children's neighbors become grandchildren, etc. We use this approach to display HTML documents where the order of the links that are followed describe the particular hierarchy imposed on the data. When a cycle is encountered (i.e. a link is followed to a page that is already loaded), the user is brought to the original copy of the page that was loaded, and the focus is put upon it.

3.2. Directory Browser

We built a zoomable directory browser as another exploration of multiscale layout. The directory browser provides a graphical interface for accessing the directory structure of a filesystem (see Figure 6). Each directory is represented by a folder icon and files are represented by solid squares colored by file type. Both directories and files show their filenames as labels when the user is sufficiently close to be able to read them. Each directory has all of its subdirectories and files organized alphabetically inside it. Searching through the directory structure can be done by zooming in and out of the directory tree, or by using the content based search mechanisms described above. Zooming into a file automatically loads its text or image inside the colored square and it can then be annotated. At any particular view, typically three levels of the hierarchy are visible.

3.2.1. Timeline

Scale can be used to convey temporal information. Events which take place over a long period of time use a large scale and brief events are shown at a small scale. We used this notion to visualize some of the history of computing and user interfaces.

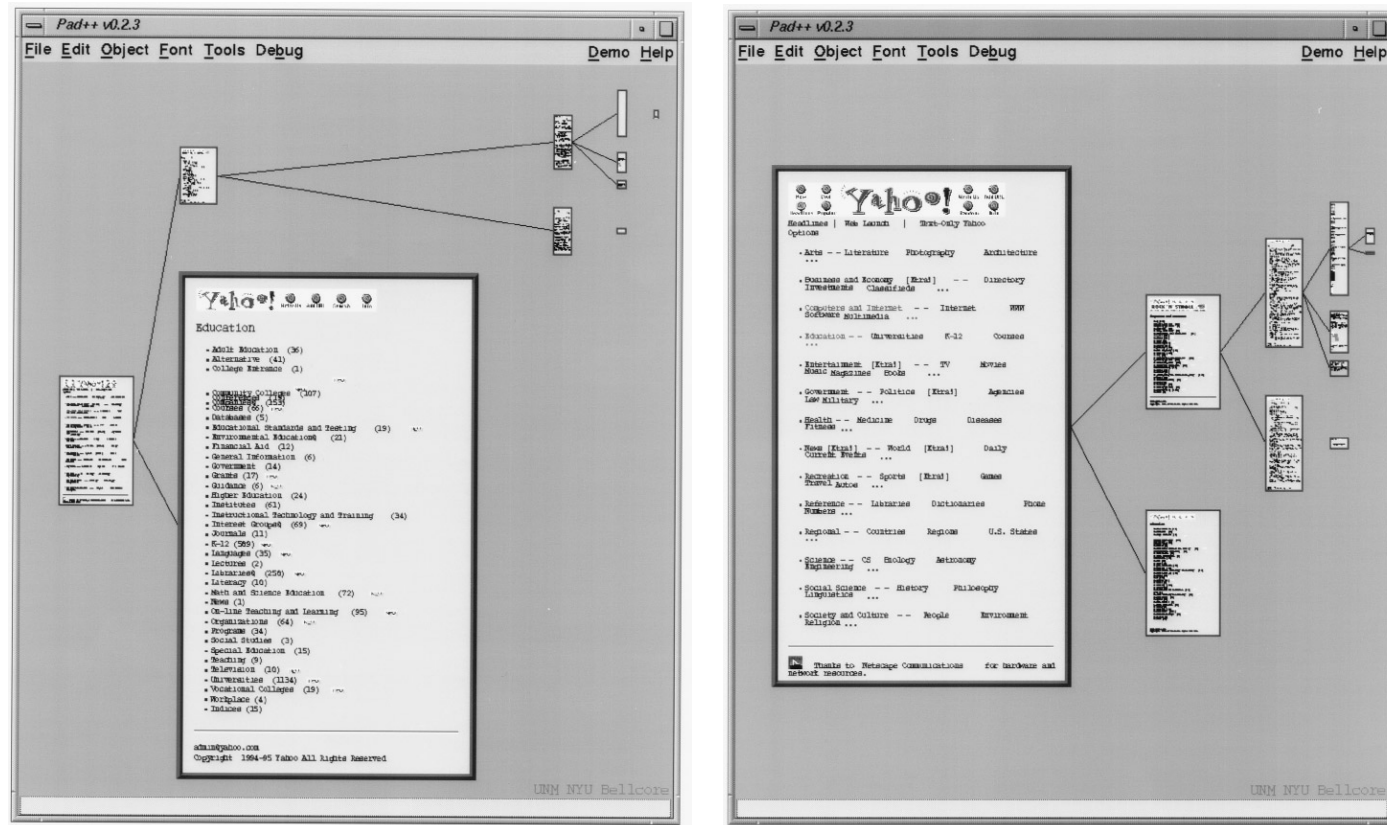


Figure 5. Many different HTML documents loaded in Pad++. Their layout implicitly shows the history of the user's interaction. The two views show the same tree focused on different nodes

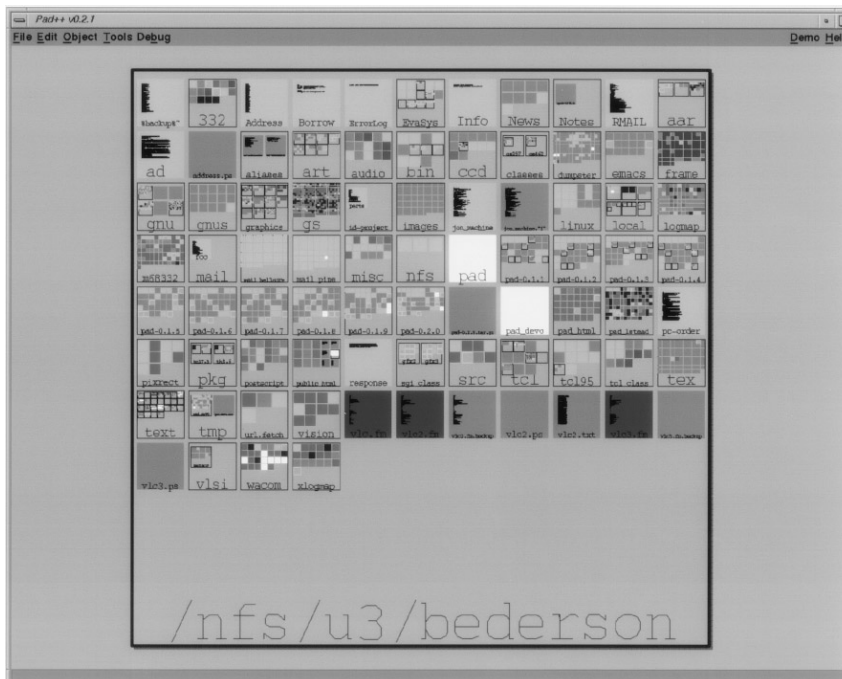


Figure 6. A view of our file system

The timeline visualization shows decades as large numbers. Zooming in on a decade reveals the years within that decade. Further zooming on a particular year shows events which took place during that year. Figure 7 shows a sequence of snapshots as the view is zoomed in.

3.2.2. Oval Document Layout

Since objects on the Pad++ surface reside at absolute locations, the relative positions of objects can be used to encode information. Thus, with the Pad++ HTML browser, position is used to encode the order of a user's traversal of a hypertext document. In the Oval Document Layout, position is used to reinforce the narrative structure of documents (such as guided tours) in which the reader follows a sequence of steps which eventually lead back to the starting point (Figure 8).

In this layout, the first page is placed at the bottom edge of an arc. Subsequent pages are placed around the edge of the arc and are drawn at a scale which reflects their position in the tour—middle pages are shown distant and small, whereas start and end pages appear larger and closer to the user.

Navigation buttons at the bottom edge of each page are used to advance through the document. Clicking on a page when it is distant causes Pad++ to pan and zoom so that the page fills most of the screen.

One advantage of this layout is that as the system animates from one page to the next, the user can infer progress through the document by the direction of the animation: near the start, pages move down and to the left; towards the end, pages move up and to the right.

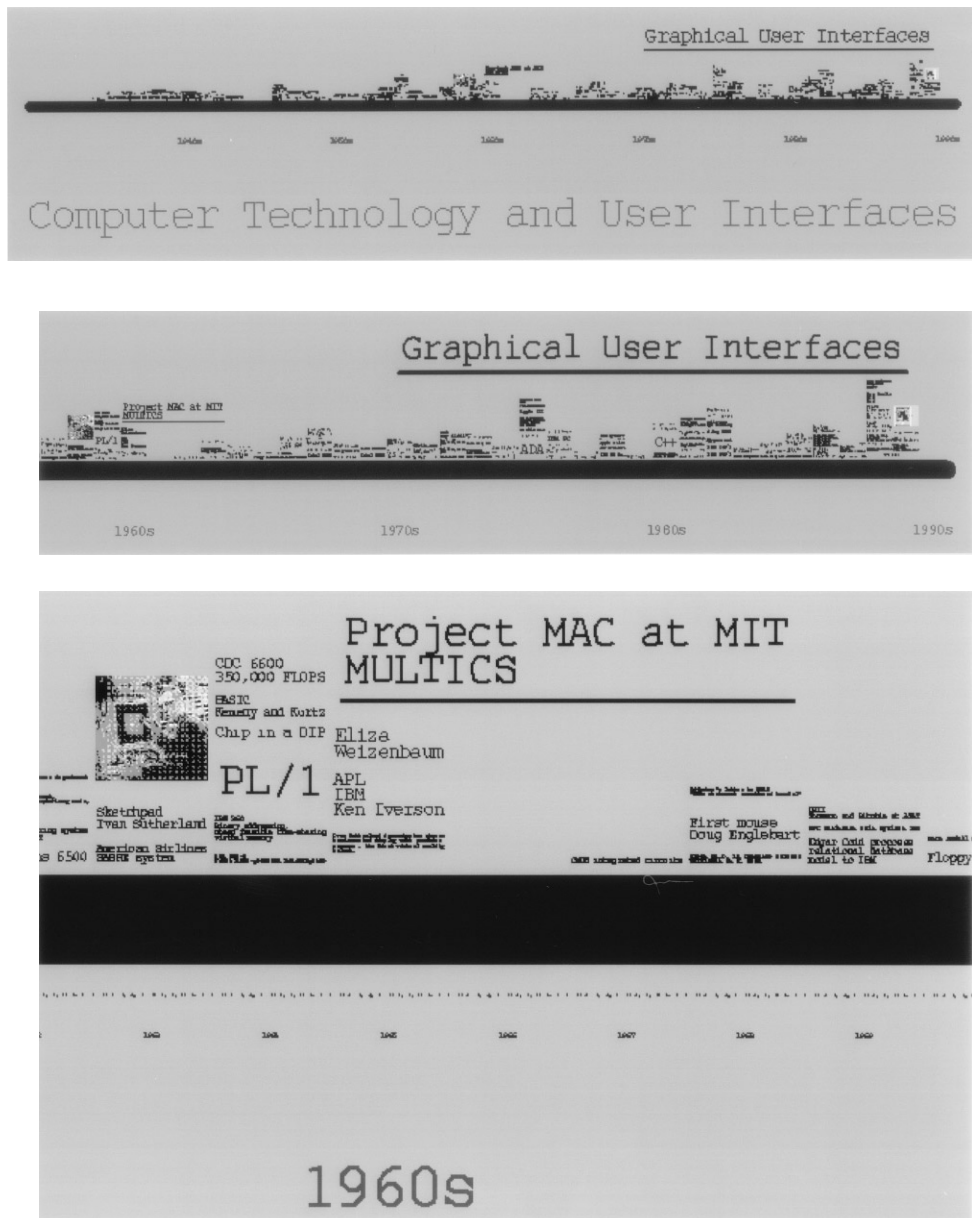


Figure 7. A sequence that views the history of computers and interfaces

The layout is also effective for non-linear access to pages within the document. Zooming out a small distance reveals the whole document, and clicking on a page within the document takes you to that page.

Hotwords and hyperlink buttons in an oval document can be shown with arrows which point towards the destination object. Clicking on the hyperlink animates the Pad++ surface in the direction indicated by the arrow, reducing the

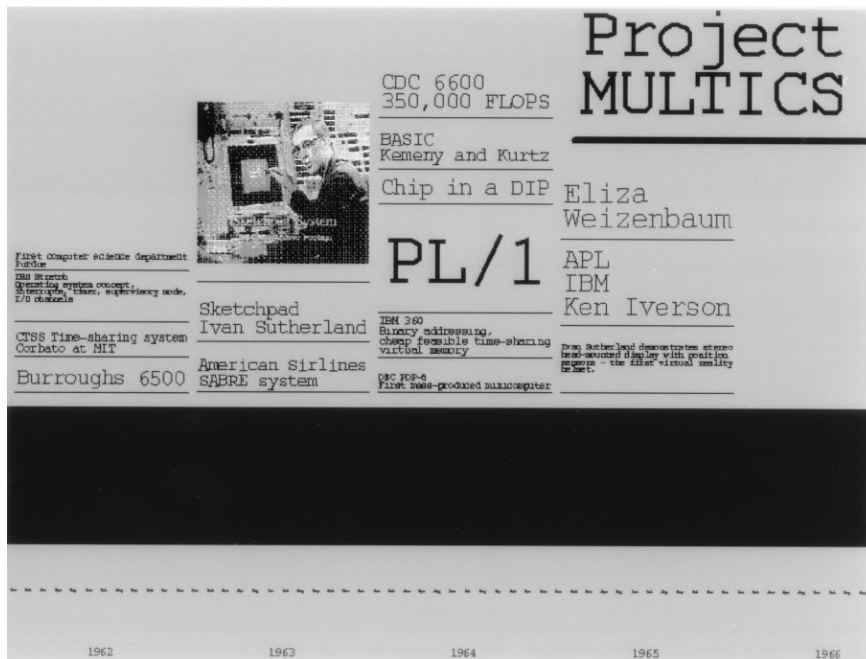


Figure 7. Continued

sense of disorientation that many users experience when navigating hypertext documents.

The Oval document view illustrates that a pan/zoom coordinate system can lead to interesting new ways of laying out even traditional page based material. However, the layout has several drawbacks. It is only practical for relatively short documents and for documents which adopt a circular narrative structure.

4. Space-Scale Diagrams

In an effort to understand multiscale spaces better, we have developed an analytical tool for describing them which we call *space-scale diagrams*. By representing the spatial structure of an information world at all its different magnifications simultaneously, these diagrams allow us to visualize various aspects of zoomable interfaces and analyze their properties. We discuss these diagrams briefly here. They are discussed in more detail in [16].

While Pad++ provides panning and zooming interactions over a two dimensional surface, the basic ideas of a space-scale diagram are most easily illustrated in one dimension. This would typically be a slice through a two-dimensional world.

The basic one-dimensional diagram concept is illustrated in Figure 9. This diagram shows six points that are copied over and over at all possible magnifications. These copies are stacked up systematically to create a two dimensional diagram whose horizontal axes represents the original spatial dimension and whose vertical axis represents the degree of magnification (or scale). Because the diagram shows an infinite number of magnifications, each point is represented by a line emanating from

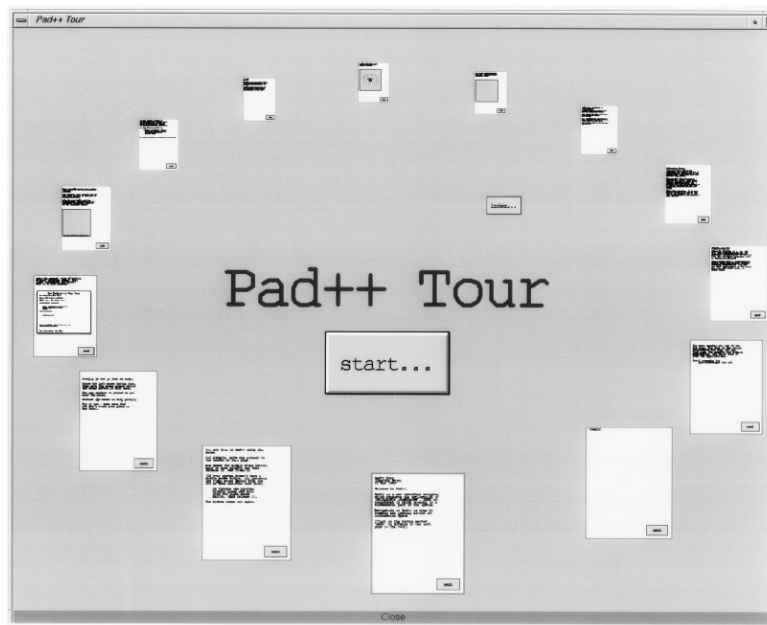
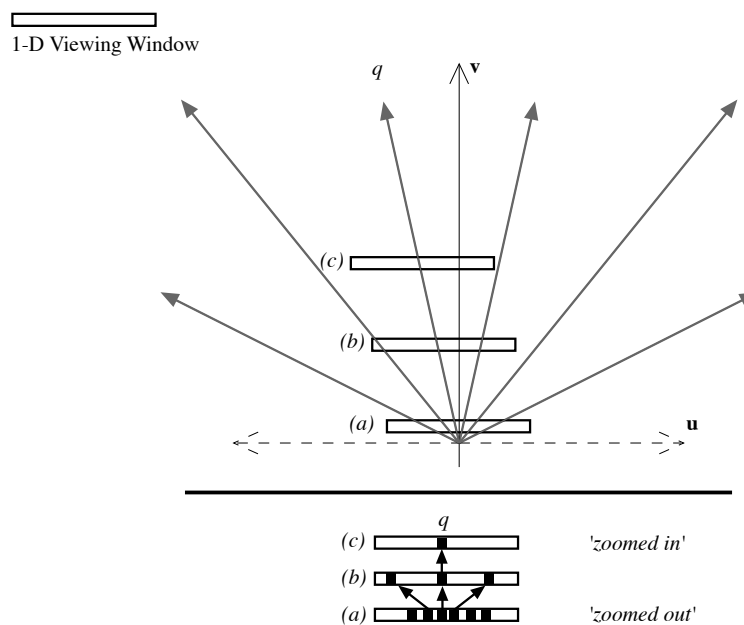


Figure 8. Pad++ help screen with oval document layout

Figure 9. A one dimensional space-scale diagram of six points as the view zooms in from (a) to (b) to (c) around point q

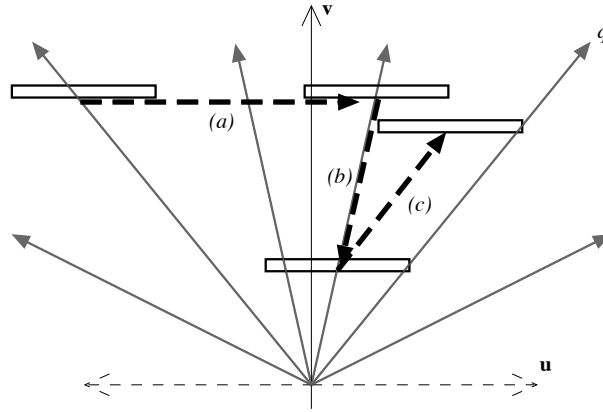


Figure 10. Basic pan-zoom trajectories are shown in the heavy dashed lines; (a) is pure pan, (b) is pure zoom, (c) is zoom around point q

the origin. We call these lines *great rays*. In the 2-D analog, whole 2D pictures would be stacked up at all magnifications, forming a 3D space-scale diagram, with points still becoming great rays and 2D regions becoming cones.

For comparison, compare this with a standard one-dimensional world. Here, a standard viewer is a small one-dimensional window that shows a small piece of the world (e.g. a view of a local-piece of a time-line). As the window is panned around it moves to different parts of that time line. As it is zoomed in, it would narrow its scope and look at a smaller region of the time line in detail. As it is zoomed out, it looks at a larger section.

In space-scale diagrams, however, while the viewing window is also represented as a one-dimensional segment, it has a constant size and is located at a particular place in both space and scale. Thus, as the user pans and zooms around the world, the viewing segment is moved rigidly (i.e. without changing its size) in space-scale. A whole sequence of such movements can be represented by a path through the space-scale diagram (Figure 10). Thus, the first advantage of these diagrams is that, by reifying scale, they allow these multiscale movements to be represented statically and so are easier to analyze. For example, a pan operation becomes a horizontal part of such a path. A zoom becomes a movement along a great ray. Other types of movement correspond to curves of other characteristic shapes.

The ability of space-scale diagrams to represent pan-zoom movements as a path in space-scale has allowed us to solve two concrete problems in designing good pan-zoom interactions. Both concern situations where the system needs to move the user's view automatically to another point in the space. This might happen, for example, as the result of following some sort of hyperlink mechanism, or jumping to the result of some content-based search.

The first problem occurs when the interface needs to not only move the user to some other region of the world, but also needs to zoom in. The solution to doing these actions in parallel, jointly panning and zooming to the new view, is not as simple as it might seem. However, if one simply computes how much to pan and how much to zoom and does the two independently in parallel, the result is disconcertingly non-monotonic. The pan covers distance at a constant pace while the zoom-in is magnifying the world exponentially. The result is that the target location first rushes

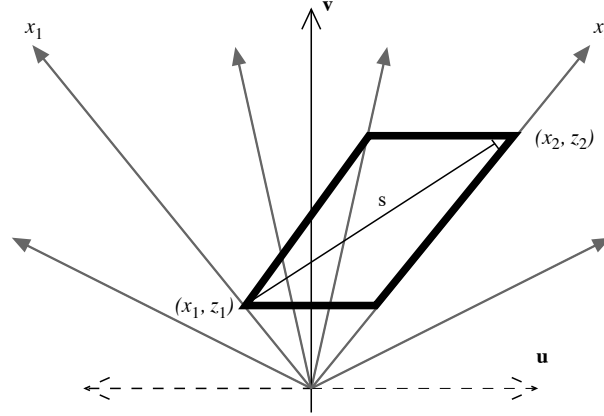


Figure 11. Solution to the simple joint pan-zoom problem. The trajectory s monotonically approaches point (x_2, z_2) in both pan and zoom

away due to the magnification, and only later does the pan catch up. Various hacks to fix this, taking logs and powers of various things, did not work.

Fortunately, using space scale diagrams, a monotonic approach to the target view is captured by a kind of parallelogram constraint on trajectories in space-scale. In Figure 11, a path from (x_1, z_1) to (x_2, z_2) that goes outside such a parallelogram is non-monotonic. If the path exits the sides of the parallelogram, it will violate the monotonicity requirement in space. If the path exits the top or bottom of the parallelogram, it will violate the monotonicity requirement in zoom. A simple path that is monotonic is just the diagonal of this parallelogram. A simple coordinate transform that defines these diagrams (given in [16]) allows one to define this path analytically, and yields a rather unintuitive hyperbolic relationship between the two. We have implemented the trajectory derived from the space-scale diagram analysis and have indeed found it far superior to any uninformed effort.

A second pan-zoom problem concerns the notion of shortest paths between two points in this pan-zoom parameter space. This is a curious question because, in space-scale motions, the shortest distance between two points is not generally a straight line. This is because while panning may be expected to take a time or have a cost that is linear in the spatial separation, zoom is logarithmic so that the fastest way to get from some point p to some point q that is far away would be very tedious by pan alone. It is in fact much shorter to zoom out, make a small pan, and then zoom in (see Figure 12.) Inspired by the space-scale diagrams we were able to define an information theoretic metric over space-scale interactions: the cost of a path is a function of the number of bits that would take to transmit a movie of the motion. Then we addressed the question of finding good paths through the space, i.e. make the movie as small as possible. We found that for points less than a few window widths away, a pure panning motion is pretty good, but for points far away, zoom must play a major role.

Another use of space-scale diagrams is to represent *semantic zooming*, where objects change not just their size but also their appearance when they are magnified. For example, an object could appear as a point when small. As it grows, it could then in turn appear as a solid rectangle, then a labeled rectangle, then a page of text, etc.

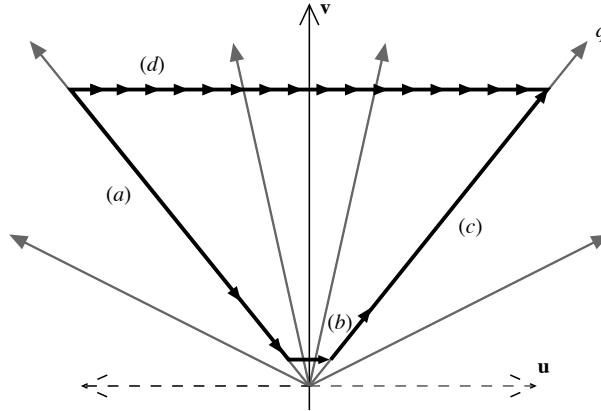


Figure 12. The shortest path between two points is often not a straight line. Here, each arrow represents one unit of cost. Because zoom is logarithmic, it is often shorter to zoom out (a), make a small pan (b) and zoom back in (c), than to make a large pan (d)

Figure 13 shows how semantic zooming differs from ordinary geometric zooming, in that the triangular regions change along the scale axis. By explicitly representing scale, the scale-dependent aspects of an object's representation can be made visible. We intend to use such diagrams to help create semantically zoomable objects. The idea is to provide an editing environment where transition boundaries could be moved or aligned by direct manipulation.

All these uses of multiscale diagrams capitalize on the fact that they statically represent scale so that multiscale concepts, which are inherently temporal, are more readily analyzed.

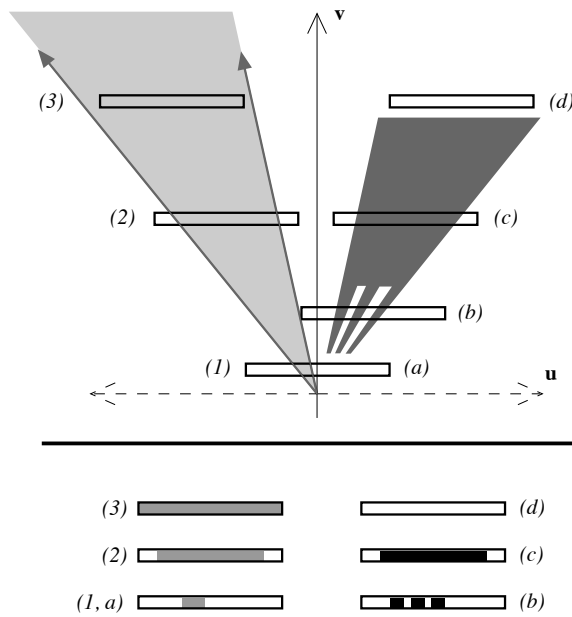


Figure 13. Semantic zooming. Bottom slices show views at different points

5. Procedural Animation

We are also using Pad++ as a substrate for building user-definable animated objects such as complex interface widgets. We have recently applied the same techniques to create animated human-like actors [27]. Although the widgets are much simpler, we employ the same mechanisms that allow us to control human-like movements and gestures to simulate personality and intentionality. The ultimate goal is to support an informational physics in which objects animate naturally. Using these tools, the Pad++ application designer can always convey to the user a clearly structured animated narrative instead of merely an assortment of disjoint temporal events.

We approach this goal by providing a mechanism for the definition of moveable graphical objects. In addition, we define high-level hierarchical control mechanisms for the movements. We are starting to define simple widgets such as buttons and sliders at a behavioral level that makes it easier for application developers to easily change the look and feel of an application. While the widget definitions we supply have a traditional Motif-like look and feel, a designer can easily change their visual style or interaction mechanism.

In addition, we are exploring novel widgets that take advantage of the Pad++ zooming environment. We used our extension mechanisms to implement a choice widget that provides an alternative to the traditional pop-up menu. Figure 14 shows two views of the same widget. The view on the left shows a zoomed out view. Here the widget just shows its current value. On the right we have zoomed into the widget and now the available choices become visible for user selection.

These widgets are implemented with our KPL rendering language. This language was designed to allow very fast run-time recompilation compact representation and efficient execution (roughly 100 times faster execution time than Tcl). It is a post-fix

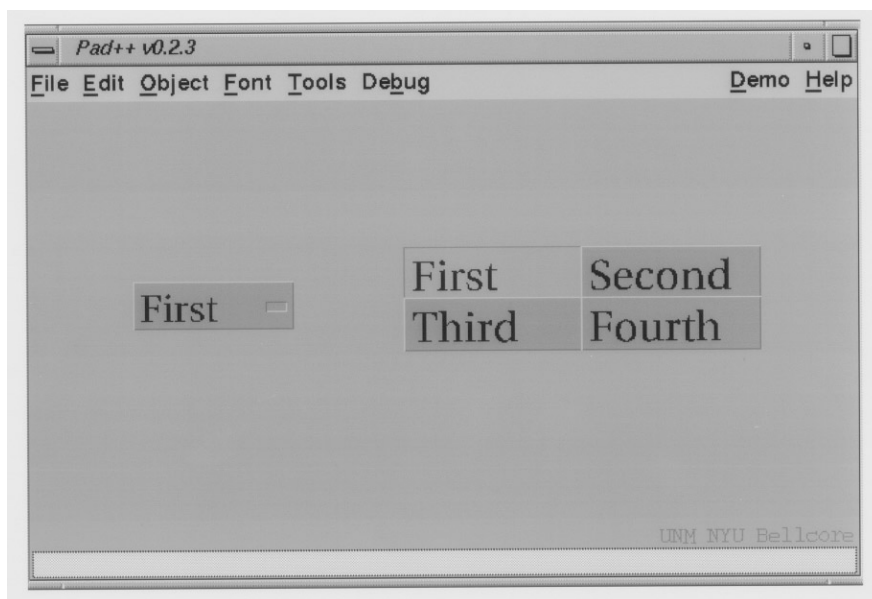


Figure 14. A prototype zoomable choice widget

stack language whose simple structure allows execution roughly 10 times faster than other interpreted or byte-compiled languages. KPL's speed allows us to execute scripts during each render. Without this efficient mechanism, we would only be able to render items pre-defined in the C++ substrate.

The next step uses KPL to create complex animations by the definition of simple repetitive motions of objects based on stochastic processes along with a built-in mechanism to make an automatic transition between different motions. The stochastic processes are defined by rotation axes, periods and magnitudes with some coherent noise [27] applied to give more natural behavior. The mechanism to change between motions gives the hierarchical control described above.

By changing the parameters of the stochastic movement in response to the environment and chaining sequences of motions, together with the transitioning mechanism, we are able to build complex animated behavior in the user interface.

We handle transitions between two actions having different tempos via morphing approach. At the start of the transition, we use the tempo of the first action, and at the end, we use the tempo of the second action. During the time of the transition, we continuously vary the speed of the master clock from the first to the second tempo. In this way, any phase dependent synchronization of the two actions is always preserved during transitions. We may also define new actions as extended transitions between two or more other actions. When there are multiple actors, each actor maintains its own individual tempo.

A related notion that we are exploring is peripheral attention. How does an actor convey that a process is proceeding normally or abnormally, without distracting the user from his/her current tasks? This is especially important in a zoomable environment where the ability to provide peripheral awareness of processes is an important attribute of the paradigm.

We are also studying the semantics of the discrete state transitions that visually represent shifts in attention. In this way an actor on the Pad++ surface can quickly convey to users which other actors and users it is interacting with. We are also interested in determining to what extent we can encode the texture of interactions in order to convey the visual impression of complex activities going on at different scales without requiring all the detail to be specified. We suspect that some of the same techniques used in character animation might be effective here too.

6. Implementation

Pad++ is implemented in C++ under various versions of the Unix operating system using the standard X graphics library system. It currently runs on SGIs, Suns, IBM RS-6000s, PCs running Linux, and should be trivially portable to other Unix systems. Pad++ is implemented as a widget in Tcl/Tk and thus allows applications to be written in the interpreted Tcl language. All Pad++ features are accessible through Tcl making it unnecessary to write any C++ code for new applications.

6.1. Efficiency

In order to keep the animation frame-rate up as the dataspace size and complexity increases, we utilized several standard efficiency methods in our implementation

which taken together create a powerful system. We have successfully loaded over 600 000 objects (with the directory browser) and maintained interactive rates of about 10 frames per second. Even when objects are not visible, appropriate checks must be done each time there is movement to see if those objects should now be visible. The key is that the rendering system takes a time roughly proportional to the number of visible objects, independent of the number of objects in the database (on average).

Briefly, the efficiency methods we use in Pad++ include:

- **Spatial Indexing:** Objects are stored internally in a hierarchy based on bounding boxes which allow fast indexing to visible objects.
- **Clustering:** Pad++ automatically restructures the hierarchy of objects to maintain a balanced tree which is necessary for the fastest indexing.
- **Refinement:** Renders fast while navigating by using lower resolution, and not drawing very small items. When the system is idle for a short time, the scene is successively refined, until it is drawn at maximum resolution.
- **Level-Of-Detail:** Renders items differently depending on how large they appear on the screen. If they are small, renders them with lower resolution.
- **Region Management:** Only updates the portion of the screen that has been changed. Linked with refinement, this allows different areas of the screen to refine independently.
- **Clipping:** Only renders the portions of large objects that are actually visible. This applies to images and text.
- **Adjustable Frame Rate:** Animations and zooming maintain constant perceptual flow, independent of processor speed, scene complexity, and window size. This is accomplished by rendering more or fewer frames, as time allows.
- **Interruption:** Slow tasks, such as animation and refinement, are interruptible by certain input events (such as key-presses and mouse-clicks). Animations are immediately brought to their end state and refinement is interrupted, immediately returning control to the user.
- **Ephemeral Objects:** Certain objects that represent large disk-based datasets (such as the directory browser) can be tagged ephemeral. They will automatically get removed when they have not been rendered for some time, and then will get reloaded when they become visible again.
- **Optimized Image Rendering:** The code to render zoomed images has been very carefully optimized and allows real-time zooming of high-resolution images.

6.2. Scripting Language Interface

An important consideration in the design and implementation of Pad++ is how to create a very fast and efficient graphics system, and yet still make it extensible. We wanted to make sure that we and others would be able to experiment easily with new interface mechanisms. Originally, Pad++ was implemented entirely in C++, making it very difficult for anyone but the authors to add new objects and interactions. Even for the authors, going through the compile and link cycle was very slow and tedious, making it difficult to do much experimentation.

We decided to create an interpreted scripting language interface to Pad++ to get around this problem. This approach is becoming quite common, and works well as long as the scripting language is at the right level. On one side, you want as much as

possible to be in the scripting language so that the system is as easy to modify as possible. On the other side, it is critical that all speed-critical code be written as efficiently as possible. In a system like ours, there are three classes of code, each of which have different speed requirements:

- Create objects: Slow—Scripting language is fine
- Handle events: Medium—Small amount of scripting language is ok
- Render scene: Fast—C++ or byte-compiled languages only

Rendering is done in C++ (for built-in Pad++ items) or in an efficient byte-compiled language such as KPL (for user defined widgets or animated items). This results in animation performance which is quite good, even on Linux based PC platforms.

We chose Tcl [24] as our primary scripting language, largely because it comes in combination with Tk, a Motif-like library for creating graphical user interfaces. Pad++ is built as a new widget in Tk. This allows it to be used in combination with standard, non-zooming widgets such as menubars, buttons, slidets, etc. This lets us make complete applications while we build and debug widgets within Pad++. Just as importantly, it provides a mechanism to compare zoomable interfaces with traditional interface mechanisms in the same system.

The Tcl interface to Pad++ is designed to be very similar to the interface to the Tk Canvas widget (which provides a surface for drawing structured graphics). While Pad++ does not implement everything in the Tk Canvas yet, it adds many extra features. The Tcl interface to Pad++ is summarized here to give a feel for what it is like to program Pad++.

We are also experimenting with other scripting languages which are better suited to some tasks—primarily those requiring higher speeds. As mentioned previously, we use KPL for high-speed animations. We also are considering incorporating an alternative language, such as Scheme or Java, for more general programming which needs high speed interaction.

6.3. TCL Interface

There are many commands that create and manipulate objects, each referring to the object's unique integer id. Objects may be grouped by using *tags*, a mechanism for associating data with each object. Every command can be directed to either a specific object id or to a tag, in which case it will apply to all objects that share that tag. Each Pad++ widget has its own name, and all commands start with the name of that widget. In the examples that follow, the name of the widget is `.pad`.

Examples:

- A red rectangle with a black outline is created whose corners are at the points (0, 0) and (200, 100):
`.pad create rectangle 0 0 200 100 -fill red -pen black`
- Put item number 5 at the point (30, 30), make the object twice as big, and make the object anchored at that point on its northwest corner:
`.pad itemconfig 5 -anchor nw -place "30 30 2"`
- Specify that item number 5 should be visible only when its largest dimension is greater than 20 pixels and less than 100 pixels.
`.pad itemconfig 5 -minsize 20 -maxsize 100`

It is straightforward to get scripts evaluated when specific events hit objects or groups of objects. Simple macros get expanded within the event script to specify information specific to that event. Some examples follow:

- Make all items with tag `foo` turn blue when the left button of the mouse is pressed over any of those objects:

```
.pad bind foo <ButtonPress-1> {
    .pad itemconfig foo -fill blue
}
```
- This is a single event binding for a group of objects that affects just the object clicked on, using the macro `%0` to expand to the specific object:

```
.pad bind foo <ButtonPress-1> {
    .pad itemconfig %0 -fill blue
}
```

Some basic navigation and searching mechanisms are provided by the Tcl interface. A few basic ones are:

- Smoothly go to the location (100, 0) with a magnification of 5, and take 1000 milliseconds for the animation:

```
.pad moveto 100 0 5 1000
```
- Smoothly go to the location such that object number 37 is centered, and fills three quarters of the screen, and take 500 milliseconds for the animation:

```
.pad center 37 500
```
- Return the list of objects ids that contain the text `'foo'`

```
.pad find withtext foo
```

6.4. Events

As briefly mentioned, it is possible to attach event handlers to items on the Pad++ surface so that when a specific event (such as `ButtonPress`, `KeyPress`, etc.) hits an item, the appropriate event handler is evaluated. This system operates much as it does with the Tk Canvas widget, but there are several significant additions:

- **Extra Macro Expansions**

When a command is invoked, several substitutions are made in the text of the command that describe the specific event that invoked the command. In addition to the substitutions that the Tk `bind` command makes, Pad++ makes a few more. These include mechanisms to find the pad widget and item that actually received the event, the coordinates of the event in Pad++ coordinates, which portals the event went through, and a few other related items.

- **New Events**

Several new events were created that get fired at special times, depending on the semantics of the event. `<Create>` gets fired whenever new Pad++ items are created. `<Modify>` gets fired whenever an item is modified. `<Delete>` gets fired whenever an item is deleted. `<Write>` gets fired whenever an item is written out with the Pad++ `write` command. `<PortalIntercept>` gets fired just before an event passes through a portal. If the event handler executes the `break` command, then the event stops at the portal and does not pass through.

- **User-Specified Modifiers**

Event handlers are defined by sequences of the same format as the Tk bind command. A sequence contains a list of modifiers which are direct mappings hardware such as the shift key, control key, etc. Event handlers only fire sequences with modifiers that are active, as defined by the hardware.

Pad++ allows user-defined modifiers where the user can control which one of the user-defined modifiers is active (if any). The advantage of modifiers is that many different sets of event bindings may be declared all at once—each with a different user-defined modifier. Then, the application may choose which set of event bindings is active by setting the active user-defined modifier. This situation comes up frequently with many graphical programs where there are modes, and the effect of interacting with the system depends on the current mode.

6.5. Callbacks

In addition to the event bindings that every item may have, every Pad++ item can define Tcl scripts associated with it which will get evaluated at special times. There are currently three types of these callbacks:

- **Render Callbacks**

A render callback script gets evaluated every time the item is rendered. The script gets executed when the object normally would have been rendered. By default, the object will not get rendered, but the script may render the object at any time with the `renderitem` function. An example follows where item number 22 is modified to call the Tcl procedures `beforeMethod` and `afterMethod` surrounding the object's rendering.

```
.pad itemconfig 22 -renderscript {
    beforeMethod
    .pad renderitem
    afterMethod
}
```

Instead of calling the `renderitem` command, an object can render itself. Several rendering routines are available to render scripts, making it possible to define an object that has any appearance whatsoever. Items which define a render script are called *procedural objects* and are used for creating animated objects (those that change the way they look on every render) and custom objects. They also can be used to implement semantically zoomable objects, since the size of an object is available within the callback.

- **Timer Callbacks**

A timer callback script gets evaluated at regular intervals, independent of whether the item is being rendered, or receiving events.

- **Zooming Callbacks**

Zooming callback scripts are evaluated when an item gets rendered at a different size than its previous render, crossing a pre-defined threshold. These are typically used for creating efficient semantically zoomable objects. Since many objects do not change the way they look except when crossing size borders, it is more efficient to avoid having scripts evaluated except for when those borders are crossed.

6.6. Extensions

Pad++ is extensible with Tcl scripts (i.e. no C/C++ code). This provides an easy to use mechanism to define new Pad++ commands as well as compound object types that are treated like first-class Pad++ objects. That is, they can be created, configured, saved, etc., with the same commands you use to interact with built-in objects, such as lines or text. These extensions are particularly well-suited for widgets, but can be used for anything.

Extensions are defined by creating Tcl commands with specific prefixes. Each extension is defined by three commands which allow creation, configuration and invocation of the extension, respectively. Defining the procedures makes them automatically available to Pad++. No specific registration is necessary. All three procedure definitions are necessary for creation of new Pad++ object types, but it is possible to define just the command procedure for defining new commands without defining new object types.

7. Physics-Based Strategies For Interface Design

Exploration of Pad++ is part of a research program to develop alternative strategies for interface design. Our goal is to move beyond mimicking the mechanisms of earlier media and to start to more fully exploit radical new computer-based mechanisms. We propose an information physics view of interface objects that we think provides an effective complement to traditional metaphor-based approaches.

While an informational physics strategy for interface design certainly involves metaphor, we think there is much that is distinctive about a physics-based approach. Traditional metaphor-based approaches map at the level of high-level objects and functionality. They yield interfaces with objects such as windows, trash cans and menus, and functions like opening and closing windows and choosing from menus. While there are ease-of-use benefits from such mappings, they also orient designers towards mimicking mechanisms of earlier media rather than towards exploring potentially more effective computer-based mechanisms. Semantic zooming is but one example mechanism that we think arises more naturally from adopting an informational physics strategy. Even geometric zooming, especially with the orders of magnitude possible in Pad++, is not a mechanism that traditional metaphors would lead designers to investigate.

We are not the first to follow a physics-inspired course in thinking about interface design. It derives, like most interesting interface ideas, from the seminal work of Sutherland [31] on Sketchpad. Simulations and constraint-based interfaces that led to the development of direct manipulation style interfaces are other examples of this general approach. They too derive from Sutherland and continue to inspire developments. Recent examples include the work of Borning and his students [5, 6]. Witkin [33] in particular has taken a physics-as-interface approach to construction of dynamic interactive interfaces.

Smith's Alternate Reality [28, 29] and languages such as Self [32] are also examples of following a physics-based strategy for interface design. These systems make use of techniques normally associated with simulation to help 'blur the distinction between data and interface by unifying both simulation objects and interface objects as concrete objects' [9]. More importantly, they are based on implementation of mechanisms at a different level than is traditional. Smith, for example, gives users

access to control of parameters of the underlying physics in his Alternate Reality Kit. With this approach comes the realization that one can do much more than just mimic reality. As Chang and Unger [9] point out about their use of cartoon animation mechanisms in Self, ‘adhering to what is possible in the physical world is not only limiting, but also less effective in achieving *realism*.’

It is important to look at the costs as well as the benefits of traditional, metaphor-based strategies. They can lead away from exploration of new mechanisms and limit views of possible interfaces in at least four ways.

First, metaphors necessarily pre-exist their use. Pre-Copernicans could never have used the metaphor of the solar system for describing the atom. In designing interfaces, one is limited to the metaphorical resources at hand. In addition, the metaphorical reference must be familiar in order to work. An unfamiliar interface metaphor is functionally no metaphor at all. One can never design metaphors the way one can design self-consistent physical descriptions of appearance and behavior. Thus, as an interface design strategy physics, in the sense described above, offers more design options than traditional metaphor-based approaches.

Second, metaphors are temporary bridging concepts. When they become ubiquitous, they die. In the same way that linguistic metaphors lose their metaphorical impact (e.g. *foot of the mountain* or *leg of table*), successful metaphors also wind up as dead metaphors (e.g. file, menu, window, desktop). The familiarity provided by the metaphor during earlier stages of use gives way to a familiarity with the interface due to actual experience.

Thus, after a while, it is the actual details of appearance and behavior (i.e. the physics) rather than any overarching metaphor that form much of the substantive knowledge of an experienced user. Any restrictions that are imposed on the behaviors of the entities of the interface to avoid violations of the initial metaphor are potential restrictions of functionality that may have been employed to better support the users’ tasks and allow the interface to continue to evolve along with the users’ increasing competency.

Similarly, the pervasiveness of dead metaphors, such as files, menus and windows, may well restrict us from thinking about alternative organizations of interaction with the computer. There is a clash between the dead metaphor of a file and newer concepts of persistent distributed object hierarchies.

Third, since the sheer amount and complexity of information with which we need to interact continues to grow, we require interface design strategies that *scale*. A traditional metaphor-based strategy does not scale. A physics approach, on the other hand, scales to organize greater and greater complexity by uniform application of sets of simple laws. In contrast, the greater the complexity of the metaphorical reference, the less likely it is that any particular structural correspondence between metaphorical target and reference will be useful. We see this often as designers start to merge the functionality of separate applications to better serve the integrated nature of complex tasks. Metaphors that work well with the individual simple component applications typically do not integrate smoothly to support the more complex task.

Fourth, it is clear that metaphors can be harmful as well as helpful since they may well lead users to import knowledge not supported by the interface. Our point is not that metaphors are not useful but that, as the primary design strategy, they may well restrict the range of interfaces designers consider and impose less effective trade-offs

than those designers might come to if they were led to consider a larger space of possible interfaces.

There are, of course, also costs associated in following a physics-based design strategy. One cost is that designers can no longer rely as heavily on users' familiarity with the metaphorical reference (at least at the level of traditional objects and functionality), and so physics-based designs may take longer to learn. However, the power of metaphor comes early in usage and is rapidly superceded by the power of actual experience. One might want to focus on easily discoverable physics. As is the case with metaphors, all physics are not created equally discoverable or equally fitted to the requirements of human cognition.

8. Future Directions

To adequately explore the effectiveness of the Pad++ substrate and the informational physics design strategy discussed here will require development of a wide range of applications. One domain we plan to investigate is construction of active documents. Most tools for interacting with documents (like World-Wide Web browsers such as Mosaic and Netscape) predefine all of the interactive widgets within the client. Hooks are provided so that documents may access those widgets but there is no method to provide new ones, except to re-define the standards, modify the client and distribute the client to enough of the user population so it becomes the new standard in practice.

Pad++'s extensibility ensures that new widgets can be defined by scripts which can be included with a document. This will allow documents to provide new forms of interactivity without depending on the client to supply it. We are currently in the design stages of an extension to HTML, we call the MultiScale Markup Language (MSML), that will be the markup language to describe documents within Pad++. MSML will allow logical formatting of documents with different sized components and will provide a method for allowing Pad++ scripts to be included with documents—allowing truly active documents.

In addition to data visualizations, we are investigating the use of Pad++ as a replacement for the standard windowing system. PadWin currently consists of a few basic semantically zoomable gauges which display statistics such as the list of tasks being scheduled, the state of the printer queue or the names of people who are logged on. We intend to extend these tools so that most of the computers resources and facilities are accessible through navigation within PadWin. We are also producing a suite of zoomable applications for use in PadWin.

In order to support existing non-zoomable applications, PadWin will incorporate a mechanism to control the placement of application windows on the screen to make them blend into the Pad++ surface. By mapping, unmapping and moving these windows appropriately, PadWin will act as an extended virtual window manager where the effective screen size is huge, and where zoomable and non-zoomable applications reside side by side.

Pad++ also seems well-suited to a collaborative work environment. While the original Pad implementation allowed some basic shared workspaces (running from a single process displaying on multiple X servers), we are designing a more sophisticated approach. The goal is to be able to use portals to look remotely on to any Pad++ surfaces on the network (assuming that the right permissions are set). Each

user's system will contain a spatial database server that will send updates to all other systems that have portals looking on to it. With this approach, there may be a lag in retrieving others' data but, once it arrives, it will be cached within the local system so the high-speed interactivity of Pad++ will not be lost.

Finally, we are building a completely visual interface to Pad++ for creation of an interactive visual dataspace. Multimedia authoring tools such as MacroMedia DirectorTM and Apple's Multimedia Authoring ToolTM are letting visual designers without programming experience create beautiful and complex interactive hypertext data retrieval systems. As we discussed with the layout of HTML, however, having a huge data surface potentially alleviates some of the problems of navigating within a large hypertext document. To this end, we are building a set of tools that will allow non-technical visual designers to create interactive zoomable multimedia systems.

9. Availability

The Pad++ substrate is approaching the point where we can start to make it available to a wider community. Our goal is to make it freely available for non-commercial use. See the Pad++ project home page (<http://www.cs.unm.edu/pad++>) for current information.

Acknowledgments

This work was supported in part by ARPA contract N66001-94-C-6039 to the University of New Mexico. We especially appreciate the support we have received from Craig Wier as part of the new HCI Initiative at ARPA. We thank David Fox and Matthew Fuchs at NYU and Eric De Mund, David Vick and Jason Stewart at UNM for enjoyable discussions about zoomable interfaces. We also would like to acknowledge members of the Computer Graphics and Interactive Media Research Group at Bellcore for discussions shared during our continuing search for the best cheeseburger.

References

1. R. M. Baecker (1990) *Human factors and typography for more readable programs*. ACM Press, Denver.
2. B. B. Bederson, L. Stead & J. D. Hollan (1994) *Pad++: Advances in multiscale interfaces*. In: *Proceedings of ACM SIGCHI Conference (CHI'94)*. Addison-Wesley, Reading, MA.
3. B. B. Bederson & J. D. Hollan (1994) *Pad++: A zooming graphical interface for exploring alternate interface physics*. In: *Proceedings of ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM Press, New York.
4. E. A. Bier, M. C. Stone, K. Pier, W. Buxton & T. D. DeRose (1993) Toolglass and magic lenses: the see-through interface. In: *Proceedings of ACM SIGGRAPH Conference (Siggraph '93)*. Addison-Wesley, Reading, MA.
5. A. Borning (1979) *Thinglab: a constraint-oriented simulation laboratory*. Technical Report SSL-79-3, Xerox Palo Alto Research Center.
6. A. Borning & R. Duisberg (1986) Constraint-based tools for building user interface. *ACM Transactions on graphics*, 5(4), 345–374.
7. R. Brooks (1986) A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2(1), 14–23.
8. S. K. Card, G. G. Robertson & J. D. Mackinlay (1991) The information visualizer, an information workspace. In: *Proceedings of ACM Human Factors in Computing Systems Conference (CHI'91)*. Addison-Wesley, Reading, MA.

9. B.-W. Chang & D. Ungar. Animation: from cartoons to the user interface. In: *Proceedings of ACM Symposium on User Interface Software and Technology (UIST'93)*. ACM Press, New York.
10. S. Deerwester, S. T. Dunais, G. W. Furnas, T. K. Landauer & R. Harshman (1990) Indexing by latent semantic analysis. In: *Journal of American Society of Information Science* **41**, 391–407.
11. W. C. Donelson (1978) Spatial management of information. In: *Proceedings of 1978 ACM SIGGRAPH Conference*. Addison-Wesley, Reading, MA.
12. D. Ebert, *Texturing and Modeling, A Procedural Approach*. Academic Press, London.
13. S. G. Eick, J. L. Steffen & E. E. Sumner, Jr (1992) Seesoft: a tool for visualizing line-oriented software statistics. In: *IEEE Transactions on Software Engineering* **18**(11), 957–968.
14. K. M. Fairchild, S. E. Poltrock & G. W. Furnas (1980) SemNet: three-dimensional graphic representations of large knowledge bases. In: *Cognitive Science and its Applications for Human-Computer Interaction*. Lawrence Erlbaum Associates, Princetown.
15. G. W. Furnas (1986) Generalized fisheye views. In: *Proceedings of 1986 ACM SIGCHI Conference*. Addison-Wesley, Reading, MA.
16. G. W. Furnas & B. B. Bederson (in press) Space-scale diagrams: understanding multiscale interfaces. In: *Proceedings of ACM SIGCHI'95*. Addison-Wesley, Reading, MA.
17. M. Gleicher (1992) Briar: a constraint-based drawing program. In: *CHI'92 Formal Video Program*. Addison-Wesley, Reading, MA.
18. W. C. Hill, J. D. Hollan, D. Wroblewski & T. McCandless (1992) Edit wear and read wear. In: *Proceedings of ACM SIGCHI'92*. Addison-Wesley, Reading, MA.
19. W. C. Hill & J. D. Hollan (1994) History-enriched digital objects: prototypes and policy issues. *The Information Society*, **10**, 139–145.
20. J. D. Hollan, E. Rich, W. Hill, D. Wroblewski, W. Wilner, K. Wittenburg, J. Grudin & Members of the Human Interface Laboratory (1991) An introduction to HITS: human interface tool suite. In: *Intelligent User Interfaces* (Sullivan & Tyler, eds) pp. 293–337.
21. J. D. Hollan & S. Stornetta (1993) Beyond being there. In: *Proceedings of the ACM SIGCHI'92*. Addison-Wesley, Reading, MA.
22. G. Lakoff & M. Johnson (1980) *Metaphors We Live By*. University of Chicago Press, Illinois.
23. J. D. Mackinlay, G. G. Robertson & S. K. Card (1991) The perspective wall; detail and context smoothly integrated. In: *Proceedings of CHI'91 Human Factors in Computing Systems*. Addison-Wesley, Reading, MA.
24. J. K. Ousterhout (1994) *Tcl and the Tk Toolkit*. Addison Wesley, New York.
25. K. Perlin & D. Fox (1993) Pad: an alternative approach to the computer interface. In: *Proceedings of 1993 ACM SIGGRAPH Conference*. Addison-Wesley, Reading, MA.
26. K. Perlin (1985) An image synthesizer. In: *Proceedings of ACM SIGGRAPH '85* **19**, 287–293.
27. K. Perlin (1994) Danse interactif. In: *Video Proceedings of ACM SIGGRAPH '94*, **28**(3).
28. R. B. Smith (1986) The alternate reality kit: an animated environment for creating interactive simulations. In: *Proceedings of the 1986 IEEE Computer Society Workshop on Visual Languages*, Rome.
29. R. B. Smith (1987) Experiences with the alternate reality kit: an example of the tension between literalism and magic. In: *Proceedings of ACM CHI + GI '87 Conference*. ACM Press, New York.
30. M. C. Stone, K. Fishkin & E. A. Bier (in press) The movable filter as a user interface tool. In: *Proceedings of ACM SIGCHI'94*. Addison-Wesley, Reading, MA.
31. I. E. Sutherland (1963) Sketchpad: A man-machine graphical communications systems. In: *Proceedings of the Spring Joint Computer Conference*. Spartan Books, Baltimore, pp. 329–346.
32. D. Ungar & R. B. Smith (1987) Self: The Power of Simplicity. In: *Proceedings of OOPSLA '87 Conference*.
33. A. Witkin, M. Gleicher & W. Welch (1990) Interactive dynamics. *Computer Graphics* **24**(2), 11–21.

PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps

Benjamin B. Bederson

Human-Computer Interaction Laboratory

Computer Science Department, Institute for Advanced Computer Studies

University of Maryland, College Park, MD 20742

+1 301 405-2764

bederson@cs.umd.edu

<http://www.cs.umd.edu/hcil/photomesa>

ABSTRACT

PhotoMesa is a zoomable image browser that uses a novel treemap algorithm to present large numbers of images grouped by directory, or other available metadata. It uses a new interaction technique for zoomable user interfaces designed for novices and family use that makes it straightforward to navigate through the space of images, and impossible to get lost.

PhotoMesa groups images using one of two new algorithms that lay out groups of objects in a 2D space-filling manner. *Quantum treemaps* are designed for laying out images or other objects of indivisible (quantum) size. They are a variation on existing treemap algorithms in that they guarantee that every generated rectangle will have a width and height that are an integral multiple of an input object size. *Bubblemaps* also fill space with groups of quantum-sized objects, but generate non-rectangular blobs, and utilize space more efficiently.

Keywords

Zoomable User Interfaces (ZUIs), Treemaps, Image Browsers, Animation, Graphics, Jazz.

INTRODUCTION

There has been much work in recent years on information retrieval systems for multimedia, including systems concentrating on images. However, these systems focus on specifying queries or presenting results in a manner that helps users quickly find an item of interest. For image searches, in particular, there has been relatively little work on new interfaces, visualizations, and interaction techniques that support users in browsing images.

Image browsing is important for a number of reasons. First of all, no matter what information retrieval system is being used, the user has to browse the results of the search. It is certainly important to build query systems that help users get results that are as close to what is wanted as possible. But there will always be images that need to be browsed visually to make the final pick.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST '01 Orlando FLA

Copyright ACM 2001 1-58113-438 -x/01/11...\$5.00

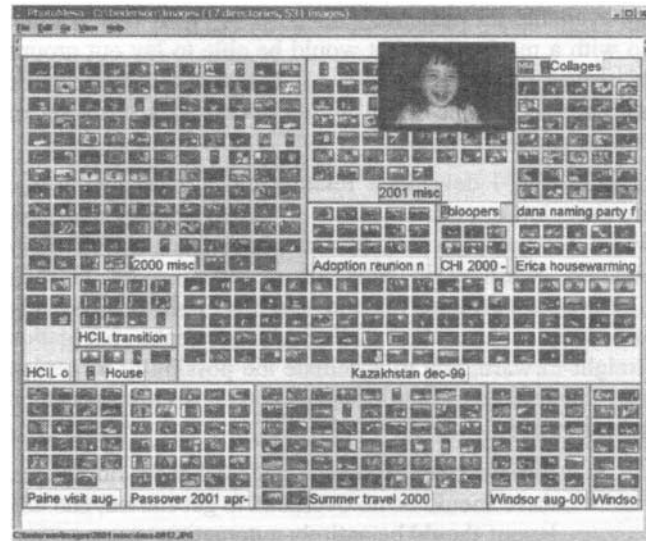


Figure 1: Screen snapshot of PhotoMesa with over 500 images in 17 groups.

Most image browsing systems present the images as a grid of thumbnails that the user can scroll through with a vertical scrollbar, and see a high resolution version of the image with some mouse interaction. There are also a few alternative designs, such as manually constructed digital photo albums, and one commercial zoomable image browser.

A second reason for needing new image browsers is more subtle, and was actually my primary motivation for doing the present work. Sometimes, people browse images just for the pleasure of looking at those images, and they often do it with other people. This is especially true for personal photos. As people take more digital family pictures, we need better tools to support users in home settings as they look at those pictures together on a computer screen. Looking at home photos has a lot of overlap with traditional retrieval systems. People still want to be able to find photos of particular people and events, etc. However, they are less likely to be time pressured to find a particular photo, and more likely to be interested in serendipity – that is, finding photos they weren't looking for [6].

I found I needed better tools to look at pictures with my two-year-old daughter. I did not want to spend the time to make custom “albums”. In addition, I found using traditional software with a grid of thumbnails, scrollbars, and popup viewer windows unpleasant in this context. I wanted to concentrate on the images – and more importantly, as I was looking at the photos with my daughter, it was crucial that she be an active part of the interaction, and not just a passive bystander.

Motivated by the need of a tool that would support browsing of images with my family, I started to investigate techniques for presenting collections of images or other visual data. While much work has been done on visualizing complex datasets, surprisingly few techniques are available for presenting images. My goal was to come up with a mechanism that would be able to lay out groups of images automatically in a way that would offer a simple interface to browse while giving access to a large set of images and their context.

To this end, I developed PhotoMesa, a zoomable image browser that organizes images in a two-dimensional grid, where images with a shared attribute (such as directory location, nearness in time, or a shared word in their filename) are grouped together (Figure 1). It uses zooming and simple interaction techniques to make navigation straight-forward, and to eliminate the possibility of getting lost. In building PhotoMesa, I kept the following design goals in mind:

- Simple to use (interaction should focus on images, there should be no overhead to get started, and any layout should be entirely automatic)
- Work well for family-use settings, encouraging shared co-present use
- Support collections of photos, and use screen space efficiently

To lay out the groups of images automatically, I ended up developing two new algorithms, called *quantum treemaps* and *bubblemaps*. Quantum treemaps are a variation on existing treemap algorithms [21]. Treemaps are a family of algorithms that partition two-dimensional space into regions that have an area proportional to a list of requested areas. The problem with existing treemap algorithms is that they return areas of arbitrary aspect ratios. A requirement of photo display is that the regions that show groups of photos must have dimensions that are integer multiples of the dimensions of the photos – that is, they must be sized to contain *quantum*, or indivisible contents. The use of treemaps to display images is the first known use of treemaps to display visual content, such as images, rather than just using the size and color of the rectangles to visualize two numerical attributes of a dataset.

The bubblemap algorithm generates non-rectangular groups. The groups are generated with a grid-based recursive fill algorithm. They fill all the cells in a grid

leaving almost no unused space, and generate groups of images that are approximately rectangular or circular.

This paper describes PhotoMesa and the quantum treemap and bubblemap layout algorithms. All the software described in this paper is written in Java 2, is fully functioning as described, and is available at <http://www.cs.umd.edu/hcil/photomesa>.

RELATED WORK

As mentioned previously, the standard way to let users browse a set of images is with a grid of thumbnails with a vertical scrollbar. Clicking on an image thumbnail usually brings up a window with the high-resolution version of the image. The user then has to manage the open windows manually, and close them when they are no longer needed. One good commercial example of this approach is ACDSee which offers a clean interface and fast interaction [1].

This approach has been extended by a research group at the University of Maryland developing PhotoFinder [16, 22]. It lets users organize photos into “collections” which are displayed with a representative image that the user selects. The interface first shows collections, and selecting a collection displays a traditional grid of thumbnails. PhotoFinder avoids the problem of window management, by displaying high-resolution photos in a pane within the interface. The PhotoFinder project concentrates on interfaces for managing and searching a database of meta information, but the browsing interface is essentially a polished traditional approach.

Document Lens is a technique that uses 2D fisheye distortion to present a grid of thumbnails of documents with a mechanism to zoom one document up to a readable size in place [18]. Document Lens, however, presents just a single collection of objects at a time.

Others have looked into automated algorithms for clustering semantically related information, and presenting the results visually. Hascoët-Zizi and Pediotakis built such a system for a digital library retrieval system, showing the available thesaurus as well as results of searches [14]. Platt has built a system for automatically clustering photos, and extracting representative photos for each cluster [17].

Several groups have investigated applications of images for story telling or sharing in the home. The Personal Digital Historian project at MERL is building a circular display on a tabletop intended for several people to interact with images together. The design includes search by several kinds of metadata, but the mechanism for interacting with many images was not described in detail [20]. This is an example of support for co-present use which is a theme described in some of the author’s prior work [23].

A group at Ricoh is building a dedicated portable story-telling device based on the construction of sequences of images. It has a dedicated hardware interface for selecting sequences of images which can then be annotated with audio, and played back when telling the story associated with those images [6].

For a pure software approach, we and others have built Zoomable User Interfaces (ZUIs) for image browsing. ZUIs are interfaces that present information on a large flat space where the user can smoothly zoom into the space to see information in more detail, or zoom out to get an overview. ZUIs have the potential advantages that they are easy to comprehend, and they give a consistent and easy to use interface for getting an overview of the information, and seeing more detail.

An earlier ZUI-based image browser was ZIB (Zoomable Image Browser) [11]. ZIB combined a zoomable presentation of a grid of images with a search engine (that searched metadata), and a history mechanism to access previous searches. However, ZIB provided access to only a single group of images, and used manual zooming which was difficult to use.

The approach started in ZIB was continued in a new project that is creating an interface for elementary school-aged children to find multimedia information in a digital library [12]. This project, called SearchKids, presents visual results in a zoomable interface with a simpler interaction mechanism that PhotoMesa is based on.

Another ZUI-based image browser is currently available commercially by Canon, and is called ZoomBrowser EX [2]. The Canon browser presents a hierarchy of images (either manually constructed, or imported from a disk hierarchy) with containment. The top level shows a grid of squares, each of which contain a grid of image thumbnails and/or smaller squares that show more thumbnails, etc. It uses a layout very similar to what we used earlier in the Pad++ directory browser [8]. This layout has the disadvantage that all directories are the same size, and the contents are scaled to fit so that images in large directories are scaled small so as to be unreadable.

The interaction is to click on a square, and the contents of the square are smoothly zoomed into. Clicking on an image brings up a traditional high-resolution image viewer in a separate window. Clicking on a special zoom-out button zooms out to the next level in the hierarchy. There is also a magnification mode which zooms in a fixed amount each click, rather than zooming into the next level of the hierarchy.

PHOTOMESA

PhotoMesa allows the user to view multiple directories of images in a zoomable environment, and uses a set of simple navigation mechanisms to move through the space of images. It also supports clustering of images by metadata available from the file system. It requires only a set of images on disk, and does not require the user to add any metadata, or manipulate the images at all before browsing, thus making it easy to get started with existing images.

PhotoMesa is written entirely in Java 2, and is built using the Jazz framework for Zoomable User Interfaces [9]. The name *PhotoMesa* derives from the Spanish word *mesa* which means table, but is commonly used in the US

southwestern states to describe the natural volcanic plateaus which are high and have flat tops. Standing atop a mesa, you can see the entire valley below, much as you can see an overview of many photos in PhotoMesa.

To start using PhotoMesa, a user opens a directory, or a set of directories, and PhotoMesa lays out the directories of images in a space-filling manner as shown in Figure 1, using a quantum treemap to create one rectangular group for each directory. Even though a hierarchical directory structure is read in, the images are displayed in a flattened, non-hierarchical manner. The rationale for this is that users looking at images are primarily interested in groups of photos, not at the structure of the groups. In addition, the interface for presenting and managing hierarchies of groups would become more complicated, and simplicity was one of the goals of the PhotoMesa. However, this is a design characteristic of PhotoMesa, not of the of the treemap algorithms which can be applied hierarchically.

As the user moves the mouse, the group the mouse is over is highlighted, and the label is shown in full (it may have been clipped if there wasn't room for it). Then when the user clicks, the view is smoothly zoomed in to that group. Now, a highlight showing a set of images under the mouse lets the user know which images will be focused on when the mouse is clicked again. The number of images highlighted is chosen to be enough to fill about half of the screen so that the user will be able to drill down quickly to a full-resolution single image. At any point, the user can press the right button (or Enter key) to zoom out to the previous magnification. In addition, the user can double-click on an image to zoom all the way into that image and avoid intermediate zoom levels, or the user can double-right click to zoom all the way out to the top level.

The user can also press alt-left/right arrows to move back and forth in their history of views. Or, they can press the arrow keys to pan up, down, left or right. When zoomed all the way into a full-resolution image, the arrow keys stay within the current group of images, wrapping as necessary. When zoomed out so more than one image is visible, the arrow keys move across groups to let the user explore the entire space.

At all times, if the cursor is left to dwell over an image thumbnail for a short time, that thumbnail is zoomed up until it is 200 pixels wide overlaying the other, unchanged images (Figure 1). This preview is immediately removed whenever the mouse is moved.

While it is not necessary for users to do any authoring to browse images with PhotoMesa, they are allowed to change the color of image groups (although group background colors are assigned by default). This can make it easier to make sense of the large display of images since the colored areas can act as landmarks which are known to be effective navigation aids [15].

PhotoMesa supports drag-and-drop to let users directly export images to email, or other applications. Since

emailing photos is a significant use, PhotoMesa automatically reduces the resolution and quality of images when they are dragged out of PhotoMesa. This resolution reduction is controllable through a preference panel. This eliminates the need to go through a special processing step when emailing images.

While the support of browsing is the primary goal of PhotoMesa, it is also sometimes desirable to find images in a specific group, and it can be difficult to scan labels in a 2D space. So, a search pane is available that shows all the directories in order. Mousing over a label highlights the corresponding group of images, and clicking on a label zooms into that group. In addition, the search pane has a search box where users can search for images by words in their filename.

After PhotoMesa was built, and we started using it to browse directories of images, I realized that another way of thinking about what PhotoMesa was doing was presenting a large set of images clustered by directory. So I then added support for clustering by other data. Since I didn't want to require users to add metadata, PhotoMesa uses whatever data is already available in the file system, which is just file date and name. If a user selects view by year, PhotoMesa uses the file date to group all the currently opened photos by year, and creates a layout with one region per year. It does the same thing for viewing by month.

Another clustering technique takes advantage of the fact that people sometimes give meaningful filenames to their images, often with several words per image to describe the contents of the image (Figure 2). If a user selects view by "filename words", it parses the filenames of all of the open images, and creates one cluster for each unique word in a filename (as tokenized with all the standard delimiters and where filename extensions and numeric tokens are ignored). Thus, if an image has 3 words in its filename (such as "ben-eats-cake"), then that image will appear in 3 clusters (one for "ben", one for "eats", and one for "cake").

PhotoMesa computes multiple sized thumbnails for each image, and dynamically loads the appropriate one. In this manner, it maintains good performance, even with large numbers of images. The thumbnails are created the first time an image is loaded, and cached in a special directory managed by PhotoMesa.

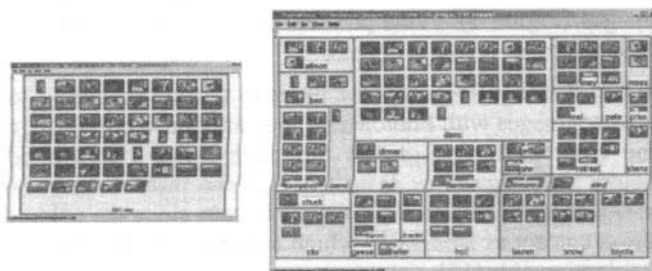


Figure 2: A directory of images (left), and the same images grouped by filename words (right).

The design of PhotoMesa presents an inherent difference compared to traditional scrolling thumbnail grids. The traditional approach has the advantage that it is searchable by navigating in one dimension (through vertical scrolling), while PhotoMesa requires navigation in two dimensions, which is typically harder for users. However, PhotoMesa has the advantage that the user can easily get an overview by zooming out. Through this interaction, the user can control the trade-off between the number of images shown and their resolution. This difference is a direct effect of the zooming nature of PhotoMesa. If a vertically oriented grid of thumbnails were zoomed out, the space would be mostly unused on either side of the linear list, and the display space would thus be largely wasted. Thus, it seems that a 2D zoomable interface and 1D displays of data are inherently incompatible.

USE OF PHOTOMESA

I have used PhotoMesa regularly with my two year old daughter for several months. We load in all of our family pictures (Figure 1) and sit together in front of a laptop computer. She will point at an area and I click and zoom in to it. I keep zooming in as she points at areas until we get all the way in to a single photo. I then zoom out one level, and if she asks to see another photo, I zoom into it. Otherwise, I zoom out another level until she sees something she is interested in. In this fashion, we look at the photos together, and she is able to stay in control and maintains a high level of interest. The zooming and smooth animation make it so that she is clearly able to follow what is going on, even though I operate the mouse.

In addition, over 9,000 people have downloaded PhotoMesa from the web. While this is obviously not a controlled study, it has been informative nevertheless. I have received very positive feedback, sometimes describing use scenarios I did not originally envision. One designer used it as a "disk mapper" to find out what was on her disk. Another put the software with photos on a CD and mailed it to family and friends. Others have envisioned embedding it in a range of applications, from supporting hobbyist aquarium logging to web-based photo sharing. Perhaps most importantly, several people reported they find it ideal to use with their families – supporting my original design goal.

QUANTUM TREEMAPS

In the course of developing PhotoMesa, I ran into a significant problem. I needed an automatic way to lay out groups of images in a visually simple manner that filled all the available space. I started to solve this by looking into treemap algorithms. Treemaps are a family of algorithms that are space-filling partitions of a two-dimensional area. Treemaps take as input, a list of n numbers and a rectangle. They partition the area into n rectangles, one per input number. The rectangles are guaranteed to fill the input rectangle, and each rectangle is proportional in area to a number on the input list. Treemaps are designed to be applied hierarchically, so any given resulting rectangle can itself contain a treemap, and so on, recursively.

In order to build PhotoMesa, I had to extend the treemap algorithms to accommodate fixed size images. To understand this, let us start by looking at existing treemap algorithms.

There are two desirable properties that treemap algorithms can have: generated rectangles with aspect ratios close to 1 (i.e., rectangles that are close to squares), and order.

Here, and for the rest of the paper, aspect ratio is defined as $\max((\text{width} / \text{height}), (\text{height} / \text{width}))$, so that an aspect ratio of 1 is perfectly square, and aspect ratios larger than one are more rectangular. Rectangles with aspect ratios close to 1 are desirable because, generally speaking, they are more visually attractive. In addition, humans seem to be able to estimate the area of a square more accurately than a skinny rectangle, and one of the goals of treemaps is to use the area of each rectangle to present some useful attribute.

I define order here to mean that a treemap algorithm is ordered if the rectangles it generates are laid out in a spatial sequence that corresponds to the input sequence. Not all treemap algorithms are ordered, and order is important since it is easier for users to find specific items in ordered displays. Rodden has showed the importance of order in image browsing [19]. In addition, ordered displays make it easier to track items if they change over time since in an ordered display, each item will stay in approximately the same place on the screen.

Until recently, there were no algorithms that provided both properties.

Treemap Related Work

The original treemap algorithm by Shneiderman [21] uses a simple “slice and dice” approach. It divides the input rectangle into a single horizontal or vertical list of rectangles – each one typically being quite skinny. If the algorithm is applied recursively, the sub-rectangle would be split in the opposite orientation as the parent. This algorithm generates ordered rectangles, but they typically have extreme aspect ratios.

An important ensuing treemap algorithm, called squarified treemaps, gave up on ordering, but created rectangles with smaller aspect ratios [10]. Squarified treemaps work by recursively dividing the space in two, and laying out some of the rectangle in one part, and the rest of the rectangles in the other part, where the list of rectangles is split based on optimizing the resulting aspect ratios. A variation of this algorithm was independently developed for SmartMoney’s MarketMap applet [4]. Recently, Shneiderman and Wattenberg introduced ordered treemaps [5] which offer a compromise solution where the resulting rectangles are ordered, and somewhat squarified, but do not have as good aspect ratios as those generated by squarified treemaps. Other approaches to space-filling algorithms have been considered but they typically do not have all the nice properties of treemaps, such as that by Harel and Yashchin

[13] which does not assign the size of the rectangles to any independent variable.

Treemaps have been applied to a number of domains, from visualizing hard disk usage [3] to the stock market [4]. However, in every current usage of treemaps to date, they are used to visualize a two-dimensional dataset where typically, one dimension is mapped to the area of the rectangles (as computed by the treemap algorithm), and the other dimension is mapped to the color of the rectangle. Then, a label is placed in the rectangles which are large enough to accommodate them, and the user can interact with the treemap to get more information about the objects depicted by the rectangles.

Surprisingly enough, there are not any published uses of treemaps where other information is placed in the rectangles. PhotoMesa appears to be the first application to put images within the area of each treemap rectangle.

There is a good reason why treemaps have not been used in this manner before. This is because while treemaps guarantee that the area of each generated rectangle is proportional to an input number, they do not make any promise about the aspect ratio of the rectangles. Some treemap algorithms (such as squarified treemaps) do generate rectangles with better aspect ratios, but the rectangles can have any aspect ratio. While this is fine for general purpose visualizations, it is not appropriate for laying out images because images have fixed aspect ratios, and they do not fit well in rectangles with inappropriate aspect ratios.

Let us look at applying existing treemap algorithms to laying out fixed size objects, such as images. For now, let us assume without loss of generality that the images are all square. We will see later that this does not affect layout issues. Given a list of groups of images to lay out, the obvious input to the treemap algorithm is the number of images in each group. The treemap algorithm will generate a list of rectangles, that each need the corresponding images to be laid out within.

For each rectangle and group of images, the first step is to decide on the dimensions of a grid with which to lay out the images in the rectangle. Given the aspect ratio of the rectangle, we compute the number of rows and columns that best fit the images.

The resulting grid may have more cells than there are images, but will not have any empty rows or columns. This layout, however, is not guaranteed to fit in the rectangle. For example, consider a rectangle that was computed to hold a single image. It will have an area of 1.0, but could be long and skinny, perhaps with a width of 10.0 and a height of 0.1. The obvious solution is to scale down the images just enough to fit in the bounds of the rectangle.

Herein lies the problem. Since each group of images has to fit in to a separate rectangle, each group of images will have to potentially be scaled down. This will result in each group of images being a different size. Furthermore, since

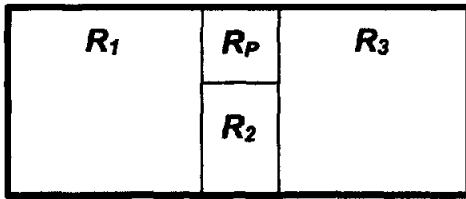


Figure 3: Basic layout strategy of the ordered treemap algorithm. The pivot is layed out in R_p , and $L_1...L_{p-1}$ are layed out in R_1 while $L_{p+1}...L_n$ are layed out in R_2 and R_3 .

the rectangles are arbitrarily sized and positioned, and the images are scaled, the resulting groups of images will not align with each other in a visually attractive way.

It is standard graphic design practice to align content in a way that makes it easy for the eye to quickly scan different areas. If each group of images is a different size and they are not aligned, this will make the resulting layout less attractive, and may make it slower for a user to quickly scan. See Figure 4 for the result of laying out a simple sequence of images using the ordered treemap and quantum treemap algorithms to see the difference in overall layout. Note how with the ordered treemap, group #4 consisting of a single image is scaled much smaller than the other images. With the quantum treemap algorithm, all images are the same size, and all images are aligned on a single grid across all the groups.

Ordered Treemaps

To understand the quantum treemap algorithm, it is necessary to first understand the basics of the ordered treemap algorithm because the former is a direct modification of the latter.

The ordered treemap algorithm, as with all treemap algorithms, take as input and produces output:

Input	$L_1...L_n$	An ordered sequence of numbers.
	Box	A box to lay out the rectangles within.
Output	$R_1...R_n$	An ordered sequence of rectangles that completely fill Box , and where the area of R_i is proportional to L_i .

The algorithm is similar to QuickSort. It chooses a pivot, L_p , and places it in Box . It then recursively lays out $L_1...L_{p-1}$ on one side of the pivot, and $L_{p+1}...L_n$ on the other side of the pivot. Figure 3 shows the basic visual strategy for a horizontal layout. A corresponding approach is used for a vertical layout.

The ordered treemap algorithm is described in detail in [5], and is summarized here.

1. If $n = 1$, then return a rectangle $R = Box$ and stop.
2. Choose a pivot element, L_p . Pivot selection strategies include picking the middle element or the largest one.
3. Calculate R_1 so that its height fills Box , and so that its width is large enough to contain $L_A = L_1...L_{p-1}$.

4. Split $L_{p+1}...L_n$ into two sublists, L_B and L_C that will be laid out in R_2 and R_3 . Calculate where the splitting point is so that R_p has an aspect ratio closest to 1.
5. Calculate R_p , R_2 and R_3 . This is performed by using the ratio between the size of the corresponding lists, and breaking up the available space by the same ratios.
6. Recursively apply the ordered treemap algorithm to L_A in R_1 , L_B in R_2 , and L_C in R_3 .

This algorithm results in rectangles that are fairly square, and are ordered approximately left to right (or top to bottom in a vertically oriented box).

Quantum Treemap Algorithm

The goal of the quantum treemap algorithm is similar to other treemap algorithms, but instead of generating rectangles of arbitrary aspect ratios, it generates rectangles with widths and heights that are integer multiples of a given elemental size. In this manner, it always generates rectangles in which a grid of elements of the same size can be laid out. Furthermore, all the grids of elements will align perfectly with rows and columns of elements running across the entire series of rectangles. It is this basic element size that can not be made any smaller that led to the name of *quantum treemaps*.

The quantum treemap (QT) algorithm is based directly on the ordered treemap (OT) algorithm. However, the basic approach could be applied to any other treemap algorithm. QT's input and output are similar to those of OT, but instead of returning a set of rectangles that precisely fill the specified input Box , it generates a set of rectangles that only approximate the input Box . Because there is some wasted space, the resulting set of rectangles are usually larger than Box , but have close to the same aspect ratio. In addition, QT takes an additional input parameter which is the aspect ratio of the elements to be laid out in Box .

QT starts in exactly the same manner as OT, picking a pivot, subdividing the space, and recursively applying the algorithm to each sub-space. It works in the same way until step 1 stops the recursion.

At this point (step 1), rather than just unwinding the recursive stack, it adjusts the computed rectangle by modifying its dimensions, making it big enough for precisely the specified number of elements.

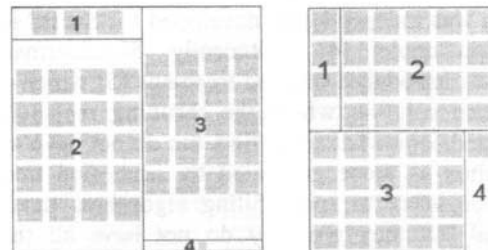


Figure 4: The result of laying out a sequence of 4 groups of elements (of size 3, 20, 1) using ordered treemap (left) and quantum treemap (right)

Then, as the recursion unwinds, the caller must accommodate the generated rectangles which may not fit precisely into the box that was asked for. This is the tricky part, and is captured in a modified version of step 6. Since the rectangles generated by the recursive call may be bigger or smaller in either dimension than was asked for, the rectangles from the other regions must be moved so they don't overlap, and possibly grown so they align nicely with neighboring rectangles. As an example, see Figure 4 (right). Rectangle #4 was originally computed to have dimensions (1x1), but since Rectangle #3 was much taller, Rectangle #4 was stretched to be 4 units tall to match the height of Rectangle #3. Similarly, Rectangle #1 was stretched to match the height of Rectangle #2. The new algorithmic steps are stated here:

- new 1.** If $n = 1$, then compute a rectangle R that contains exactly L quantums in a grid arrangement that has an aspect ratio as close as possible to that of Box and stop.
- new 6.** Recursively apply the ordered treemap algorithm to L_A in R_1 , L_B in R_2 , and L_C in R_3 .
- new 6a.** Translate the rectangles in R_P , R_2 , and R_3 to avoid overlapping R_1 or each other.
- new 6b.** Even out the rectangles in the sub-regions in the following manner. Make sure that R_P and R_2 have the same width. Make sure that R_P and R_2 together have the same height as R_1 . Make sure that R_3 has the same height as R_1 . Each of these evening steps can be accomplished similarly by finding if one of the regions is too small. Then if it is not wide enough, add the extra amount to the width of the rectangles in that region that touch the right boundary of the region. Do the analogous action to rectangles not tall enough.

Element Aspect Ratio Issues

QT assumes that all elements that will be laid out in the rectangles produced by QT are the same aspect ratio, and that aspect ratio is an input parameter to QT. It turns out, however, that it is not necessary to modify the internal structure of QT to accommodate the element's aspect ratio. Instead, the dimensions of the starting box can simply be stretched by the inverse of the element aspect ratio.

Growing Horizontally or Vertically

In step 1, the requested rectangle may be grown to accommodate the quantum element size. There is a basic question of whether to grow this rectangle horizontally or vertically. The simple answer is just to grow in the direction that results in a rectangle that most closely matches the aspect ratio of the original rectangle. However, the algorithm as a whole produces better layouts if it always grows horizontally (or vertically for layout boxes that are oriented vertically).

The issue here is somewhat subtle, but is related to step 6b where the rectangles are evened. If, for example, rectangles in R_3 are made taller, than all of R_1 and R_2 will

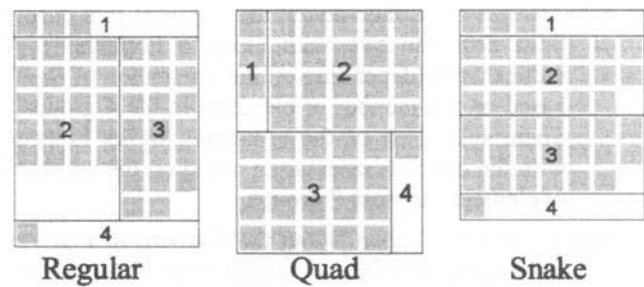


Figure 5: The result of applying the three stopping conditions to a sequence of 4 groups of elements (of size 3, 20, 20, 1).

have to made taller as well to match R_3 . If instead, the rectangles in R_3 are made wider, than only the other rectangles in R_3 will need to be made wider, and the rectangles in R_1 and R_2 can be left alone.

In general, the evening aspect of the QT algorithm remains somewhat problematic. While it works well for most data sets, it occasionally yields undesirable layouts due to too much wasted space. This can happen when one region ends up growing a fair amount to accommodate data that doesn't happen to fit the starting rectangles, and then the other regions have to be grown to match. When these other regions are grown to match, the resulting rectangles are bigger than necessary, and there is wasted space. This doesn't seem to be a problem for datasets unless they contain many regions with a very small number of elements (< 10). In practice, it has not been a significant problem for the real image datasets I have viewed, although sometimes there is a little more wasted space than I would like.

Stopping Condition Improvements

Changing the stopping conditions and offering special layouts for a small number of special cases can produce substantially better total results. The new stopping conditions apply equally to QT as well as to OT.

The improvement is because the layout of rectangles depicted in Figure 5 (left) is not necessarily the one with the smallest aspect ratios. In addition, it generates a layout that is somewhat difficult to parse visually because the eye has to move in 3 directions to focus on the 4 rectangles (vertically from #1 to #2, horizontally from #2 to #3, and then vertically from #3 to #4).

The layout can be improved, and visual readability by offering two alternative layouts. The first produces a "quad" of (2x2) rectangles. The second produces a "snake" layout with all 4 rectangles laid out sequentially – either horizontally or vertically. The snake layout can be equally well applied to 2, 3, or more rectangles. PhotoMesa applied it up to 5 rectangles. Figure 5 shows the result of laying out a sequence of 4 groups of elements using the three strategies. The new algorithmic step is:

- new 1a.** If $n = 4$, then first try the regular layout by continuing and letting the recursion get down to the bottom level

new 1b. If $n = 4$, then layout the 4 groups in a quad. Split *Box* into two with either a horizontal or vertical split (depending on the orientation of *Box*) based on the number of elements in the 4 groups. Then, split each of the remaining boxes in two with the opposite orientation based on the number of elements in those 2 groups.

new 1c. If $n = 4$, then layout the 4 groups in a snake by dividing *Box* into 4 sub boxes (horizontally or vertically, depending on the orientation of *Box*), based on the number of elements in the 4 groups.

new 1d. Compute the aspect ratios and wasted space of the 4 resulting rectangles from steps 1a, 1b, and 1c, and use the layout with the best overall results.

Since no one layout strategy always gives the best result for all input data, for 5 or fewer rectangles, PhotoMesa computes layouts using all strategies (original, quad, and snake) and picks the best one. In practice, this strategy produces layouts with substantially squarer aspect ratios. Running 100 randomized tests with 100 rectangles, and random numbers of elements per rectangle, ranging from 10 to 1000 produced an average aspect ratio of 3.92 with the original stopping conditions, and 2.68 for the new stopping conditions.

Pivot Selection Improvement

In addition to the two pivot selection strategies discussed in [5], a third strategy specifically targeted at the evening problem previously discussed offers improved results. The new strategy, called “*split size*” gives better results for some input data.

The basic approach is to pick the pivot that will split the lists of elements into equal sizes, or as close to equal as possible. With the sublists containing similar numbers of elements to lay out, there tends to be less evening, and therefore less wasted space. The new algorithmic step is:

new 2. Choose a pivot element, R_p . Pivot selection strategies include picking the middle element, the largest one, or the one that results in splitting the elements into lists that are as close to equal size as possible.

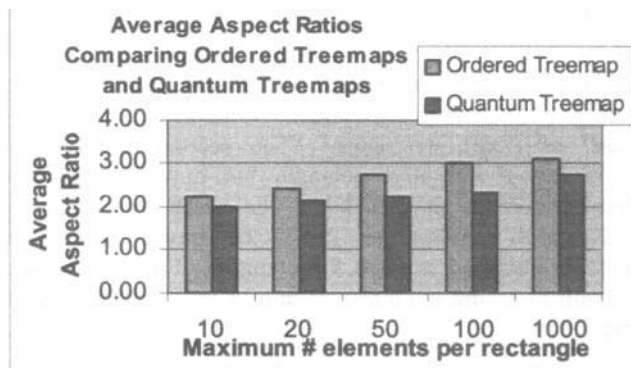


Figure 6: Average aspect ratio of all rectangles run on both algorithms with 100 rectangles with random numbers of elements per rectangle.

No single pivot selection strategy always works best, so in practice, PhotoMesa computes the layouts with all three pivot selection strategies, and picks the best one based on the average aspect ratios of the resulting rectangles, and the amount of wasted space.

Quantum Treemap Analysis

One of the basic characteristics of QT is that it works better when there are more elements per group. This is because it gives the algorithm more flexibility when computing rectangles. A rectangle of, say, 1000 elements, can be arranged in quantified grids of many different sizes such as (30x34), (31x33), (32x32), etc. – each of which use the space quite efficiently. Rectangles containing smaller numbers of elements, however, do not offer as many options, and often use space less efficiently. For example, a rectangle containing 5 elements can be laid out in (1x5), (2x3), (3x2), or (5x1). These four options do not give the algorithm as much flexibility as the dozens of grid options afforded by the larger number of elements.

In order to assess the effectiveness of QT, it was compared to OT with a series of trials using random input. Each algorithm was run 100 times generating 100 rectangles with the number of elements in each rectangle being randomly generated. This was done for 5 different ranges of the number of elements per rectangle. For each test, the average aspect ratio of all the rectangles was recorded as well as the space utilization, which was recorded as the percentage of space not used to display elements (wasted space). The same random numbers were used for each algorithm. Figures 6 and 7 show the results of these tests. Quantum treemaps did better in terms of aspect ratio, and ordered treemaps did better in terms of wasted space.

However, the crucial visual advantage of QT is that it always produces layouts where elements are the same size and are aligned on a single global grid.

BUBBLE MAPS

While quantum treemaps work well, they waste some space. While the use of some white space can be helpful to visually distinguish the groups, too much white space

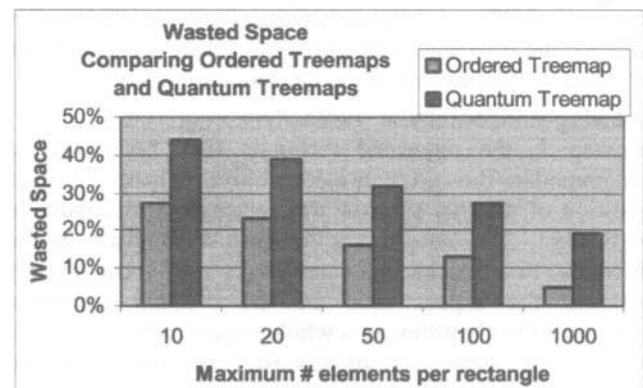


Figure 7: Average wasted space as a percentage of the entire layout space available. Tests run on both algorithms with 100 rectangles with random numbers of elements per rectangle.

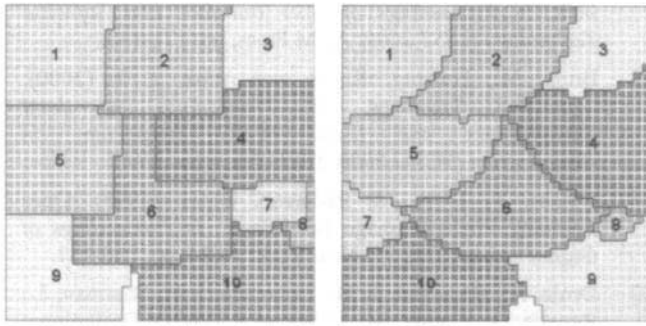


Figure 8: Bubblemap layouts of 10 groups of up to 200 rectangles. Rectangular (left) and circular (right).

becomes unattractive and wasteful. While it may be possible to improve the quantum treemap algorithm, it is impossible to lay out images in a rectangle without sometimes leaving unused space. An alternative approach is to give up on the idea that the space must be divided into rectangles, and instead allow more complex shapes.

Bubblemap is a new algorithm that lays out groups of quantum-sized objects in an ordered layout with no wasted space per group, although there is some wasted space for the entire area. The groups of objects can be created in different shapes, such as rectangular or circular, but the groups of objects only approximate those shapes, rather than define them exactly. Figure 8 shows a rectangular and a circular bubblemap layout of 10 groups of up to 200 rectangles per group. The bubblemap algorithm has also been integrated into PhotoMesa as a user-selectable layout option. Figure 9 shows the bubblemap algorithm applied to a set of images in PhotoMesa. There is no wasted space, but the regions have arbitrary shapes.

A more sophisticated approach to laying out related images in a grid has been pursued by Basalaj with his *Proximity Grid* algorithm [7]. It takes a set of objects with a high-dimensional set of relationships and generates a grid layout of those objects so that similar objects will be near each other on the grid. Bubblemaps, on the other hand, are much simpler and assumes the input is pre-clustered. They keep the clusters of images together, rather than optimizing an n -dimensional set of relationships.

The bubblemap algorithm is completely different than the treemap algorithm. Rather than subdividing rectangles, it is based on a standard pixel-based bucket fill algorithm. It works by filling cells in a grid, keeping track of which cells get assigned to images from which group. It fills the cells one group at a time. By using different algorithms to select the next cell to fill, the shape of the groups can be controlled. The basic algorithm runs in $O(n)$ time for n images. The basic algorithm follows:

Input: $L_1...L_n$, Aspect Ratio

1. Compute the size of the overall grid based on total number of images to layout, and the desired resulting aspect ratio.
2. Create a grid of size computed from step 1, and set each cell to the value UNASSIGNED.

3. For each group of images, L_i , call the fill algorithm, starting at step 4, and then stop.
4. Find the starting point to fill by looking for the first UNASSIGNED cell in the grid (in left-right, top-bottom order). Initialize a list of cells, called LIST, and add the starting point to LIST.
5. If LIST is empty then stop. Else, take the first element, P, off of LIST, and set the cell at P's location to the value ASSIGNED.
6. For each UNASSIGNED neighbor of P, Q, add Q to LIST, and set the cell at Q's location to the current group ID. GOTO 5. Note that the order in which the neighbors are added to LIST affect the shape of the resulting groups.

Bubblemaps are easy to implement, use space efficiently, and give ordered space-filling layouts like treemaps do. However, since they produce arbitrary-shaped regions rather than rectangles, it is likely that users will find them harder to parse visually. So, while they may be appropriate for applications where the goal is simply to have related items near each other, they may not be as appropriate for tasks where users need to distinguish clearly between groups.

FUTURE WORK

The PhotoMesa application has a long list of features to be added. An important one is to integrate it with PhotoFinder and other sources of metadata. Another area to look into is deploying PhotoMesa in different modalities, such as stand-alone CDs for sending to family and friends, running on a kiosk for museums or lobbies, and integration with the Web. The zoomable characteristics of PhotoMesa make it a good match with the Web, offering a potentially efficient manner to browse large image databases on the Web since only the resolution needed for the current view needs to be sent to the client. And of course, a detailed user study is needed to understand how the zoomable interaction of PhotoMesa compares to more traditional approaches, and which layout techniques are most effective.

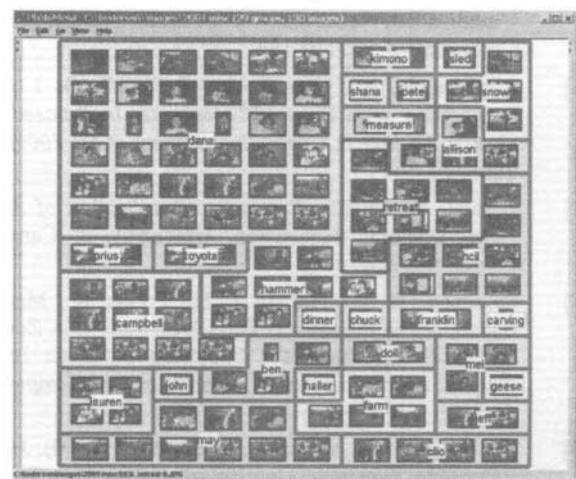


Figure 9: The same images as in Figure 3, but laid out with a rectangular bubblemap.

CONCLUSION

This paper presents PhotoMesa, a zoomable image browser, and two new algorithms for laying out groups of images or other fixed-size visual objects. The primary innovations are: 1) a simplified set of interactions for navigating through a set of objects in a zoomable user interface; and 2) algorithms to lay out fixed-size objects, such as images, in two-dimensional space, automatically creating groups for related objects.

By bringing together the aforementioned innovations with existing zoomable user interface technology, PhotoMesa offers a significant advance in the ability to comfortably browse large numbers of images. Based on its initial popularity and enthusiastic feedback, PhotoMesa appears to have satisfied its initial design goals of being simple to use in a family setting, requiring no setup time, and naturally supporting co-present use.

ACKNOWLEDGEMENTS

I appreciate the feedback on PhotoMesa by many HCIL members. In particular, thanks to Jesse Grosjean who suggested the approach taken in the bubblemap algorithm. In addition, I thank Ben Shneiderman for suggesting the interactive textual list of groups, to Allison Druin for suggesting the ability to color groups, to Jon Meyer and Catherine Plaisant for advise on the visual design of PhotoMesa, to Matthias Mayer for first suggesting I try to display several directories of images at once, and to Mark Stefik from Xerox PARC for suggesting the magnified preview images. Finally, I appreciate Susanne Jul's excellent editorial comments on this paper.

REFERENCES

- [1] ACDSee (2001). <http://www.acdsystems.com/english/products/acdsee/acdsee-node.htm>.
- [2] Canon ZoomBrowser (2001). http://www.powershot.com/powershot2/software/ps_pc_view.html.
- [3] DiskMapper (2001). <http://www.miclog.com/dmdesc.htm>.
- [4] SmartMoney MarketMap (2001). <http://www.smartmoney.com/marketmap/>.
- [5] Baker, M. J., & Eick, S. G. (1995). Space-Filling Software Visualization. *Journal of Visual Languages and Computing*, 6, pp. 119-133.
- [6] Balabanovic, M., Chu, L. L., & Wolff, G. J. (2000). Storytelling With Digital Photographs. In *Proceedings of Human Factors in Computing Systems (CHI 2000)* ACM Press, pp. 564-571.
- [7] Basalaj, W. (2000). *Proximity Visualization of Abstract Data*. Doctoral dissertation, University of Cambridge, Cambridge, England.
- [8] Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, pp. 3-31.
- [9] Bederson, B. B., Meyer, J., & Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proceedings of User Interface and Software Technology (UIST 2000)* ACM Press, pp. 171-180.
- [10] Bruls, M., Huizing, K., & van Wijk, J. J. (2000). Squarified Treemaps. In *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000)* IEEE Press, pp. 33-42.
- [11] Combs, T. T. A., & Bederson, B. B. (1999). Does Zooming Improve Image Browsing? In *Proceedings of Digital Library (DL 99)* New York: ACM, pp. 130-137.
- [12] Druin, A., Bederson, B. B., Hourcade, J. P., Sherman, L., Revelle, G., Platner, M., & Weng, S. (2001). Designing a Digital Library for Young Children: An Intergenerational Partnership. In *Proceedings of Joint Conference on Digital Libraries (JCDL 2001)* ACM Press, pp. 398-405.
- [13] Harel, D., & Yashchin, G. (2000). An Algorithm for Blob Hierarchy Layout. In *Proceedings of Advanced Visual Interfaces (AVI 2000)* ACM Press, pp. 29-40.
- [14] Hascoët-Zizi, M., & Pediotakis, N. (1996). Visual Relevance Analysis. In *Proceedings of International Conference on Digital Libraries (DL 96)* ACM Press, pp. 54-62.
- [15] Jul, S., & Furnas, G. W. (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. In *Proceedings of User Interface and Software Technology (UIST 98)* ACM Press, pp. 97-106.
- [16] Kang, H., & Shneiderman, B. (2000). Visualization Methods for Personal Photo Collections Browsing and Searching in the PhotoFinder. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2000)* New York: IEEE, pp. 1539-1542.
- [17] Platt, J. (2000). AutoAlbum: Clustering Digital Photographs Using Probabilistic Model Merging. In *Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL-2000)* IEEE Press,
- [18] Robertson, G. G., & Mackinlay, J. D. (1993). The Document Lens. In *Proceedings of User Interface and Software Technology (UIST 93)* ACM Press, pp. 101-108.
- [19] Rodden, K., Basalaj, W., Sinclair, D., & Wood, K. (2001). Does Organisation by Similarity Assist Image Browsing. In *Proceedings of Human Factors in Computing Systems (CHI 2001)* ACM Press, pp. 190-197.
- [20] Shen, C., Moghaddam, B., Lesh, N., & Beardsley, P. (2001). Personal Digital Historian: User Interface Design. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2001)* ACM Press,
- [21] Shneiderman, B. (1992). Tree Visualization With Treemaps: A 2-D Space-Filling Approach. *ACM Transactions on Graphics*, 11(1), pp. 92-99.
- [22] Shneiderman, B., & Kang, H. (2000). Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos. In *Proceedings of IEEE Conference on Information Visualization (IV2000)* New York: IEEE, pp. 88-98.
- [23] Stewart, J., Bederson, B. B., & Druin, A. (1999). Single Display Groupware: A Model for Co-Present Collaboration. In *Proceedings of Human Factors in Computing Systems (CHI 99)* ACM Press, pp. 286-293.

Automatic Thumbnail Cropping and its Effectiveness

Bongwon Suh*, Haibin Ling, Benjamin B. Bederson*, David W. Jacobs

Department of Computer Science

*Human-Computer Interaction Laboratory

University of Maryland

College Park, MD 20742 USA

+1 301-405-2764

{sbw, hbling, bederson, djacobs}@cs.umd.edu

ABSTRACT

Thumbnail images provide users of image retrieval and browsing systems with a method for quickly scanning large numbers of images. Recognizing the objects in an image is important in many retrieval tasks, but thumbnails generated by shrinking the original image often render objects illegible. We study the ability of computer vision systems to detect key components of images so that automated cropping, prior to shrinking, can render objects more recognizable. We evaluate automatic cropping techniques 1) based on a general method that detects salient portions of images, and 2) based on automatic face detection. Our user study shows that these methods result in small thumbnails that are substantially more recognizable and easier to find in the context of visual search.

Keywords

Saliency map, thumbnail, image cropping, face detection, usability study, visual search, zoomable user interfaces

INTRODUCTION

Thumbnail images are now widely used for visualizing large numbers of images given limited screen real estate. The QBIC system developed by Flickner *et al.* [10] is a notable image database example. A zoomable image browser, PhotoMesa [3], lays out thumbnails in a zoomable space and lets users move through the space of images with a simple set of navigation functions. PhotoFinder applied thumbnails as a visualization method for personal photo collections [14]. Popular commercial products such as Adobe Photoshop Album [2] and ACDSSee [1] also use thumbnails to represent image files in their interfaces.

Current systems generate thumbnails by shrinking the original image. This method is simple. However, thumbnails generated this way can be difficult to recognize,

especially when the thumbnails are very small. This phenomenon is not unexpected, since shrinking an image causes detailed information to be lost. An intuitive solution is to keep the more informative part of the image and cut less informative regions before shrinking. Some commercial products allow users to manually crop and shrink images [20]. Burton *et al.* [4] proposed and compared several image simplification methods to enhance the full-size images before subsampling. They chose edge-detecting smoothing, lossy image compression, and self-organizing feature map as three different techniques in their work.

In quite a different context, DeCarlo and Santella [8] tracked a user's eye movements to determine interesting portions of images, and generated non-photorealistic, painterly images that enhanced the most salient parts of the image. Chen *et al.* [5] use a visual attention model as a cue to conduct image adaptation for small displays.

In this paper, we study the effectiveness of saliency based cropping methods for preserving the recognizability of important objects in thumbnails. Our first method is a general cropping method based on the saliency map of Itti and Koch that models human visual attention [12][13]. A saliency map of a given image describes the importance of each position in the image. In our method, we use the saliency map directly as an indication of how much information each position in images contains. The merit of this method is that the saliency map is built up from low-level features only, so it can be applied to general images. We then select the most informative portion of the image.

Although this saliency based method is useful, it does not consider semantic information in images. We show that semantic information can be used to further improve thumbnail cropping, using automatic face detection. We choose this domain because a great many pictures of interest show human faces, and also because face detection methods have begun to achieve high accuracy and efficiency [22].

In this paper we describe saliency based cropping and face detection based cropping after first discussing related work from the field of visual attention. We then explain the

design of a user study that evaluates the thumbnail methods. This paper concludes with a discussion of our findings and future work.

RELATED WORK

Visual attention is the ability of biological visual systems to detect interesting parts of the visual input [12][13][16][17][21]. The saliency map of an image describes the degree of saliency of each position in the image. The saliency map is a matrix corresponding to the input image that describes the degree of saliency of each position in the input image.

Itti and Koch [12][13] provided an approach to compute a saliency map for images. Their method first uses pyramid technology to compute three feature maps for three low level features: color, intensity, and orientation. For each feature, saliency is detected when a portion of an image differs in that feature from neighboring regions. Then these feature maps are combined together to form a single saliency map. After this, in a series of iterations, salient pixels suppress the saliency of their neighbors, to concentrate saliency in a few key points.

Chen *et al.* [5] proposed using semantic models together with the saliency model of Itti and Koch to identify important portions of an image, prior to cropping. Their method is based on an attention model that uses attention objects as the basic elements. The overall attention value of each attention object is calculated by combining attention values from different models. For semantic attention models they use a face detection technique [15] and a text detection technique [6] to compute two different attention values. The method provides a way to combine semantic information with low-level features. However, when combining the different values, their method uses heuristic weights that are different for five different predefined image types. Images need to be manually categorized into these five categories prior to applying their method. Furthermore, it heavily relies on semantic extraction techniques. When the corresponding semantic technique is not available or when the technique fails to provide a good result (e.g. no face found in the image), it is hard to expect a good result from the method. On the other hand, our algorithm is totally automatic and works well without manual intervention or any assumptions about the image types.

THUMBNAIL CROPPING

Problem Definition

We define the thumbnail cropping problem as follows: Given an image I , the goal of thumbnail cropping is to find a rectangle R_C , containing a subset of the image I_C so that the main objects in the image are visible in the subimage. We then shrink I_C to a thumbnail. In the rest of this paper, we use the word “cropping” to indicate thumbnail cropping.

In the next subsection, we propose a general cropping method, which is based on the saliency map and can be applied to general images. Next, a face detection based cropping method is introduced for images with faces.

A General Cropping Method Based on the Saliency Map

In this method, we use the saliency value to evaluate the degree of informativeness of different positions in the image I . The cropping rectangle R_C should satisfy two conditions: having a small size and containing most of the salient parts of the image. These two conditions generally conflict with each other. Our goal is to find the optimal rectangle to balance these two conditions.

An example saliency map is given in Figure 1:



Figure 1: left: original image, right: saliency map of the image shown left

Find Cropping Rectangle with Fixed Threshold using Brute Force Algorithm

We use Itti and Koch’s saliency algorithm because their method is based on low-level features and hence independent of semantic information in images. We choose Itti and Koch’s model also because it is one of the most practical algorithms on real images.

Once the saliency map S_I is ready, our goal is to find the crop rectangle R_C that is expected to contain the most informative part of the image. Since the saliency map is used as the criteria of importance, the sum of saliency within R_C should contain most of the saliency value in S_I . Based on this idea, we can find R_C as the smallest rectangle containing a fixed fraction of saliency. To illustrate this formally, we define candidates set $\mathfrak{R}(\lambda)$ for R_C and the fraction threshold λ as

$$\mathfrak{R}(\lambda) = \left\{ r : \frac{\sum_{(x,y) \in r} S_I(x,y)}{\sum_{(x,y)} S_I(x,y)} > \lambda \right\}$$

Then R_C is given by

$$R_C = \arg \min_{r \in \mathfrak{R}(\lambda)} (\text{area}(r))$$

R_C denotes the minimum rectangle that satisfies the threshold defined above. A brute force algorithm was developed to compute R_C .

Find Cropping Rectangle with Fixed Threshold using Greedy Algorithm

The brute force method works, however, it is not time efficient. Two main factors slow down the computation. First, the algorithm to compute the saliency map involves several series of iterations. Some of the iterations involve convolutions using very large filter templates (on the order of the size of the saliency map). These convolutions make the computation very time consuming.

Second, the brute force algorithm basically searches all sub-rectangles exhaustively. While techniques exist to speed up this exhaustive search, it still takes a lot of time.

We found that we can achieve basically the same results much more efficiently by: 1) using fewer iterations and smaller filter templates during the saliency map calculation; 2) squaring the saliency to enhance it; 3) using a greedy search instead of brute force method by only considering rectangles that include the peaks of the saliency.

```

Rectangle GREEDY_CROPPING ( $S, \lambda$ )
  thresholdSum  $\leftarrow \lambda * \text{Total saliency value in } S$ 
   $R_c \leftarrow \text{the center of } S$ 
  currentSaliencySum  $\leftarrow \text{saliency value of } R_c$ 
  WHILE currentSaliencySum < thresholdSum DO
     $P \leftarrow \text{Maximum saliency point outside } R_c$ 
     $R' \leftarrow \text{Small rectangle centered at } P$ 
     $R_c \leftarrow \text{UNION}(R_c, R')$ 
    UPDATE currentSaliencySum with new region  $R_c$ 
  ENDWHILE
  RETURN  $R_c$ 

```

Figure 2: Algorithm to find cropping rectangle with fixed saliency threshold. S is the input saliency map and λ is the threshold.

Figure 2 shows the algorithm GREEDY_CROPPING to find the cropping rectangle with fixed saliency threshold λ . The greedy algorithm calculates R_c by incrementally including the next most salient peak point P . Also when including a salient point P in R_c , we union R_c with a small rectangle centered at P . This is because if P is within the foreground object, it is expected that a small region surrounding P would also contain the object.

This algorithm can be modified to satisfy further requirements. For example, the UNION function in Figure 2 can be altered when the cropped rectangle should have the same aspect ratio as the original image. Rather than just merging two rectangles, UNION needs to calculate the minimum surrounding bounds that have the same aspect ratio as the original image. As another example, the initial value of R_c can be set to either the center of image, S , or the most salient point or any other point. Since the initial point always falls in the result thumbnail, it can be regarded as a point with extremely large saliency. When the most salient point is selected as an initial point, the

result can be optimized to have the minimum size. But, we found that to begin the algorithm with the center of images gives more robust and faster results even though it might increase the size of the result thumbnail especially when all salient points are skewed to one side of an image.

Find Cropping Rectangle with Dynamic Threshold

Experience shows that the most effective threshold varies from image to image. We therefore have developed a method for adaptively determining the threshold λ .

Intuitively, we want to choose a threshold at a point of diminishing returns, where adding small amounts of additional saliency requires a large increase in the rectangle. We use an area-threshold graph to visualize this. The X axis indicates the threshold (fraction of saliency) while the Y axis shows the normalized area of the cropping rectangle as the result of the greedy algorithm mentioned above. Here the normalized area has a value between 0 and 1. The solid curve in Figure 3 gives an example of an area-threshold graph.

A natural solution is to use the threshold with maximum gradient in the area-threshold graph. We approximate this using a binary search method to find the threshold in three steps: First, we calculate the area-threshold graph for the given image. Second, we use a binary search method to find the threshold where the graph goes up quickly. Third, the threshold is tuned back to the position where a local maximum gradient exists. The dotted lines in Figure 3 demonstrate the process of finding the threshold for the image given in Figure 1.

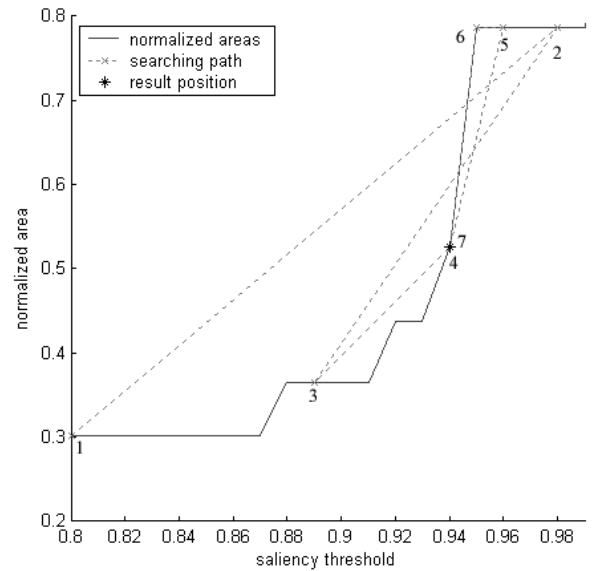


Figure 3: The solid line represents the area-threshold graph. The dotted lines show the process of searching for the best threshold. The numbers indicate the sequence of searching

Examples of Saliency Map Based Cropping

After getting R_C , we can directly crop the input image I . Thumbnails of the image given in Figure 1 are shown in Figure 4. It is clear from Figure 4 that the cropped thumbnail can be more easily recognized than the thumbnail without cropping.



Figure 4 (left): the image cropped based on the saliency map; (middle): the cropping rectangle which contains most of the saliency parts; (right top): a thumbnail subsampled from the original image; (right bottom): a thumbnail subsampled from the cropped image (left part of this figure).

Figure 5 shows the result of an image whose salient parts are more scattered. Photos focusing primarily on the subject and without much background information often have this property. A merit of our algorithm is that it is not sensitive to this.

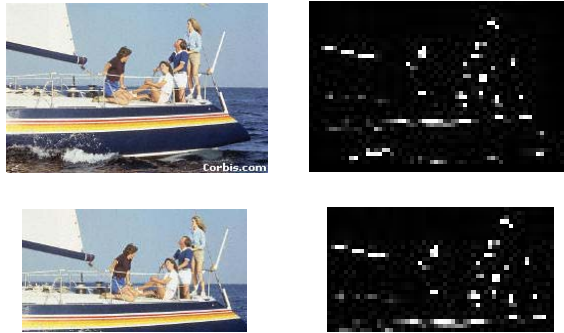


Figure 5 (left top): the original image (courtesy of Corbis [7]); (right top): the saliency map; (left bottom): the cropped image; (right bottom): the cropped saliency map which contains most of the salient parts.

Face Detection Based Cropping

In the above section, we proposed a general method for thumbnail cropping. The method relies only on low-level features. However, if our goal is to make the objects of interest in an image more recognizable, we can clearly do this more effectively when we are able to automatically detect the position of these objects.

Images of people are essential in a lot of research and application areas. At the same time, face processing is a rapidly expanding area and has attracted a lot of research effort in recent years. Face detection is one of the most important problems in the area. [22] surveys the numerous methods proposed for face detection.

For human image thumbnails, we claim that recognizability will increase if we crop the image to contain only the face

region. Based on this claim, we designed a thumbnail cropping approach based on face detection. First, we identify faces by applying CMU's on-line face detection [9][19] to the given images. Then, the cropping rectangle R_C is computed as containing all the detected faces. After that, the thumbnail is generated from the image cropped from the original image by R_C .

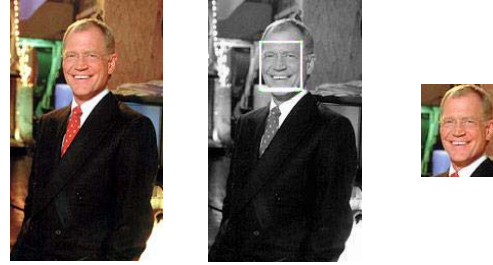


Figure 6 (left): the original image; (middle): the face detection result from CMU's online face detection [9]; (right): the cropped image based on the face detection result.

Figure 6 shows an example image, its face detection result and the cropped image. Figure 7 shows the three thumbnails generated via three different methods. In this example, we can see that face detection based cropping method is a very effective way to create thumbnails, while saliency based cropping produces little improvement because the original image has few non-salient regions to cut.



Figure 7: Thumbnails generated by the three different methods. (left): without cropping; (middle): saliency based cropping; (right): face detection based cropping.

USER STUDY

We ran a controlled empirical study to examine the effect of different thumbnail generation methods on the ability of users to recognize objects in images. The experiment is divided into two parts. First, we measured how recognition rates change depending on thumbnail size and thumbnail generation techniques. Participants were asked to recognize objects in small thumbnails (Recognition Task). Second, we measured how the thumbnail generation technique affects search performance (Visual Search Task). Participants were asked to find images that match given descriptions.

Design of Study

The recognition tasks were designed to measure the successful recognition rate of thumbnail images as three conditions varied: image set, thumbnail technique, and thumbnail size. We measured the correctness as a dependent variable.

The visual search task conditions were designed to measure the effectiveness of image search with thumbnails generated with different techniques. The experiment employed a 3x3 within-subjects factorial design, with image set and thumbnail technique as independent variables. We measured search time as a dependant variable. But, since the face-detection clipping is not applicable to the Animal Set and the Corbis Set, we omitted the visual search tasks with those conditions as in Figure 8. The total duration of the experiment for each participant was about 45 minutes.

Thumbnail Technique	Animal Set	Corbis Set	Face Set
Plain shrunken thumbnail	√	√	√
Saliency based cropping	√	√	√
Face detection based cropping	X	X	√

Figure 8: Visual search task design. Checkmarks (√) show which image sets were tested with which image cropping techniques.

Participants

There were 20 participants in this study. Participants were college or graduate students at the University of Maryland at College Park recruited on the campus. All participants were familiar with computers. Before the tasks began, all participants were asked to pick ten familiar persons out of fifteen candidates. Two participants had difficulty with choosing them. Since the participants must recognize the people whose images are used for identification, the results from those two participants were excluded from the analysis.

Image Sets

We used three image sets for the experiment. We also used filler images as distracters to minimize the duplicate exposure of images in the visual search tasks. There were 500 filler images and images were randomly chosen from this set as needed. These images were carefully chosen so that none of them were similar to images in the three test image sets.

Animal Set (AS)

The “Animal Set” includes images of ten different animals and there are five images per animal. All images were gathered from various sources on the Web. The reason we chose animals as target images was to test recognition and visual search performance with familiar objects. The basic criteria of choosing animals were 1) that the animals should be very familiar so that participants can recognize them without prior learning; and 2) they should be easily distinguishable from each other. As an example, donkeys and horses are too similar to each other. To prevent confusion, we only used horses.

Corbis Set (CS)

Corbis is a well known source for digital images and provides various types of tailored digital photos [7]. Its images are professionally taken and manually cropped. The goal of this set is to represent images already in the best possible shape. We randomly selected 100 images out of 10,000 images. We used only 10 images as search targets for visual search tasks to reduce the experimental errors. But during the experiment, we found that one task was problematic because there were very similar images in the fillers and sometimes participants picked unintended images as an answer. Therefore we discarded the result from the task. A total of five observations were discarded due to this condition.

Face Set (FS)

This set includes images of fifteen well known people who are either politicians or entertainers. Five images per person were used for this experiment. All images were gathered from the Web. We used this set to test the effectiveness of face detection based cropping technique and to see how the participants’ recognition rate varies with different types of images.

Some images in this set contained more than one face. In this case, we cropped the image so that the resulting image contains all the faces in the original image. Out of 75 images, multiple faces were detected in 25 images. We found that 13 of them contained erratic detections. All erroneously detected faces were included in the cropped thumbnail sets since we intended to test our cropping method with available face detection techniques, which are not perfect.

Thumbnail Techniques

Plain shrinking without cropping

The images were scaled down to smaller dimensions. We prepared ten levels of thumbnails from 32 to 68 pixels in the larger dimension. The thumbnail size was increased by four pixels per level. But, for the Face Set images, we increased the number of levels to twelve because we found that some faces are not identifiable even in a 68 pixel thumbnail.

Cropping Technique and Image Set		Ratio	Variance
Saliency based cropping	Corbis Set	61.3%	0.110
	Animal Set	53.9%	0.127
	Face Set	54.3%	0.128
	All	57.6%	0.124
Face detection based cropping (Face Set)		16.1%	0.120

Figure 9: Ratio of cropped to original image size.

Saliency based cropping

By using the saliency based cropping algorithms described above, we cropped out background of the images. Then we shrunk cropped images to ten sizes of thumbnails. Figure 9 shows how much area was cropped for each technique.

Face detection based cropping

Faces were detected by CMU's algorithm as described above. If there were multiple faces detected, we chose the bounding region that contains all detected faces. Then twelve levels of thumbnails from 36 to 80 pixels were prepared for the experiment.

Recognition Task

We used the "Animal Set" and the "Face Set" images to measure how accurately participants could recognize objects in small thumbnails. First, users were asked to identify animals in thumbnails. The thumbnails in this task were chosen randomly from all levels of the Animal Set images. This task was repeated 50 times.

When the user clicked the "Next" button, a thumbnail was shown as in Figure 10 for two seconds. Since we intended to measure pure recognizability of thumbnails, we limited the time thumbnails were shown. According to our pilot user study, users tended to guess answers even though they could not clearly identify objects in thumbnails when they saw them for a long time. To discourage participants' from guessing, the interface was designed to make thumbnails disappear after a short period of time, two seconds. For the same reason, we introduced more animals in the answer list. Although we used only ten animals in this experiment, we listed 30 animals as possible answers as seen in Figure 10, to limit the subject's ability to guess identity based on crude cues. In this way, participants were prevented from choosing similarly shaped animals by guess. For example, when participants think that they saw a bird-ish animal, they would select swan if it is the only avian animal. By having multiple birds in the candidate list, we could prevent those undesired behaviors.

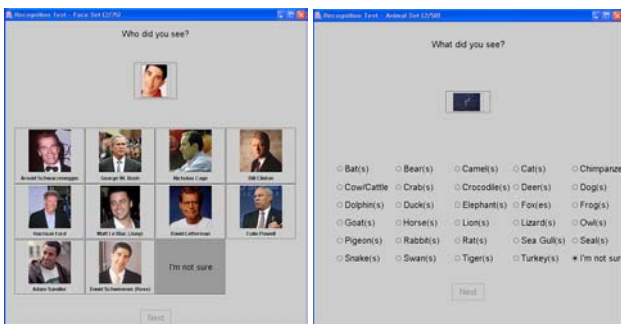


Figure 10: Recognition task interfaces. Participants were asked to click what they saw or "I'm not sure" button. Left: Face Set recognition interface, Right: Animal Set recognition interface

After the Animal Set recognition task, users were asked to identify a person in the same way. This Face Set recognition task was repeated 75 times. In this session, the candidates were shown as portraits in addition to names as seen in Figure 10.

Visual Search Task

For each testing condition in Figure 8, participants were given two tasks. Thus, for each visual search session, fourteen search tasks were assigned per participant. The order of tasks was randomized to reduce learning effects.

As shown in Figure 11, participants were asked to find one image among 100 images. For the visual search task, it was important to provide equal search conditions for each task and participant. To ensure fairness, we designed the search condition carefully. We suppressed the duplicate occurrences of images and manipulated the locations of the target images.

For the Animal Set search tasks, we randomly chose one target image out of 50 Animal Set images. Then we carefully selected 25 non-similar looking animal images. After that we mixed them with 49 more images randomly chosen from the filler set as distracters. For the Face Set and Corbis Set tasks, we prepared the task image sets in the same way.

The tasks were given as verbal descriptions for the Animal Set and Corbis set tasks. For the Face Set tasks, a portrait of a target person was given as well as the person's name. The given portraits were separately chosen from an independent collection so that they were not duplicated with images used for the tasks.

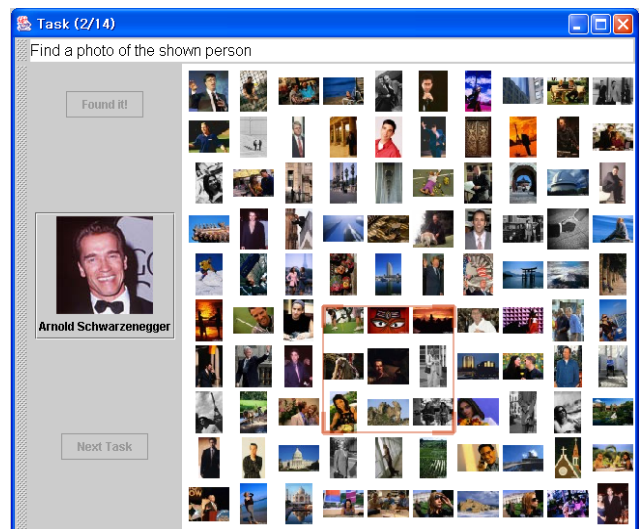


Figure 11: Visual search task interface. Participant were asked to find an image that matches a given task description. Users can zoom in, zoom out, and pan freely until they find the right image.

We used a custom-made image browser based on PhotoMesa [3] as our visual search interface. PhotoMesa provides a zooming environment for image navigation with a simple set of control functions. Users click the left mouse button to zoom into a group of images (as indicated by a red rectangle) to see the images in detail and click the right mouse button to zoom out to see more images to overview. Panning is supported either by mouse dragging or arrow keys. The animation between zooming helps user to remember where things fit together based on spatial relationships. PhotoMesa can display a large number of thumbnails in groups on the screen at the same time. Since this user study was intended to test pure visual search, all images were presented in a single cluster as in Figure 11.

Participants were allowed to zoom in, zoom out and pan freely for navigation. When users identify the target image, they were asked to zoom into the full scale of the image and click the “Found it” button located on the upper left corner of the interface to finish the task. Before the visual search session, they were given as much time as they wanted until they found it comfortable to use the zoomable interface. Most participants found it very easy to navigate and reported no problem with the navigation during the session.

RECOGNITION TASK RESULTS

Figure 12 shows the results from the recognition tasks. The horizontal axis represents the size of thumbnails and the vertical axis denotes the recognition accuracy. Each data point in the graph denotes the successful recognition rate of the thumbnails at that level. As shown, the bigger the thumbnails are, the more accurately participants recognize objects in the thumbnails. And this fits well with our intuition. But the interesting point here is that the automatic cropping techniques perform significantly better than the original thumbnails.

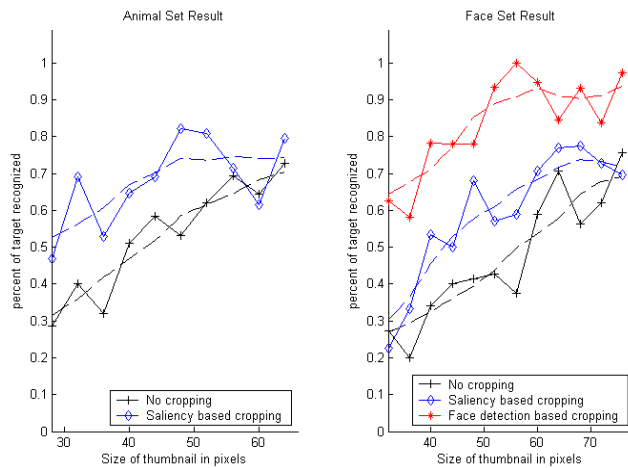


Figure 12: Recognition Task Results. Dashed lines are interpolated from jagged data points.

There were clear correlations in the results. Participants recognized objects in bigger thumbnails more accurately regardless of the thumbnail techniques. Therefore, we used Paired T-test (two tailed) to analyze the results. The results are shown in Figure 13.

The first graph shows the results from the “Animal Set” with two different thumbnail techniques, no cropping and saliency based cropping. As clearly shown, users were able to recognize objects more accurately with saliency based cropped thumbnails than with plain thumbnails with no cropping. One of the major reasons for the difference can be attributed to the fact that the effective portion of images is drawn relatively larger in saliency based cropped images. But, if the main object region is cropped out, this would not be true. In this case, the users would see more non-core parts of images and the recognition rate of the cropped thumbnails would be less than that of plain thumbnails. The goal of this test is to measure if saliency based cropping cut out the right part of images. The recognition test result shows that participants recognize objects better with saliency based thumbnails than plain thumbnails. Therefore, we can say that saliency based cropping cut out the right part of images.

Condition	t-Value	P value
No cropping vs. Saliency based cropping on Animal Set	4.33	0.002
No cropping vs. Saliency based cropping on Face Set	4.16	0.002
No cropping vs. Face Detection based cropping on Face Set	9.56	< 0.001
Saliency based cropping vs. Face detection based cropping on Face Set	7.34	< 0.001
Animal Set vs. Face Set with no cropping	5.00	0.001
Animal Set vs. Face Set with saliency based cropping	3.08	0.005

Figure 13: Analysis results of Recognition Task (Paired T-Test). Every curve in Figure 12 is significantly different from each other.

During the experiment, participants mentioned that the background sometimes helped with recognition. For example, when they saw blue background, they immediately suspected that the images would be about sea animals. Similarly, the camel was well identified in every thumbnail technique even in very small scale thumbnails because the images have unique desert backgrounds (4 out of 5 images).

Since saliency based cropping cuts out large portion of background (42.4%), we suspected that this might harm recognition. But the result shows that it is not true. Users performed better with cropped images. Even when

background was cut out, users still could see some of the background and they got sufficient help from this information. It implies that the saliency based cropping is well balanced. The cropped image shows the main objects bigger while giving enough background information.

The second graph shows results similar to the first. The second graph represents the results from the “Face Set” with three different types of thumbnail techniques, no cropping, saliency based cropping, and face detection based cropping. As seen in the graph, participants perform much better with face detection based thumbnails. It is not surprising that users can identify a person more easily with images with bigger faces.

Compared to the Animal Set result, the Face Set images are less accurately identified. This is because humans have similar visual characteristics while animals have more distinguishing features. In other words, animals can be identified with overall shapes and colors but humans cannot be distinguished easily with those features. The main feature that distinguishes humans is the face. The experimental results clearly show that participants recognized persons better with face detection based thumbnails.

The results also show that saliency cropped thumbnails are useful for recognizing humans as well as animals. We found that saliency based cropped images include persons in the photos so that persons in the images can be presented larger in cropped images. The test results show that the saliency based cropping does increase the recognition rate.

In this study, we used two types of image sets and three different thumbnail techniques. To achieve a higher recognition rate, it is important to show major distinguishing features. If well cropped, a small sized thumbnail would be sufficient to represent the whole image. Face detection based cropping shows benefits when this type of feature extraction is possible. But, in a real image browsing task, it is not always possible to know users’ searching intention. For the same image, users’ focus might be different for browsing purposes. For example, users might want to find a person at some point, but the next time, they would like to focus on costumes only. We believe that the saliency based cropping technique can be applied in most cases when semantic object detection is not available or users’ search behavior is not known.

In addition, the recognition rate is not the same for different types of images. This implies that the minimum recognizable size should be different depending on image types.

VISUAL SEARCH TASK RESULTS

Figure 14 shows the result of the visual search tasks. Most participants were able to finish the tasks within the 120 second timeout (15 timeouts out of 231 tasks) and also

chose the desired answer (5 wrong answers out of 231 tasks). Wrong answers and timed out tasks were excluded from the analysis.

A two way analysis of variance (ANOVA) was conducted on the search time for two conditions, thumbnail technique and image sets. As shown, participants found the answer images faster with cropped thumbnails. Overall, there was a strong difference for visual search performance depending to thumbnail techniques, $F(2, 219) = 5.58$, $p = 0.004$.

Since we did not look at face detection cropping for the Animal Set and the Corbis Set, we did another analysis with the two thumbnail techniques (plain thumbnail, saliency based cropped thumbnail) to see if the saliency based algorithm is better. The result shows a significant improvement on visual search with saliency based cropping, $F(1, 190) = 3.823$, $p = 0.05$. We therefore believe that the proposed saliency based cropping algorithm make a significant contribution to visual search.

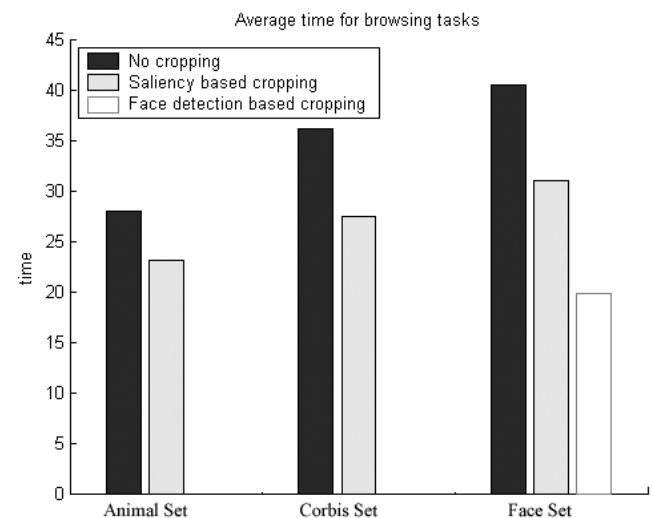


Figure 14: Visual search task results.

Condition	F value	P value
Thumbnail techniques on three sets	5.58	0.004
Thumbnail techniques on Face Set	4.56	0.013
No cropping vs. Saliency based thumbnail on three image sets	3.82	0.052
Three image sets regardless of thumbnail techniques	2.44	0.089

Figure 15 List of ANOVA results from the visual search task

When the results from the Face Set alone were analyzed by one way ANOVA with three thumbnail technique conditions, there also was a significant effect, $F(2, 87)=4.56$, $p = 0.013$. But for the Animal Set and the Corbis

Set, there was only a borderline significant effect over different techniques. We think that this is due to the small number of observations. We believe those results would also be significant if there were more participants because there was a clear trend showing an improvement of 18% on the Animal Set and 24% on the Corbis Set. Lack of significance can also be attributed to the fact that the search task itself has large variances by its nature. We found that the location of answer images affects the visual search performance. Users begin to look for images from anywhere in the image space (Figure 11). Participants scanned the image space from the upper-left corner, from the lower-right corner, or sometimes randomly. If the answer image is located in the initial position of users' attention, it would be found much earlier. Since we could not control users' behavior, we randomized the location of the answer images. But as a result, there was large variance.

Before the experiment, we were afraid that the cropped thumbnails of the Corbis Set images would affect the search result negatively since the images in the Corbis Set are already in good shape and we were concerned that cutting off their background would harm participants' visual search. But according to our result, saliency based cropped thumbnails does not harm users' visual search. Rather, it showed a tendency to increase participants' search performance. We think that this is because the saliency based cropping algorithm cut the right amount of information without removing core information in the images. At least, we can conclude that it did not make visual search worse to use the cropped thumbnails.

Another interesting thing we found is that the visual search task with the Animal Set tends to take less time than with the Corbis Set and the Face Set, $F(2, 219) = 2.44$, $p = 0.089$. This might be because the given Corbis Set and Face Set tasks were harder than the Animal Set. But we think there is another interesting factor. During the experiment, when he found the answer image after a while, one participant said that "Oh... This is not what I expected. I expected blue background when I'm supposed to find an airplane." Since one of the authors was observing the experiment session, it was observed that the participant passed over the correct answer image during the search even though he saw the image at reasonably big scale. Since all of the visual search tasks except finding faces were given as verbal descriptions, users did not have any information about what the answer images would be like. We think that this verbal description was one of the factors in performance differences between image sets. We found that animals are easier to find by guessing background than other image sets.

DISCUSSION AND CONCLUSION

We developed and evaluated two automatic cropping methods. A general thumbnail cropping method based on a saliency model finds the informative portion of images and cuts out the non-core part of images. Thumbnail images generated from the cropped part of images increases users' recognition and helps users in visual search. This technique is general and can be used without any prior assumption about images since it uses only low level features. Furthermore, it also can be used for images already in good shape. Since it dynamically decides how much to cut away, it can prevent cutting out too much.

The face detection based cropping technique shows how semantic information can be used to enhance thumbnail cropping. With a face detection technique, we created more effective thumbnails, which significantly increased users' recognizing and finding performance.

Our study shows strong empirical evidence that the more salient a portion of image, the more informative it is. We also showed that using more recognizable thumbnails increases visual search performance.

Another finding of interest is that users tend to have mental models about search targets. Users tend to develop a model about what a target will look like by guessing its color and shape. We observed that they spent a long time searching or even skipped the correct answer when their guesses were wrong or they were unable to guess. It is known that humans have an "attentional control setting" – a mental setting about what they are (and are not) looking for while performing a given task. Interestingly, it is also known that humans have difficulty in switching their attentional control setting instantaneously [11]. This theory explains our observation. We think that this phenomenon should be regarded in designing image browsing interfaces especially in situations where users need to skim a large number of images.

There are several interesting directions for future research. One direction involves determining how to apply these techniques to other browsing environments. In our study, we used a zoomable interface for visual search. We believe that the image cropping techniques presented in this paper can benefit other types of interfaces that deal with a large number of images as well. While our research confirms that well cropped thumbnails can increase users' visual search performance, we did not try to build a model about recognition, attention and its relationship on image browsing. Further research about human's attention and perception model [18][21] would help designing a better image browsing system.

Another interesting direction would be to combine image adaptation techniques (i.e. saliency based smoothing) with the image cropping techniques. This would allow faster thumbnail processing and delivery for thumbnail-based retrieval systems.

ACKNOWLEDGMENTS

We would like to acknowledge the face group at Carnegie Mellon University for providing resources for face detection processing.

REFERENCES

1. ACDSee, ACD Systems, <http://www.adsystems.com>
2. Adobe Photoshop Album, Adobe Systems Inc., <http://www.adobe.com/products/photoshopalbum/>
3. Bederson, B. B. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. *UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters*, 3(2), pp. 71-80. 2001.
4. Burton, C., Johnston, L., and Sonenberg, E. Case Study: An Empirical Investigation of Thumbnail Image Recognition, *In Proc. of Information Visualization, Atlanta, Georgia*, pp115-121, 1995.
5. Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. A Visual attention model for adapting images on small displays, *ACM Multimedia Systems Journal*, To appear in Fall 2003.
6. Chen, X., and Zhang, H. Text Area Detection from Video Frames. *In Proc. Of 2nd IEEE Pacific-Rim Conference on Multimedia (PCM2001)*, Beijing, China, pp. 222-228, October 2001.
7. Corbis, <http://www.corbis.com>
8. DeCarlo, D., and Santella, A. Stylization and Abstraction of Photographs, *In Proc. on ACM SIGGRAPH 2002*, pp. 769-776, 2002.
9. Face Detection Demonstration. Robotics Institute, Carnegie Mellon University <http://www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi>
10. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. Query by Image and Video Content: The QBIC System, *IEEE Computer*, Volume: 28, Issue: 9 , pp.23 -32, Sept. 1995.
11. Folk, C.L., Remington, R.W., and Johnston, J.C. Involuntary covert orienting is contingent on attentional control settings. *Journal of Experimental Psychology: HP&P*, 18:1030-44, 1992.
12. Itti, L., and Koch, C. A Comparison of Feature Combination Strategies for Saliency-Based Visual Attention Systems, *SPIE human vision and electronic imaging IV(HVEI'99)*, San Jose, CA, pp. 473-482, 1999.
13. Itti, L., Koch, C., and Niebur, E., A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), pp. 1254-9, 1998.
14. Kang, H., and Shneiderman, B. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder, *In Proc. of IEEE International Conference on Multimedia and Expo (ICME2000)* New York: IEEE, pp. 1539-1542, 2000.
15. Li, S., Zhu, L., Zhang, Z., Blake, A., Zhang, H., and Shum, H. Statistical Learning of Multi-view Face Detection. *In Proc. of European Conference on Computer Vision (ECCV) 2002*, Vol. 4, pp. 67-81, 2002.
16. Milanese, R., Wechsler H., Gil S., Bost J., and Pun T. Integration of Bottom-Up and Top-Down Cues for Visual Attention Using Non-Linear Relaxation, *In proc. of Computer Vision and Pattern Recognition, IEEE*, pp. 781-785, 1994.
17. Milanese, R. Detecting Salient Regions in an Image: from Biological Evidence to Computer Implementation, *Ph.D. thesis, Univ. of Geneva*, 1993.
18. Palmer, J. Ames, C.T., Lindsey, D.T. Measuring the Effect of Attention on Simple Visual Search. *Journal of Experimental Psychology: Human Perception & Performance*, 19, pp. 108-130, 1993.
19. Schneiderman, H., and Kanade, T. A Statistical Model for 3D Object Detection Applied to Faces and Cars. *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, June, 2000.
20. Vimas Technologies. <http://www.vimas.com>
21. Wolfe, J.M. Guided Search 2.0: A Revised Model of Visual Search, *Psychonomic Bulletin and Review*, Vol. 1, No. 2, pp. 202-238, 1994.
22. Yang, M., Kriegman, D., and Ahuja, N. Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Mach Intelligence*, 24(1), pp. 34-58, 2002.

FIND THAT PHOTO!

Interface Strategies to Annotate, Browse, and Share

By BEN SHNEIDERMAN, BENJAMIN B. BEDERSON,
and STEVEN M. DRUCKER

As digital photos become the standard media for personal photo taking, supporting users to explore those photos becomes a vital goal. Dominant strategies that have emerged involve innovative user interfaces that support annotation, browsing, and sharing that add up to rich support for exploratory search. Successful retrieval is based largely on attaching appropriate annotations to each image and collection since automated image content analysis is still limited. Therefore, innovative techniques, novel hardware, and social strategies have been proposed. Interactive visualization to select and view dozens or hundreds of photos extracted from tens of thousands has become a popular strategy. And since the goal of photo search is to support sharing, storytelling, and reminiscing, experiments with new collaborative strategies are being examined.

While digital photographic databases and retrieval systems have been in use for many years, these systems were typically designed for

professionals in museums, libraries, advertising, and journalism, to name a few specialties. Such systems employed a cadre of financially motivated individuals to hand-annotate the pictures with metadata such as keywords, dates, and locations, often using fixed vocabularies, to support traditional search techniques. By contrast, consumers typically put little effort into photo annotation; they are more focused on exploratory search and serendipitous discovery of photos with a stronger emphasis on entertainment. This leads to a very different set of requirements for personal photo use where ease of annotation, support for exploratory browsing, and convenient sharing is crucial.

Annotate. In textual exploratory search, users can enter key phrases from a document to retrieve similar content. But for images, retrieval based on content through automated analysis is often limited to some forms of shape analysis (such as finding the presence of faces in an image) and color matching to find sunrises or determine whether an image was taken inside or outside.

To support effective exploratory search on photos, appropriate annotations must be asso-





Figure 1. The WorldWide Media Exchange (WWMX) interface showing map and calendar views along with images as published in ACM Multimedia 2003; wwwx.org.

ciated with the images either by the camera or by users of the images, such as the photographer or potentially a larger community of users. Cameras are increasingly recording information about the photograph including time and date stamps, tilt sensors for orientation, light levels, focal distances, and even global position. Barcodes, RFID tags, or other labeling methods could enable a higher percentage of photos to be annotated automatically.

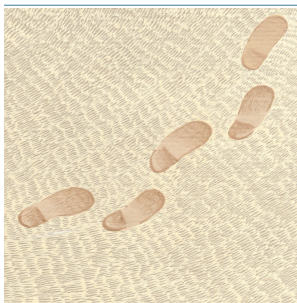
Many interfaces enable manual annotation of photographs by “painting” keywords [3] or dragging and dropping names onto images. Commercial tools such as Adobe PhotoShop Album make tags drag-able onto photo borders. Other tools perform temporal clustering to create a more manageable set of photo groups [1]. As with many tasks, manual annotation can be improved by designing interfaces that support faster and easier annotation as

well as making the future benefits more apparent. Automatic and manual annotations are valuable in supporting both searching and browsing.

Browse. Users browse for fun and to find a specific photograph. They may be looking for photos of their grandfather, their hike down the Grand Canyon, or a friend’s wedding. They also may be looking for a great photo to accompany a story of a sunrise hike or memorable baseball game.

Clearly, if the photo collection has been extensively annotated, techniques such as faceted search (see Hearst’s article in this section) can help users filter down a collection and show potential targets for browsing. User-controlled visualization of photos grouped by date, location, or annotation can greatly facilitate browsing and increase enjoyment [4]. Different layouts of photos can exploit this metadata to help people find desired photos and discover new ones. In particular, geo-tagging of photos and interfaces, like WWMX, allows people to find all those photographs associated with a particular area (see Figure 1).

Chronological displays work well for dates as well, but large numbers of photos can be overwhelming, so groups of photos can be *clustered* by date and representative photos can be manually or automatically chosen for each cluster [1, 2]. These representative photos again help to provide landmarks in order for users to locate photos from particular events. Interfaces such as PhotoMesa use powerful filtering tools, plus flexible grouping and rapid zooming, to enable users to explore thousands of photos fluidly (see Figure 2).



Consumers typically put little effort into photo annotation; they are more focused on exploratory search and serendipitous discovery of photos with a stronger emphasis on entertainment.

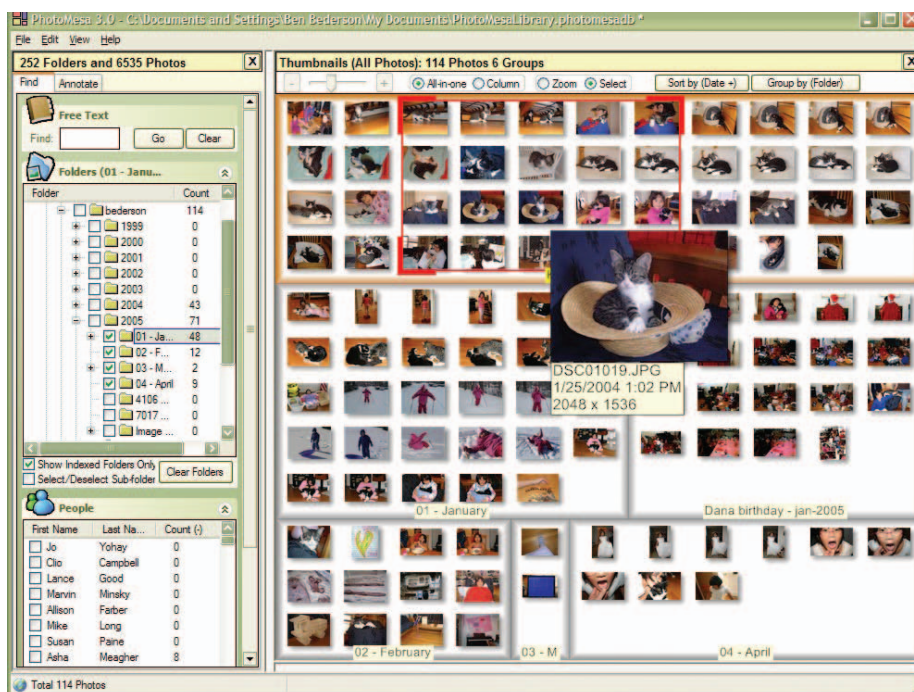


Figure 2. PhotoMesa showing 114 photos in six groups in a single view with integrated annotation and search tools as published in ACM UIST 2001; (courtesy of www.photomesa.com).

select photos and make them available to others, they seem to be willing to invest more effort in annotation. Also by making them public, they invite others to comment and add annotations. More elaborate story-generating tools invite users to provide slideshow sequences with text captions and audio narration.

Recent innovations in social experiences on the Web have sought to encourage annotation by increasing satisfaction and making the benefits immediately apparent. A game-like approach to image annotation gets players to cooperate with anonymous, remotely located partners in assigning keywords for photographs [5]. This surprisingly addictive game has succeeded in labeling over 10 million images as of August 2005 (since its introduction in 2003). Other communities, such as Flickr, allow users to share and annotate images on a Web site using tags. These “folksonomies” have now gone past photos to Web pages and blogs as well (such as technorati and deli.cio.us).

The trend toward annotating, browsing, and sharing your photos via Web sites such as Flickr, Ofoto, and Shutterfly is perhaps one of the biggest changes enabled by the transformation from analog to digital photography. Photos no longer sit unattended in shoeboxes stored in attics, but are available for ready viewing by friends and family distributed around the world.

Share. Sharing photos by email, instant messaging, Web sites, and cell phones is a growing success story. When users

phers. However, we see significant work remaining, especially in metadata standardization to help users cope with their rapidly growing and increasingly valued collections. **C**

SUMMARY

A combination of annotation, browsing, and sharing of photos can support the special exploratory search needs of personal digital photo users by getting around the fact that direct search of image content continues to be beyond the capabilities of current systems.

The special needs of amateur digital photographers are pushing the photo industry to support users with their desired activities. Social networking, in combination with innovative user interfaces and visualization, is just beginning to support everyday photogra-

REFERENCES

1. Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T. Time as essence for photo browsing through personal digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries* (2002). ACM Press, NY, 326–335.
2. Huynh, D., Drucker, S., Baudisch, P., and Wong, C. Time Quilt: Scaling up zoomable photo browsers for large, unstructured photo collections. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2005). ACM Press, NY, 1937–1940.
3. Kuchinsky, A., Pering, C., Creech, M., Freeze, D., Serra, B., and Gwizdka, J. FotoFile: A consumer multimedia organization and retrieval system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1999). ACM Press, NY, 496–503.
4. Kustanowitz, J. and Shneiderman, B. Meaningful presentations of photo libraries: Rationale and applications of bi-level radial quantum layouts. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (2005). ACM Press, NY, 188–196.
5. van Ahn, L. and Dabbish, L. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2004). ACM Press, NY, 319–326.

BEN SHNEIDERMAN (ben@cs.umd.edu) is a professor and the founding director of the Human-Computer Interaction Lab, Computer Science Department, at the University of Maryland, College Park, MD.

BENJAMIN B. BEDERSON (bederson@cs.umd.edu) is an associate professor and director of the Human-Computer Interaction Lab, Computer Science Department, at the University of Maryland, College Park, MD.

STEVEN M. DRUCKER (sdrucker@microsoft.com) is lead researcher of the Next Media Research Group, Microsoft Research, Redmond, WA.

Capture, Annotate, Browse, Find, Share: Novel Interfaces for Personal Photo Management

Hyunmo Kang

Human-Computer Interaction Laboratory,
University of Maryland Institute for Advanced Computer Studies, College Park, MD

Benjamin B. Bederson

Human-Computer Interaction Laboratory,
Computer Science Department, University of Maryland
Institute for Advanced Computer Studies, College Park, MD

Bongwon Suh

Palo Alto Research Center, Palo Alto, CA

The vision of ubiquitous digital photos has arrived. Yet, despite their broad popularity, significant shortfalls remain in the tools used to manage them. We believe that with a bit more creativity and effort, the photo industry can solve many of these problems, offering tools which better support accurate, rapid, and safe shared annotations with comfortable and efficient browsing and search. In this article, we review a number of projects of ours and others on interfaces for photo management. We describe the problems that we see in existing tools and our vision for improving them.

1. INTRODUCTION

The days when people debated the relative merits of film versus digital imagery now seem almost quaint. And with hindsight, film seems destined to have been a chapter in history along with LP vinyl records—a temporary physical analog recording medium. Although some aficionados may still prefer some qualities of those dust-gathering mediums, the advantages of digital media have become clear. The rapid and inexpensive ability to edit, annotate, search, share, and access has brought digital media to ubiquity.

And yet, with all this promise, shortfalls remain in the overall user experience. How many of us have replaced old unlooked-at shoeboxes of prints with unlooked-at digital archives of image files? How much has our ability to find a particular set of

Color figures are available at <http://www.cs.umd.edu/hcil/ben60>

Correspondence should be addressed to Hyunmo Kang, AVW 3211, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. E-mail: kang@cs.umd.edu

photos really improved (i.e., can you find those photos of a visiting uncle with your sister when they were children?) How do we record stories from our parents describing family photos? How do we make sure those stories stay with the photo in question and get distributed to all copies of that photo within the family? Perhaps of most concern, how do we ensure that these annotations stand the test of time and remain accessible as computers, file formats, recording mediums, and software change?

These changes are all happening within the context of human behavior, which does not change so rapidly. People still like immediate gratification: take pictures rapidly, print them, and share in social settings. Some people spend a lot of effort creating photo albums or “scrapbooking.” Of course, many do not. Understanding which behaviors are fundamental and which are side effects of current technology is crucial, because this understanding can and should influence where researchers spend their effort.

We explore these issues and more with much of the intellectual motivation coming from Ben Shneiderman, our close colleague who has pushed for a deeper understanding of and better support of photo management for more than 10 years. His personal photo archives document the field of human–computer interaction (HCI) going back to its beginning. He regularly shares those photos with great enthusiasm to visitors, motivating and exciting all of us—largely because of the care and consistency he has applied to annotating and organizing his photos. He regularly pulls up old photos of lab visitors showing everyone what they worked on 5 or 15 years ago (and of course, showing what they looked like too!). His early exploration of tools to support photo management (with co-author Kang) led to PhotoFinder (Kang & Shneiderman, 2000), and the ensuing PhotoMesa tools (Bederson, 2001; Bederson, Shneiderman, & Wattenberg, 2002). His personal interest helped inspire the authors of this article as well as other lab members to pursue the development of approaches and software to improve all of our user experiences when managing photos.

This, of course, all happened during a time of tremendous commercial activity in this area. There are wildly popular photo sharing Web sites, such as Flickr (<http://www.flickr.com>; Yahoo!), Picasa Web Albums (<http://picasaweb.google.com>; Google), Snapfish (<http://www.snapfish.com>; HP), Shutterfly (<http://www.shutterfly.com>; Shutterfly), and PhotoBucket (<http://www.photobucket.com>; Photobucket), as well as equally well-used desktop photo applications such as Google Picasa (<http://www.picasa.com>), Adobe PhotoShop Album (<http://www.adobe.com/products/photoshopalbum/starter.html>), and ACDSee (<http://www.acdsee.com>). The two approaches (desktop application and Web site) are interesting to look at because they each offer distinct advantages to users. For example, Web sites are available anywhere and facilitate sharing, whereas desktop applications are faster, support higher resolution photos more easily, provide local ownership of photos, and offer richer interaction capabilities. It is interesting that each approach is gaining characteristics of the other. Web applications begin to offer dynamic, interactive content rather than static html pages through AJAX and Flash technologies. In addition, they often include plug-ins to ease uploading or improve performance, and some offer APIs to enable desktop applications to access their data directly. At the same time, many desktop applications are offering Web capabilities such as sharing.

Yet even with this commercial activity, the full potential of personal photo management has not been reached. There is the opportunity for richer annotation interfaces, automated content analysis, improved sharing, and more creative organizational strategies. Our hope is that more photos end up with better metadata, enabling faster, easier, and more accurate and enjoyable retrieval and use.

In this article, we look at some of the key activities and behavior patterns of personal photo users and examine how innovative user interfaces have the potential to enhance users' power, satisfaction, and control in managing and exploring their images. Starting with a close look at annotation, we examine how a combination of manual and automated techniques can improve how people associate metadata with photos. We then look at how the resulting richer metadata can enable better interfaces for searching and browsing photos. Finally, we end with a discussion about the importance of sharing photos and how new interfaces enable that.

2. GUIs FOR ANNOTATION

An essential question is, How valuable is photo metadata? Our own assessment of user needs (Shneiderman & Kang, 2000) coupled with reports from other researchers (Chalfen, 1987; Naaman, 2005; Rodden & Wood, 2003), and our personal experience come together on this. They indicate that the photo metadata such as dates, locations, and content of the photos (especially people's names) play a crucial role in management and retrieval of personal photo collections.

However, in spite of the high utility of the photo metadata, the usability of software for acquiring and managing the metadata has been poor. The manual photo annotation techniques typically supported by photo software are time-consuming, tedious, and error prone, and users typically put little effort into annotating their photos. In fact, the industry attitude tends to be that because users do not annotate their photos very much, it is not necessary to spend much energy adding good support for it. However the success of keyword labeling, so-called *tagging*, systems such as del.icio.us (<http://www.del.icio.us>) and Flickr hints that users indeed want to make annotations when reasonable utility and usability are supported. We believe that the photo annotation has enough utility for some users and it is the usability of software that needs to be improved.

In this section, a few innovative approaches are presented to show how interaction and graphical user interface (GUI) design can improve the usability of the photo annotation process when they are based on careful understanding of users' behavior and usage pattern. In addition, we explain how those designs have been evolving over time to support a broader range of annotation tasks by combining the accessible technologies with the analysis of users and their needs.

2.1. Advanced Manual Annotation: Direct Annotation

Because annotations based on automated image content analysis are still limited, we developed an advanced manual annotation mechanism that can significantly reduce users' annotation workload under certain circumstances. From the

observations of personal photo annotations (Shneiderman & Kang, 2000), we found that there were three interesting characteristics that could be useful for our interaction design.

- Personal photo libraries often contain many images of the same people at different events. In the libraries we looked at, we typically found 100 to 200 identifiable people in several thousand photos. Furthermore, the library has a highly skewed distribution with immediate family members and close friends appearing very frequently.
- Textual search often doesn't work reliably because of inconsistency in names with misspellings or variants (e.g., Bill, Billy, William).
- Lists of names of people appearing in photos are often difficult to associate with individuals, especially in group shots. Textual captions often indicate left-to-right ordering in front and back rows, or give even more specific identification of who is where. However this approach is tedious and error prone.

Based on these observations, we collaborated with Ben Shneiderman to develop the concept of *direct annotation* (U.S. Patent #7010751), which is a technique using selectable, draggable labels that can be placed directly on the photo (Shneiderman & Kang, 2000). Similar interfaces have also appeared recently on Flickr and MySpace. Users can select from a scrolling or pop-up list and drag by mouse or touch screen. This applies direct manipulation principles (Shneiderman, 1983, 2005) that avoid the use of a keyboard, except to enter a name the first time it appears. The name labels can be moved or hidden, and their presence is recorded in the database or in the header of an image file in a resolution-independent manner. The relative location of the target is stored based on an origin in the upper left-hand corner of the photo with the point in the range (0, 0) – (1.0, 1.0) corresponding to the full image. This approach not only associates a name with a position on the photo but also ensures that each name is always spelled the same way.

This simple rapid process also enables users to annotate at any time. They can add annotations when they first see their photos on the screen, when they review them and make selections, or when they are showing them to others. This design, which supports continuous annotation encouraged users to do more annotation, especially in collaborative situations such as with PhotoFinder Kiosk (Kules, Kang, Plaisant, Rose, & Shneiderman, 2004; Shneiderman et al., 2002). In a public setting, visitors of PhotoFinder Kiosk added 1,335 name labels using direct annotation while adding 399 captions using traditional type-in method.

The direct annotation mechanism was revised later so that it enables users to define their own categories such as activities, events, locations, objects in a photo in a hierarchical way (Figure 1) in addition to person names. A few alternative and complementary direct annotation mechanisms such as split menu annotation, hotkey annotation, and pop-up menu annotation have also been designed and developed to accelerate the annotation process. These are described in more detail in the following subsection.

A pilot experiment was conducted to see if the direct annotation method improved the annotation process in terms of annotation time and users' subjective preference compared with the traditional caption method or the click-and-type

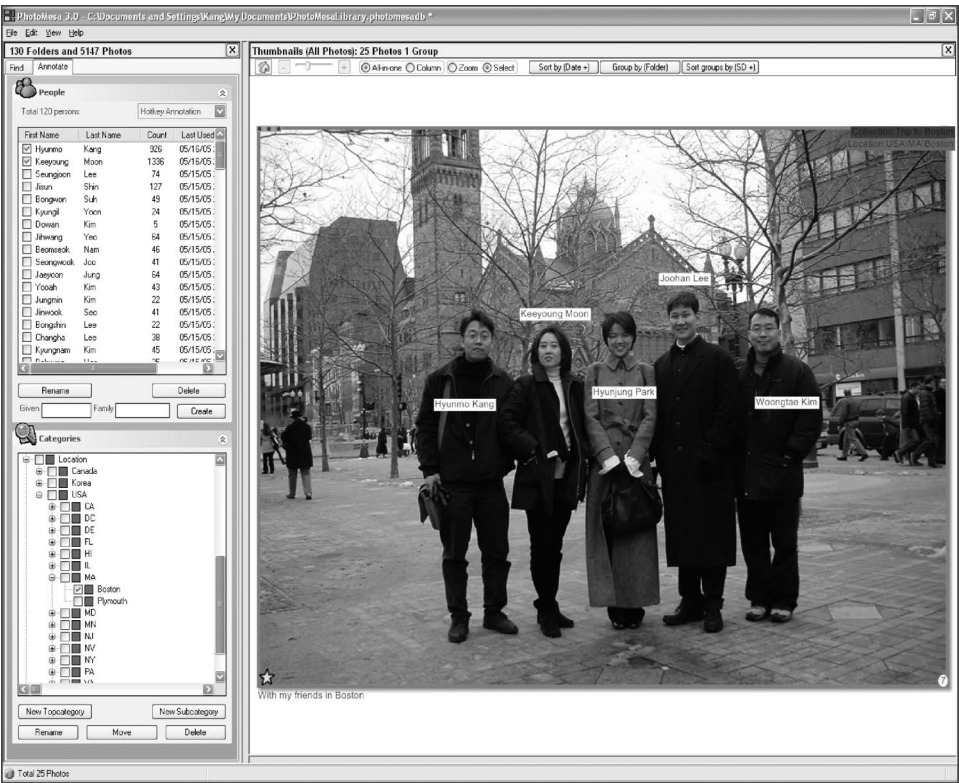


FIGURE 1 The revised direct annotation mechanisms as implemented in PhotoMesa, the successor to PhotoFinder. *Note.* Users can add a caption under a photo, or add a particular attribute such as favorite (a yellow star on the bottom left of photo) or hidden. In addition, labels can be dragged from the list of people (on the top left) or from the user-defined category tree (on the bottom left). A label can be directly placed on the photo to represent where the individuals or objects are located within the photo.

method (Goto, Jung, Ma, & McCaslin, 2000; Jung, 2000). Forty-eight volunteers participated in a within-subject design, whereby each participant attempted an annotation task (20 names in five photos) on each system. The direct annotation method was significantly preferred, but no significant difference was found in the mean annotation time. A more extensive user study might help identify under what circumstances (e.g., number of total labels, average number of people in a picture, pattern of people’s appearance in a personal photo collection, etc.) different annotation mechanisms work better than others in terms of completion time, error rate, users’ satisfaction, or confidence.

2.2. Enhanced Direct Annotation: Bulk Annotation

The direct annotation mechanism was enhanced through a series of design cycles to support more efficient and productive annotation. Perhaps the most notable

one was applying the *bulk annotation* technique, which lets users annotate several photos with a single interaction (Kuchinsky et al., 1999). Bulk annotation is especially useful when users have a set of photos with the same person or group of people involved in the same events. We thus designed and developed several variations of direct annotation and bulk annotation mechanisms to accelerate the annotation process.

The split menu annotation mechanism (Figure 2a) was designed to minimize users' time of scrolling the list to find correct labels to be used for photo annotation. The split menu featured a resizable splitter bar so that the number of the most frequent labels displayed in the top window (Sears & Shneiderman, 1994). The scrollbar was removed from the top window, whereas the bottom half retained its scrollbar. The split menu raises interesting questions about what kind of automatic algorithms should be used to predict users' future access and facilitate rapid annotation.

Because some annotations get used so much more frequently than others, we designed the interface so that each label (either person's name or categories) can be associated with a hotkey (Figure 2b). After the key is assigned to a particular label, users can press that key whenever the mouse is over a photo. At that point, the spot that the cursor was over will be annotated with the specified label. When the expert user has a good idea about who or what is to be annotated, then the hotkeys can be put to work very efficiently. Instead of having to find the name desired and drag the name over to the position, the user can simply position the mouse and press the hotkey. This method is especially useful when a photo collection has only a few names that appear frequently.

Pop-up menu annotation (Figure 2c) also aims at reducing mouse movement in selecting labels to be used for annotation. This method offers a menu when the right mouse button is pressed on a picture. The menu consists of the currently selected labels in the list so that users can annotate pictures with a label in the menu at the position the right mouse button was pressed. This mechanism was revised again later as Label Paint Annotation so that a photo can be annotated with the currently selected labels whenever users click on a photo without selecting a pop-up menu.

In addition to the proposed methods for improving the speed in annotating individual photos, we also designed two additional methods for improving the efficiency of bulk annotation as follows:

- **Checkbox Annotation:** Select one or more photos and click the check box next to the label in the list (Figure 1).
- **Drag-Drop Annotation:** Select one or more photos and drag and drop the selected labels onto the photo. All the photos that were selected get annotated with the labels.

The best solution may be to include as many methods as possible while providing options for users to select and customize the methods. However it raises the question of how much the user can learn at first; if presented by too many options in the beginning, the user may become confused and frustrated. Therefore it may be optimal to use the multilayered approach (Kang, Plaisant, & Shneiderman, 2003, Shneiderman, 2003) when designing the initial interface.



(a) Split menu annotation



(b) Hotkey annotation



(c) Pop-up menu annotation

FIGURE 2 Enhanced direct annotation approaches within PhotoFinder.

2.3. Semiautomatic Annotation

The performance of direct annotation may be improved by supporting bulk annotation as described in the previous section. However this approach introduces two new challenges. First, unless photos to be annotated with the same labels are clustered together in some way so that they can be selected together, users need to manually select photos one by one before applying bulk annotation. Second, because multiple photos are annotated with a single interaction, the position of the labels in each photo cannot be explicitly specified by users.

To cope with these challenges, we explored a semiautomatic annotation strategy that takes advantage of human and computer strengths (Suh & Bederson, 2007). The semiautomatic approach enables users to efficiently update automatically obtained metadata interactively and incrementally. Even though automatically identified metadata are compromised with inaccurate recognition errors, the process of correcting inaccurate information can be faster and easier than manually adding new metadata from scratch. To facilitate efficient bulk annotation, two clustering algorithms were introduced to generate meaningful photo clusters (Suh & Bederson, 2007): *hierarchical event clustering* and *clothing-based person recognition*. The first method clusters photos based on their time stamps so that each group of photos corresponds to a meaningful event. The hierarchical event clustering identifies multiple event levels in personal photo collection and allows users to choose the right event granularity. The second method uses a clothing-based human model to group similar-looking people together. The clothing-based human model is based on the assumption that people who wear similar clothing and appear in photos taken within relatively short periods of time are very likely to be the same person.

To explore our semiautomatic strategies, we designed and implemented a prototype called SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface). The prototype provides an annotation framework focusing on making bulk annotations on automatically identified photo groups. The two automatic clustering algorithms provide meaningful clusters to users and make annotation more usable than relying solely on vision techniques such as face recognition or human identification in photos. In addition, the clothing-based person recognition automatically detects the positions of people in a photo, which can be used to position annotations. It is interesting that, although we found that absolute performance using this approach was not much better than manual annotation, the user study showed that the semiautomatic annotation was significantly preferred over manual annotation (Suh & Bederson, 2007). As computer vision techniques improve, the relative advantage of semiautomated techniques can only improve.

Figure 3 shows a prototype of SAPHARI, which shows the clusters of identified people whose upper bodies are cropped from photos of the same event. Because the automatic clustering techniques can also have errors, we designed the GUI so that users can correct errors manually by dragging a photo into the correct group. After error correction, users can annotate multiple photos at once (also with position information) by dragging a label onto the group.

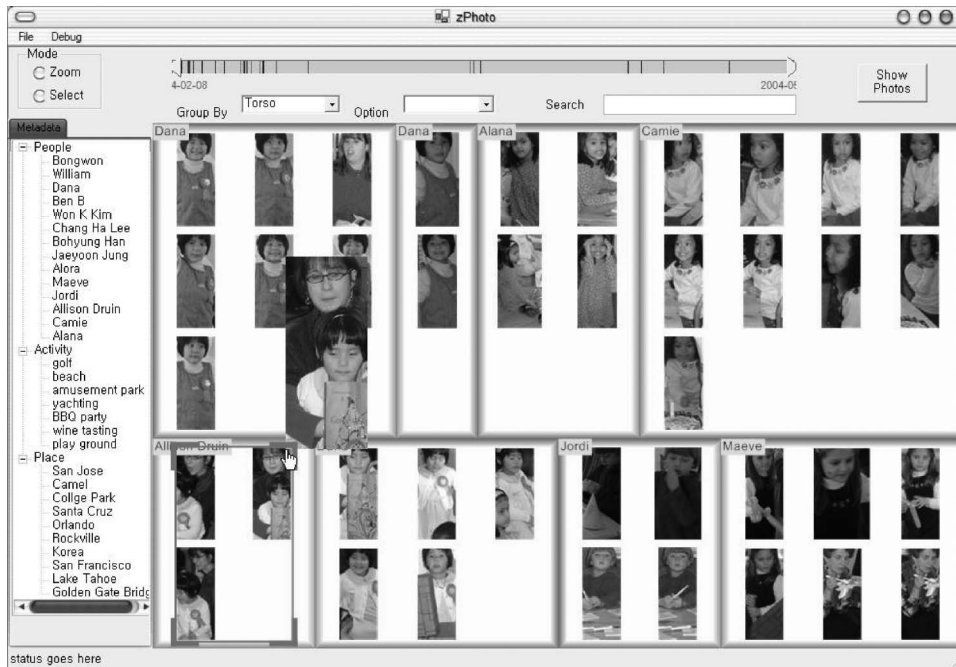


FIGURE 3 Identified people who are cropped from photos are laid out on the screen in the SAPHARI prototype. *Note.* Similar looking people are clustered based on their clothing, facilitating bulk annotation.

2.4. Image Annotation Discussion

There has been significant research on how to acquire useful metadata for images.

Many researchers have focused on automatic extraction of metadata by understanding images. For example, researchers have focused on automatic object detection such as face recognition and content-based categorization and retrieval (Yang, Kriegman, & Ahuja, 2002; Yoshitaka & Ichikawa, 1999; Zhao, Chellappa, Philips, & Rosenfeld, 2003; Chellappa, Wilson & Sirohey, 1995). However, such automatic techniques have achieved limited success so far when applied to personal photo management (Phillips et al., 2003; Suh & Bederson, 2007).

Rather than pure image-based approaches, some researchers used related context information to identify relevant metadata. For example, Naaman, Yeh, Garcia-Molina, and Paepcke (2005) used time and location information to generate label suggestions to identify the people in photos. Lieberman and Liu (2002) used relationships between text and photographs to semiautomatically annotate pictures.

On the other hand, Web-based collaborative approaches to collecting metadata become popular. Web pages, online photographs, and Web links have been actively annotated with rich metadata through tagging (e.g., Flickr and Del.icio.us). Tagging is a simple kind of annotation where keywords can be

associated with objects such as photos. However, the simple underlying approach belies the richness and power gained by broad communal access to these tags which enable cross-collection aggregation and searching along with innovative interfaces for creating and presenting those tags (i.e., tag clouds). With Flickr, users can share personal photos with friends, family, and colleagues, allowing the invited users to make annotations on shared photos. When users select photos and make them available to others, they seem to be willing to invest more effort in annotation (Shneiderman, Bederson, & Drucker, 2006). Also, by making them public, they invite others to comment and add annotations. We believe that this “folksonomy” based collaborative annotation has great potential for creating more useful metadata.

On the other hand, folksonomy based tagging system has a set of limitations. Folksonomic tagging can be inconsistent because of its vocabulary problem (Furnas, Landauer, Gomez, & Dumais, 1987). In folksonomy-based systems, there is no bona fide standard for selecting a tag, and users can choose to use any word as a tag. For example, one can use the tag *TV*, whereas others choose to use the tag *television*. Furthermore, Furnas et al. showed that it is not always easy for users to come up with good descriptive keywords that can be shared. Tags, therefore, can be easily idiosyncratic and often causes meta-noise, which decreases the retrieval efficiency of systems. Going further with collaborative annotation approaches, Von Ahn addressed the challenge by adding gaming to the mix. The ESP Game (Von Ahn & Dabbish, 2004) combines a leisure game with practical goals. It lets people play an image guessing game that gathers image metadata as a side effect of their playing.

Another ongoing challenge for metadata management is where to store the metadata. It is crucial that users own their metadata just as much as they own their photos. Yet some companies (such as Apple with their iPhoto software) create a custom database that separates metadata from the photos, leaving no easy way to get the annotations back for sharing or for use in other programs. Software developers tend to like centralized metadata stores because they make it easier to write software that perform fast searches (and because they “lock in” customers to their products). However this approach is not necessary. Instead, we suggest storing all annotations and metadata using the “IPTC” standard format (International Press Telecommunications Council; <http://www.iptc.org>) in the “EXIF” header (Japan Electronic and Information Technology Industries Association, 2002) of JPEG images. Then the application can create a cached copy of the metadata for efficient operations while leaving the “original” metadata with the photo. This is how we implemented PhotoMesa, and it appears to be the same approach taken by some other commercial software such as Picasa and ACDSee. However, even this approach has some challenges because of the following limitations:

- Not every image format supports metadata fields in the file headers.
- Metadata standards (i.e., EXIF/IPTC) do not have rich enough fields for objects such as people and hierarchical categories.
- Some operating systems make it difficult and possibly dangerous to modify the image header without changing the file modification date (which users sometimes use to determine when a file was created).

Although the issues and challenges relating to metadata standards are significant, they are beyond the scope of this article. We thus raised these few issues to point out the importance of getting it right, as it has such a dramatic affect on user experience. We therefore call on the industry to collaborate more openly in developing consistent and rich photo metadata standards addressing at least the aforementioned issues.

Technological change in society is rapid. There are new opportunities that we do not explore in this article. For example, the growing number of cell phones and even dedicated cameras that offer voice recording and network connectivity offer new possibilities such as supporting annotation at the moment the photo is captured. On the other hand, collaborative tagging has shown users are willing to make annotations when certain conditions are met. Further research is needed to investigate how the approaches we discuss in this article apply to those new settings.

Even though the importance of photo metadata is well known, the development of systems to support annotation has not been so successful. That is partly because of the limitations of automatic content analysis approaches, and partly because software designers have not understood in which situations users will be ready to spend a substantial amount of energy annotating photos. In this section, we have illustrated how novel interface designs might facilitate photo annotation by individuals or communities of users.

GUIs for using metadata in image management and exploration

Once richer image metadata is available either by automatic image content analysis or by manual annotation, users can make use of it for various image management tasks including not only search but also organization, meaning extraction, navigation, and distribution.

From our research experience with personal media management systems (Bederson, 2001; Bederson et al., 2002; Kang & Shneiderman, 2000, 2002, 2006; Khella & Bederson, 2004; Kules et al., 2004; Shneiderman et al., 2006; Shneiderman & Kang, 2000; Shneiderman et al., 2002; Suh & Bederson, 2007), we learned that managing personal media objects is a challenge for most users, who may struggle to understand, interpret, arrange, and use them. They wrestle with at least two major problems. First, most tools were designed and developed based on rigid and system-driven organizing metaphors (such as file-folder hierarchy). Second, those tools were not suitably designed to use metadata in exploring the media objects—beyond merely searching them. To address these challenges, we designed novel GUIs that hope to improve task performance as well as user satisfaction in exploring and managing personal media data.

3.1. Organization

With rich image metadata, even if users can find the images they need, they still frequently want to organize them for other reasons such as supporting future serendipitous browsing or to have the satisfaction of putting their images in order

(Balabanovic, Chu & Wolff, 2000; Kuchinsky et al., 1999; Rodden & Wood, 2003). Hence, supporting image organization is important.

One of the main challenges in designing a novel user interface for organizing images is how to let end-users represent and flexibly apply their conceptual models to their image collections. Users typically understand their data by constructing conceptual models in their minds. There is no unique or right model. Rather, the models are personal, have meaning for the individual who creates them, and are tied to specific tasks. Even in a simple personal photo library, images can be organized by time lines, locations, events, people, or other attributes, depending on users' conceptual models and specific tasks. Despite the diversity of users' conceptual models, the means available for users to organize and customize their information spaces are typically poor and driven mostly by storage and distribution models, not by users' needs.

Ben Shneiderman and Hyunmo Kang tried to resolve this challenge by providing users an environment to customize their information space appropriately for their conceptual models and specific tasks. We introduced a model called *Semantic Regions* (Kang & Shneiderman, 2006; see Figure 4), which are query regions drawn directly on a 2D information space. Users can specify the shapes, sizes, and positions of the regions and thus form the layout of the regions meaningful to them. Semantic Regions are spatially positioned and grouped on the 2D space based on personally defined clusters or well-known display representations such as a map, tree, time line, or organization chart. Once the regions are created and

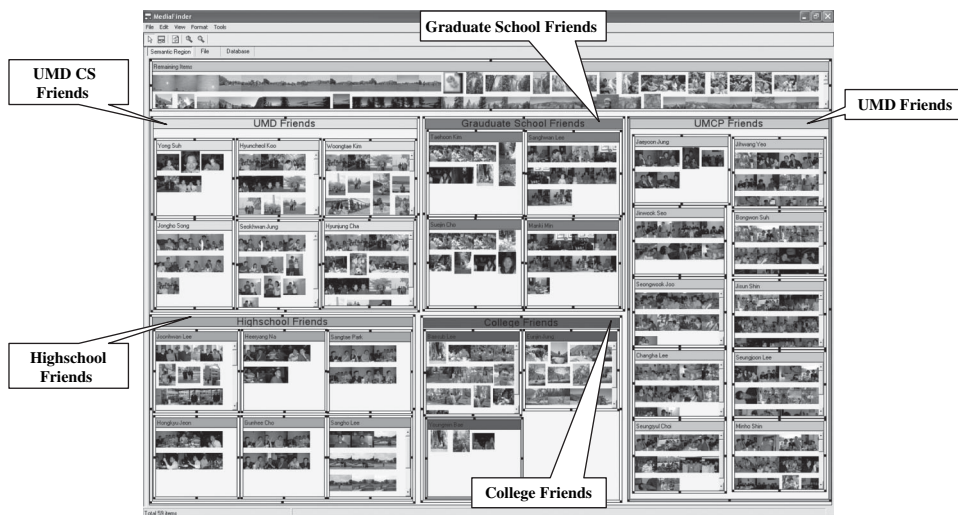


FIGURE 4 A friend group conceptual model: Each region represents a person and contains all the photos annotated with the name defined in it. *Note.* The regions are grouped into five clusters to represent different friend groups (UMD friends—University of Maryland computer science students-, high school friends, graduate school friends, college friends, and UMCP friends—University of Maryland noncomputer science students). Each group has its own color to represent the different group of friends.

arranged, users can specify semantics for each region using image metadata. The specified image metadata are conjunctively joined to form the semantics of a region.

In Figure 4, each region represents a person, and the regions are grouped into five clusters to represent different friend groups. When the images are dragged onto the constructed model, they are automatically placed in the appropriate regions based on the annotations. This metaphor is called *fling-and-flock*; that is, users fling the objects and the objects flock to the regions. If photos do not satisfy any of the semantics of regions, they are collected in the *remaining items region* located at the top of the panel.

Small usability studies of the Semantic Region approach were conducted to discover potential user interface improvements and to gain a deeper understanding about users' ability to understand, construct, and use semantic regions. The studies mainly focused on qualitative attributes such as user satisfaction rather than task performance. From these studies (Kang & Shneiderman, 2006), we learned that users liked the idea of spatial organization and dynamic grouping, but they were not too interested in constructing their own models, because it took too much time and effort. Users wanted rich built-in templates rather than user-defined templates or more automatic features to reduce the steps for region construction and semantic specification. Based on these studies, we simplified this approach by removing steps from the region construction and semantic specification while supporting automatic grouping and layout based on image metadata.

In a different project, we created an alternative and simplified approach for displaying groups of photos. This tool, called PhotoMesa, was developed as a successor to PhotoFinder by adding a zoomable user interface and the ability to show multiple groups of photos on the screen at the same time (Bederson, 2001; Bederson et al., 2002). In PhotoMesa, users can dynamically control how photos are grouped based on available metadata (Figure 5a). For example, photos can be grouped by the folder they belong to, by when the photos were taken (month or year), by people in the photo, or by any user-defined categories (locations, events, objects, etc.) that the photos are annotated with. The groups are automatically placed using quantum treemap space-filling algorithm (Bederson et al., 2002), and the groups can be sorted by photo metadata within a group such as number of photos in each group, starting date of photos in the group, title of the group, and so on. Each photo can be contained in multiple groups as long as it satisfies the grouping category.

Another approach created by Ben Shneiderman and Jack Kustanowitz is designed for groups of photos that have some relationship among them. In this case, it can be advantageous to show this relationship in the layout, rather than only in textual captions. Figure 5b shows a novel bilevel hierarchical layout, in which one region is designated for primary content, which can be a single photo, text, graphic, or combination (Kustanowitz & Shneiderman, 2005, 2006). Adjacent to that primary region, groups of photos are placed radially in an ordered fashion, such that the relationship of the single primary region to its many secondary regions is immediately apparent. A compelling aspect is the interactive experience in which the layout is dynamically resizable, allowing users to rapidly, incrementally, and reversibly alter the dimensions and content.



FIGURE 5 Organization of personal photo collections. (a) PhotoMesa interface enables users to dynamically group photos based on photo metadata. The group layout is automatically generated using space filling algorithm but they can also be sorted by the metadata of group photos. PhotoMesa enables users to control the visible photos in various ways. In this example, PhotoMesa shows only the representative photos of each group. (b) Bi-level hierarchical layout shows the photo groups taken in 6 locations during a trip by placing the groups clockwise in chronological order (left). (c) A real estate browser with 5 communities and the price ranges of selected homes is shown on the right.

3.2. Navigation and Browsing

Image browsing is important for a number of reasons. First, no matter what information retrieval system is being used, the user has to browse the results of the search. It is certainly important to build query systems that help users get results that are as close to what is wanted as possible. However, there will always be images that need to be browsed visually to make the final pick. The second reason is that sometimes people browse images just for the pleasure of looking at those images, and they often do it with other people. This is especially true for personal photos. Looking at home photos has a lot of overlap with traditional retrieval systems. People still want to be able to find photos of particular people, events, and so on. However, they are less likely to be time pressured to find a particular photo and more likely to be interested in serendipity (i.e., finding photos they weren't looking for; Balabanovic, Chu, & Wolff, 2000; Bederson, 2001; Rodden & Wood, 2003, Shneiderman et al., 2006).

We designed PhotoMesa to support this serendipitous style of use. It uses a zoomable user interface (Figure 6) to enable users to see all the images in a single view and concentrate on the images while browsing photos by zooming in or out without the need for interacting with other UI widgets such as scrollbar or pop-up window. This design is also based on the characteristics of personal image collections. As users have seen them at least once, they already have rough idea about their images and metadata. Hence, they often want to start browsing from the overview of their whole collection without worrying about missing some photos by accident (because they scrolled outside the currently visible region).

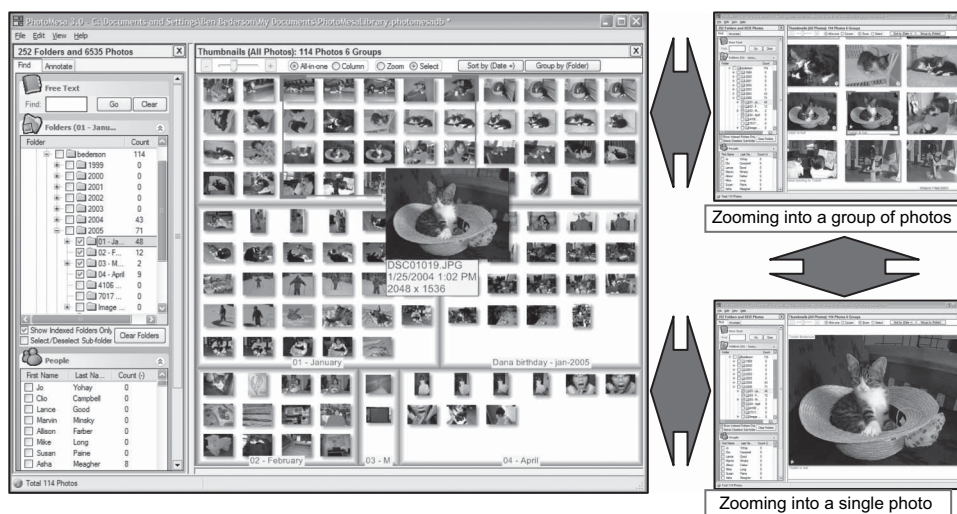


FIGURE 6 PhotoMesa shows all photos in a single view. *Note.* The layout of photos depends on how many groups and photos have been loaded. A larger preview of any photo can be obtained by moving the mouse over one of the small thumbnails. PhotoMesa uses a Zoomable User Interface and supports browsing of photos by zooming in or out. The order as well as the group that photos are displayed in can be controlled dynamically by photo metadata.

This zoomable browsing approach was further enhanced in terms of system performance and users' cognitive workload by supporting semantic zooming (Bederson et al., 1996). For example, if there are too many photos for users to recognize any of them, PhotoMesa has the option of displaying just representative photos (Figure 4a). *Time quilt* (Huynh, Drucker, Baudisch, & Wong, 2005; see Figure 7a) showed the improvement in photo browsing performance by making use of a linear timeline on top of the 2D space-filling techniques with semantic zooming. This layout is designed to convey a temporal order while making reasonable use of screen space by packing a time line layout into a rectangular screen space using a "line break" algorithm. Their user study shows that this combined layout enabled users to complete some photo browsing tasks faster than the quantum treemap layout and the time line layout. A new tool called PhotoSynth (Snavely, Seitz, & Szeliski, 2006; Figure 7b) uses 3D space for photo browsing. The system takes a large collection of photos of a place or an object, analyzes the photos for similarities, and displays them in a reconstructed 3D space so that users can interactively browse and explore large collections of photos of a scene using a 3D interface.

In addition to the design of novel browsing interfaces, we also developed a novel thumbnail generation algorithm that aims to reduce users' cognitive workload during browsing (Suh, Ling, Bederson, & Jacobs, 2003). Using thumbnails has a trade-off; bigger thumbnails use more screen space, but smaller thumbnails are hard to recognize. Automatic thumbnail cropping is a method to find the optimal part of an image for thumbnail generation. The algorithm finds the informative portion of images based on a saliency model and cuts out the nonsalient part of images (Figure 8). Thumbnail images generated from the cropped part of images increases users' recognition and helps users in visual search. On the other hand, the face detection based cropping technique (Suh et al., 2003) shows how semantic information can be used to enhance thumbnail generation further (when that semantic information about an image is known). For example, using face detection can more effectively create thumbnails with the images of people, which was demonstrated to significantly increase users' recognition and visual search performance.

3.3. Search and Meaning Extraction

There is no doubt that search is a hugely important photo management task. Users may be looking for photos of their grandmother, their trip to Disney World, or a friend's wedding. Interactive visualization to select and view dozens or hundreds of photos extracted from tens of thousands has become a popular strategy (Shneiderman et al., 2006). In addition to simple photo search based on annotation, users sometimes want to extract meaning from their photo collections such as, Who else appeared in the pictures of my grandmother and me over 20 years and when and where did I meet her? Who appeared both in my friend's wedding pictures and in my wife's friend's wedding pictures? How many photos did I take of my daughter each year? What were the special events since she was born? Any seasonal differences in photo distribution. This kind of meaning extraction from personal photo collections becomes

(a)



(b)



FIGURE 7 Photo browsing interfaces. (a) Time quilt—a layout designed to convey temporal order while making better use of screenspace than a timeline, showing approximately 5,500 photos with representative thumbnail overview. (b) PhotoSynth takes a large collection of photos of a place or an object, analyzes them for similarities, and displays them in a reconstructed three-dimensional space.

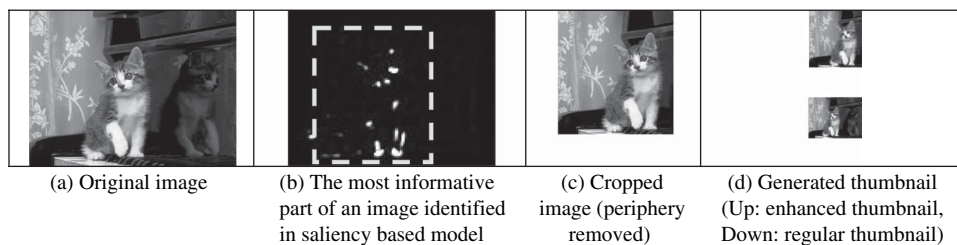


FIGURE 8 The saliency based model identifies the most informative part of an image. Thumbnail images created out of the core part help users' recognition and visual search performance.

more common and popular tasks as users have an increasing number of digital photos and richer metadata. However, little work has been done so far in this area.

A related effort in this area is the Semantic Region project (Kang & Shneiderman, 2006) described previously. It provides an interactive visualization technique called *region brushing* to support meaning extraction by visualizing the relationships among the semantic regions. Region brushing is used to highlight the personal media items contained in multiple regions simultaneously (Figure 9a) and is often used for depicting both the intrarelations and the interrelationships among models. Because a personal media item can be contained in multiple semantic regions, the existence as well as the amount of the shared personal media items among the regions represents their relationships. In Figure 9b, three models; HCI researcher, U.S. map, and CHI conference calendar are combined together. Because a photo can be contained in multiple regions across the different models; many questions concerning the interrelationships of models can be

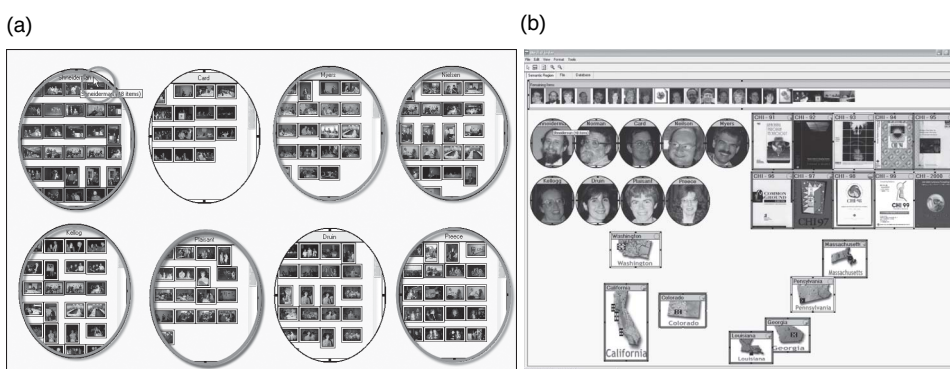


FIGURE 9 Region brushing: An interaction technique to visualize the shared items among the Semantic Regions. *Note.* (a) regions containing any of the photos in the region the mouse is over (Shneiderman's region) are highlighted. (b) visualization of interrelationships of the multiple models. By placing the mouse on Shneiderman's region, users can see which conference and which state his photos were taken.

answered through region brushing. Such questions might be, “Find the name of the conferences that all nine people participated in and where was the conference held?” “What was the name of the conference held in Atlanta, Georgia, and who did not appear in the photos taken at this conference?” “Find the conferences in which Ben Shneiderman did not appear in the photos and where was the conference held?” and so on. Furthermore, constructing new semantic regions recursively inside a region (e.g., location regions inside a person region) enables users to dynamically regroup the photos contained in a region and extract meanings.

Techniques such as *query preview* (Greene et al., 1999) or *faceted search* (Yee, Swearingen, Li, & Hearst, 2003) can help users extract meanings as well as rapidly filter down results sets by showing counts of available items along multiple dimensions or facets. PhotoMesa applies these approaches by letting users search for photos using the Find tab (left panel in Figure 10) in six ways: with keywords, by the folder a photo is in, by the people in a photo, by the categories a photo was annotated with, by the year a photo was taken, or by the month a photo was taken. If a new photo set is searched by combining those six ways, each search panel previews the distribution of the searched photos so that users can see how

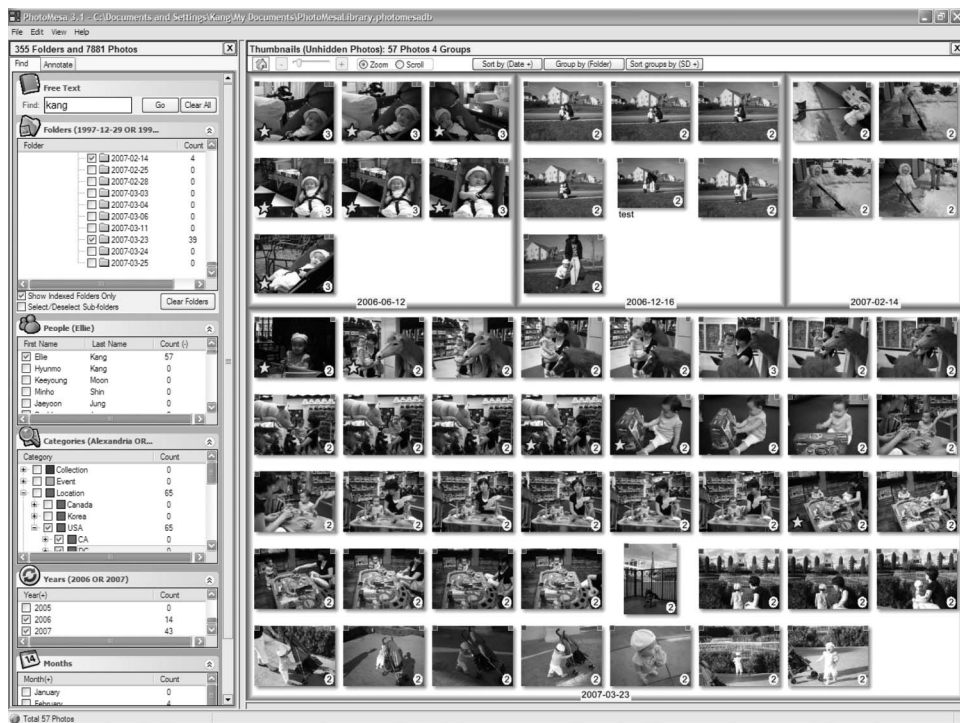


FIGURE 10 PhotoMesa search panel on the left enables users to search photos in six ways: with keywords; by the folder a photo is in; by the people in a photo; by the categories a photo was annotated with; and by the year or month a photo was taken. *Note.* The searched photos are displayed in the thumbnail viewer on the right and their distributions are shown in search panel.

many photos are from each folder, how many people (and who) appear in the searched photos, where the searched photos were taken (or how many photos were from other user-defined categories), and when they were taken (month and year). With this feature, users can get information about the searched photos and use it for further exploratory search.

3.4. *Sharing and Distribution*

Photo sharing is becoming one of the most common activities of digital photo management. Our earlier work provided interesting insights about how individuals and communities share their personal photographs. For example, a client-server model of PhotoFinder, PhotoFinder Kiosk (which was installed at CHI 2001), enabled members of a community of HCI professionals to enjoy, share, and annotate photos at an interactive public exhibit (Kules et al., 2004; Shneiderman et al., 2002). There is now a tremendous commercial activity in this area, with numerous popular photo sharing Web sites such as Flickr and Picasa Web Albums.

We previously mentioned some of the trade-offs of desktop applications and Web-based photo-sharing sites. However, the current bifurcation requires users to pick one approach or the other to manage their photos. We believe that this split is an artifact of business requirements and software development practices and that with a little more creativity and effort, it is possible to build a hybrid solution that bridges both worlds providing the advantages of each. For example, Sharpcast (<http://www.sharpcast.com>; Sharpcast) enables users to share, access, and organize their photos through either local file systems or Web sites. We also have heard of people using independent file synchronization tools, such as FolderShare (<http://www.foldershare.com>; Microsoft) to keep multiple copies of their photos synched with metadata between different machines, including Web servers. All this points to the necessity of a hybrid approach. From a user's perspective, it doesn't make any sense that they should have to choose in advance whether they should commit to storing photos locally or online.

To assess the usefulness of the hybrid approach, we built a prototype extension to PhotoMesa that hides the distinction between local and Web access from the user. Because photos almost always start locally, after being transferred from a camera to a computer, we assume users will open the photos regularly in their local tool. Then the tool continuously synchronizes the local photos with the Flickr Web site using its public API in the background. In this manner, all the user's photos are also available within their private Flickr account. Furthermore, any changes or annotations to the photos on Flickr are synched back to the desktop application the next time they are accessed locally. In addition, full searches of Flickr's remote database can be performed within PhotoMesa the same way local searches are made, and remote photos are shown grouped by whatever metadata Flickr makes available.

There is a largely unmet need of users to be freed from the shackles of tightly controlled applications that limit access to their photos and metadata. In the increasingly decentralized world of information, that iron-fisted approach is just not sustainable anymore.

4. CONCLUSION

The world of digital photo management has come a long way in the past few years, and we are pleased to see that the basic needs of most personal photo users are now being met. However we aren't done yet. There remain large gaps in meeting users' needs such as recording richer annotations that are more accurate and broadly available. It is crucial to support seamless sharing of photos and their stories in a way that lets multiple parties participate without losing ownership of the photos and metadata.

It is not clear how well current approaches will scale up. With the ever-shrinking cost of storage at the same time that cameras are becoming ubiquitous, we can expect that over an individual's life, it will be common to manage tens or even hundreds of thousands of photos.

The challenge of applying automated or human-assisted analysis techniques to these photos in a way that remains understandable and usable is ongoing. It also remains a challenge to find the right balance between local and Web access that enables people to have reliable long-term access to their full resolution photos in a way that does not lock them into any particular vendor's system. People must also be able to maintain their privacy and control over photos while sharing and collaboratively editing the photos and stories about them. When we envision the addition of audio and video to these collections, which is already happening, all of these challenges grow even further.

The transition to digital photography has already largely happened. This implies that consumers are satisfied enough with current solutions—or at least prefer digital to film. However that doesn't mean that their long-term problems are solved or that things are good as they could and should be. It is our job as designers and developers of these systems to address these challenges.

In this article, we described a range of projects that attempt to improve some of the problems already described. By integrating better annotation tools, visual presentation, and query systems, and through the use of more open metadata storage approaches, we have the opportunity to take the growing digital photo market and accelerate improvements for all of us to use with our own dear personal photos. Some directions not covered in this article, such as integrating annotations into other life activities (such as blogging, diary writing, e-mail, etc.) also offer promise. In the end, we expect a rich ecosystem of photo-related technologies will continue to evolve that support the diverse set of users that take photos.

REFERENCES

- Balabanovic, M., Chu, L. L., & Wolff, G. J. (2000). Storytelling with digital photographs. In *Proceedings of Human Factors in Computing Systems* (pp. 564–571). New York: ACM Press.
- Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7, pp. 3–31.
- Bederson, B. B. (2001). PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of ACM Conference on User Interface and Software Technology* (pp. 71–80). New York: ACM Press.

- Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21, 833–854.
- Chalfen, R. (1987). *Snapshot versions of life*. Bowling Green, OH: Bowling Green State University Popular Press.
- Chellappa, R., Wilson, C. L., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 705–740.
- Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Communication of ACM*, 30, 964–971.
- Goto, Y., Jung, J., Ma, K., & McCaslin, O. (2000). *The effect of direct annotation on speed and satisfaction*. Student HCI Online Research Experiment, Department of Computer Science, University of Maryland, College Park, MD. Retrieved from <http://www.otal.umd.edu/SHORE2000/annotation/>
- Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Olsen, L., Major, G., et al. (1999). The end of zero-hit queries: Query previews for NASA's Global Change Master Directory. *International Journal of Digital Libraries*, 2(2–3), 79–90.
- Huynh, D., Drucker, S., Baudisch, P., & Wong, C. (2005). Time quilt: Scaling up zoomable photo browsers for large, unstructured photo collections. *Extended Abstracts of Human Factors in Computing Systems*, 1937–1940.
- Japan Electronic and Information Technology Industries Association. (2002). Exchangeable image file format for digital still cameras: Exif Version 2.2. Retrieved from <http://www.exif.org>
- Jung, J. (2000). Empirical comparison of four accelerators for direct annotation of photos. Retrieved from <http://www.cs.umd.edu/hcil/photolib/paper/cmsc498paper.doc>
- Kang, H., Plaisant, C., & Shneiderman, B. (2003). New approaches to help users get started with visual interfaces: Multi-layered interfaces and integrated initial guidance. *Proceedings of Digital Government Conference*, 1–6.
- Kang, H., & Shneiderman, B. (2000). Visualization methods for personal photo collections: Browsing and searching in the PhotoFinder. *Proceedings of IEEE International Conference on Multimedia and Expo*, 1539–1542.
- Kang, H., & Shneiderman, B. (2002). Dynamic layout management in a multimedia bulletin board. *Proceeding of IEEE International Symposium on Human-Centric Computing Languages and Environments*, 51–53.
- Kang, H., & Shneiderman, B. (2006). Exploring personal media: A spatial interface supporting user-defined semantic regions. *Journal of Visual Language and Computing*, 17, 254–283.
- Khella, A., & Bederson, B. B. (2004). Pocket PhotoMesa: A zoomable image browser for PDAs. In *Proceedings of Mobile and Ubiquitous Multimedia* (pp. 19–24). New York: ACM Press.
- Kuchinsky, A., Pering, C., Creech, M. L., Freeze, D., Serra, B., & Gwizdka, J. (1999). FotoFile: A consumer multimedia organization and retrieval system. *Proceedings of ACM Conference on Human Factors in Computing Systems*, 496–503.
- Kules, B., Kang, H., Plaisant, P., Rose, A., & Shneiderman, B. (2004). Immediate usability: A case study of public access design for a community photo library. *Journal of Interacting With Computers*, 16, 1171–1193.
- Kustanowitz, J., & Shneiderman, B. (2005). Meaningful presentations of photo libraries: Rationale and applications of bi-level radial quantum layouts. In *Proceedings of Joint Conference on Digital Libraries* (pp. 188–196). New York: ACM Press.
- Kustanowitz, J., & Shneiderman, B. (2006). Hierarchical layouts for photo libraries. *IEEE MultiMedia*, 13, 62–72.
- Lieberman, H., & Liu, H. (2002). Adaptive linking between text and photos using common sense reasoning. *Proceedings of Adaptive Hypermedia and Adaptive Web Systems*, 2–11.

- Naaman, M. (2005). Leveraging geo-referenced digital Photographs. Stanford University, Stanford, CA.
- Naaman, M., Yeh, R. B., Garcia-Molina, H., & Paepcke, A. (2005). Leveraging context to resolve identity in photo albums. *Proceedings of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries*, 178–187.
- Phillips, P. J., Grother, P. J., Michaels, R. J., Blackburn, D. M., Tabassi, E., & Bone, J. M. (2003). Face recognition vendor test 2002: Evaluation report, NISTIR 6965. Retrieved from <http://www.frvt.org/FRVT2002/default.htm>
- Rodden, K., & Wood, K. (2003). How do people manage their digital photographs? *Proceedings of ACM Conference on Human Factors in Computing Systems*, 409–416.
- Sears, A., & Shneiderman, B. (1994). Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, 1, 27–51.
- Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16(8), 57–69.
- Shneiderman, B. (2003). Promoting universal usability with multi-layer interface design. In *ACM Conference on Universal Usability* (pp. 1–8). New York: ACM Press.
- Shneiderman, B., Bederson, B. B., & Drucker, S. M. (2006). Find that photo!: Interface strategies to annotate, browse, and share. *Communications of the ACM*, 49, 69–71.
- Shneiderman, B., & Kang, H. (2000). Direct annotation: A drag-and-drop strategy for labeling photos. *Proceedings of International Conference Information Visualization (IV2000)*, 88–95.
- Shneiderman, B., Kang, H., Kules, B., Plaisant, C., Rose, A., & Ruchir, R. (2002). A photo history of SIGCHI: Evolution of design from personal to public. *ACM Interactions*, 9(3), 17–23.
- Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3), 835–846.
- Suh, B., & Bederson, B. B. (2007). Semi-automatic image annotation strategies using event based clustering and clothing based person recognition. *Journal of Interacting with Computers* 19(4), 524–544.
- Suh, B., Ling, H., Bederson, B. B., & Jacobs, D. (2003). Automatic thumbnail cropping and its effectiveness. *Proceedings of ACM Conference on User Interface and Software Technology*, 95–104.
- Von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of ACM Conference on Human Factors in Computing Systems* (pp. 319–326). New York: ACM Press.
- Yang, M., Kriegman, D., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 34–58.
- Yee, K-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proceedings of ACM Conference on Human Factors in Computing Systems* (pp. 401–408). New York: ACM Press.
- Yoshitaka, A., & Ichikawa, T. (1999). A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 81–93.
- Zhao, W., Chellappa, R., Philips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4), 399–458.

Capture, Annotate, Browse, Find, Share:

Novel Interfaces for Personal Photo Management

Hyunmo Kang¹, Benjamin B. Bederson^{1,2}

Human-Computer Interaction Laboratory

University of Maryland Institute

for Advanced Computer Studies¹

Computer Science Department²

College Park, MD 20742

{kang, bederson}@cs.umd.edu

Bongwon Suh

Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

suh@parc.com

Abstract

The vision of ubiquitous digital photos has arrived. Yet, despite their broad popularity, significant shortfalls remain in the tools used to manage them. We believe that with a bit more creativity and effort, the photo industry can solve many of these problems, offering tools which better support accurate, rapid, and safe shared annotations with comfortable and efficient browsing and search. In this article, we review a number of projects of ours and others which relate to Ben Shneiderman's work on interfaces for photo management. With these, we describe the problems that we see in existing tools and our vision for improving them.

1. Introduction

The days when people debated the relative merits of film vs. digital imagery now seem almost quaint. And with hindsight, film seems destined to have been a blip in history along with LP vinyl records – a temporary physical analog recording medium. While some aficionados may still prefer some qualities of those dust-gathering mediums, the advantages of digital media have become clear. The rapid and inexpensive ability to edit, annotate, search, share and access has brought digital media to ubiquity.

And yet, with all this promise, shortfalls remain in the overall user experience. How many of us have replaced old unlooked at shoeboxes of prints with unlooked at digital archives of image files? How much has our ability to find a particular set of photos really improved (i.e., can you find those photos of a visiting uncle with your sister when they were children?) How do we record stories from our parents describing family photos? And how do we make sure those stories stay with the photo in question and get distributed to all copies of that photo within the family? And perhaps of most concern, how do we insure that these annotations stand the test of time and remain accessible as computers, file formats, recording mediums, and software change?

These changes are all happening within the context of human behavior, which does not change so rapidly. People still like immediate gratification, take pictures rapidly, print them, and share in social settings. Some people spend a lot of effort creating photo albums or “scrapbooking”. And of course, many do not. Understanding which behaviors

are fundamental and which are side effects of current technology is crucial because this understanding can and should influence where researchers spend their effort.

We explore these issues and more with much of the intellectual motivation coming from Ben Shneiderman, our close colleague who has pushed for a deeper understanding of and better support of photo management for over 10 years. His personal photo archives document the field of HCI going back to its beginning. He regularly shares those photos with great enthusiasm to visitors, motivating and exciting all of us – largely because of the care and consistency he has applied to annotating and organizing his photos. He regularly pulls up old photos of lab visitors showing everyone what they worked on 5 or 15 years ago (and of course, showing what they looked like too!). His early exploration of tools to support photo management (with co-author Kang) led to PhotoFinder [Kang and Shneiderman, 2000], and the ensuing PhotoMesa tools [Bederson, 2001; Bederson et al., 2002]. His personal interest helped inspire the authors of this article as well as other lab members (particularly Catherine Plaisant) to pursue the development of approaches and software to improve all of our user experiences when managing photos.

This, of course, all happened during a time of tremendous commercial activity in this area. There are wildly popular photo sharing websites, such as Flickr¹, Picasa Web Albums², Snapfish³, Shutterfly⁴, and PhotoBucket⁵ as well as equally well-used desktop photo

¹ Yahoo! Inc. www.flickr.com

² Google Inc. picasaweb.google.com

³ HP Inc., www.snapfish.com

⁴ Shutterfly Inc., www.shutterfly.com

⁵ Photobucket Inc. www.photobucket.com

applications such as Google Picasa⁶, Adobe PhotoShop Album⁷, and ACDSee. The two approaches (desktop application and website) are interesting to look at because they each offer distinct advantages to users. For example, websites are available anywhere and facilitate sharing, while desktop applications are faster, support higher resolution photos more easily, provide local ownership of photos, and offer richer interaction capabilities. Interestingly, each approach is gaining characteristics of the other. Web applications begin to offer dynamic, interactive content rather than static html pages through AJAX and Flash technologies. In addition, they often include plugins to ease uploading or improve performance and some offer APIs to enable desktop applications to access their data directly. At the same time, many desktop applications are offering web capabilities such as sharing.

Yet even with this commercial activity, the full potential of personal photo management has not been reached. There is the opportunity for richer annotation interfaces, automated content analysis, improved sharing and more creative organizational strategies. Our hope is that more photos end up with better metadata, enabling faster, easier and more accurate and enjoyable retrieval and use.

In this article, we look at some of the key activities and behavior patterns of personal photo users and examine how innovative user interfaces have the potential to enhance users' power, satisfaction, and control in managing and exploring their images. Starting with a close look at annotation, we examine how a combination of manual and automated

⁶ Google Inc. www.picasa.com

⁷ Adobe Systems Inc. www.adobe.com/products/photoshopalbum/starter.html

techniques can improve how people associate metadata with photos. We then look at how the resulting richer metadata can enable better interfaces for searching and browsing photos. Finally, we end with a discussion about the importance of sharing photos and how new interfaces enable that.

2. GUIs for Annotation

An essential question is how valuable photo metadata is. Our own assessment of user needs [Shneiderman and Kang, 2000] coupled with reports from other researchers [Chalfen, 1987; Naaman, 2005; Rodden and Wood, 2003], and our personal experience come together on this. They indicate that the photo metadata such as dates, locations, and content of the photos (especially people's names) play a crucial role in management and retrieval of personal photo collections.

However, in spite of the high utility of the photo metadata, the usability of software for acquiring and managing the metadata has been poor. The manual photo annotation techniques typically supported by photo software are time-consuming, tedious and error-prone, and users typically put little effort into annotating their photos. In fact, the industry attitude tends to be that since users do not annotate their photos very much, it is not necessary to spend much energy adding good support for it. However the success of keyword labeling, so called *tagging*, systems such as del.icio.us **Error! Bookmark not defined.** and Flickr **Error! Bookmark not defined.** hints that users indeed want to make annotations when reasonable utility and usability are supported. We believe that the photo annotation has enough utility for some users and it is the usability of software that needs to be improved.

In this section, a few innovative approaches will be presented to show how interaction and GUI design can improve the usability of the photo annotation process when they are

based on careful understanding of users' behavior and usage pattern. In addition, we will explain how those designs have been evolving over time to support a broader range of annotation tasks by combining the accessible technologies with the analysis of users and their needs.

2.1 Advanced Manual Annotation: Direct Annotation

Since annotations based on automated image content analysis are still limited, we developed an advanced manual annotation mechanism that can significantly reduce users' annotation workload under certain circumstances. From the observations of personal photo annotations [Shneiderman and Kang, 2000], we found that there were three interesting characteristics that could be useful for our interaction design.

- Personal photo libraries often contain many images of the same people at different events. In the libraries we looked at, we typically found 100-200 identifiable people in several thousand photos. Furthermore, the library has a highly skewed distribution with immediate family members and close friends appearing very frequently.
- Textual search often doesn't work reliably because of inconsistency in names with misspellings or variants (e.g. Bill, Billy, William).
- Lists of names of people appearing in photos are often difficult to associate with individuals, especially in group shots. Textual captions often indicate left-to-right

ordering in front and back rows, or give even more specific identification of who is where. However this approach is tedious and error-prone.

Based on these observations, we collaborated with Ben Shneiderman to develop the concept of *direct annotation*⁸ which is a technique using selectable, draggable labels that can be placed directly on the photo [Shneiderman and Kang, 2000]. Similar interfaces have also appeared recently on Flickr and MySpace. Users can select from a scrolling or pop-up list and drag by mouse or touch screen. This applies direct manipulation principles [Shneiderman, 1983, 2005] that avoid the use of a keyboard, except to enter a name the first time it appears. The name labels can be moved or hidden, and their presence is recorded in the database or in the header of an image file in a resolution-independent manner. The relative location of the target is stored based on an origin in the upper left hand corner of the photo with the point in the range (0, 0) – (1.0, 1.0) corresponding to the full image. This approach not only associates a name with a position on the photo, but also ensures that each name is always spelled the same way.

This simple rapid process also enables users to annotate at any time. They can add annotations when they first see their photos on the screen, when they review them and make selections, or when they are showing them to others. This design, which supports continuous annotation encouraged users to do more annotation, especially in collaborative situations such as with PhotoFinder Kiosk [Kules et al., 2004, Shneiderman et al., 2002]. In a public setting, visitors of PhotoFinder Kiosk added 1,335 name labels using direct annotation while adding 399 captions using traditional type-in method.

⁸ US Patent #7010751

The direct annotation mechanism was revised later so that it enables users to define their own categories such as activities, events, locations, objects in a photo in a hierarchical way (Figure 1) in addition to person names. A few alternative and complementary direct annotation mechanisms such as split menu annotation, hotkey annotation, popup menu annotation have also been designed and developed to accelerate the annotation process. These are described in more detail in the following subsection.

An informal experiment was conducted to see if the direct annotation method improved the annotation process in terms of annotation time and users' subjective preference compared with the traditional caption method or the click-and-type method [Goto et al, 2000][Jung 2000]. 48 volunteers participated in the experiment and a within-subject design was used, whereby each subject attempted an annotation task (20 names in five photos) on each system. While no significant difference was found in the mean annotation time, the direct annotation method was significantly preferred. In addition, there was a trend towards faster annotations with direct annotation. We believe that a more formal follow-up user study is required to identify under what circumstances (e.g. number of total labels, average number of people in a picture, pattern of people's appearance in a personal photo collection, and so on) the direct annotation mechanism works better than other annotation mechanisms in terms of several dependent variables such as completion time, error rate, users' satisfaction and confidence.

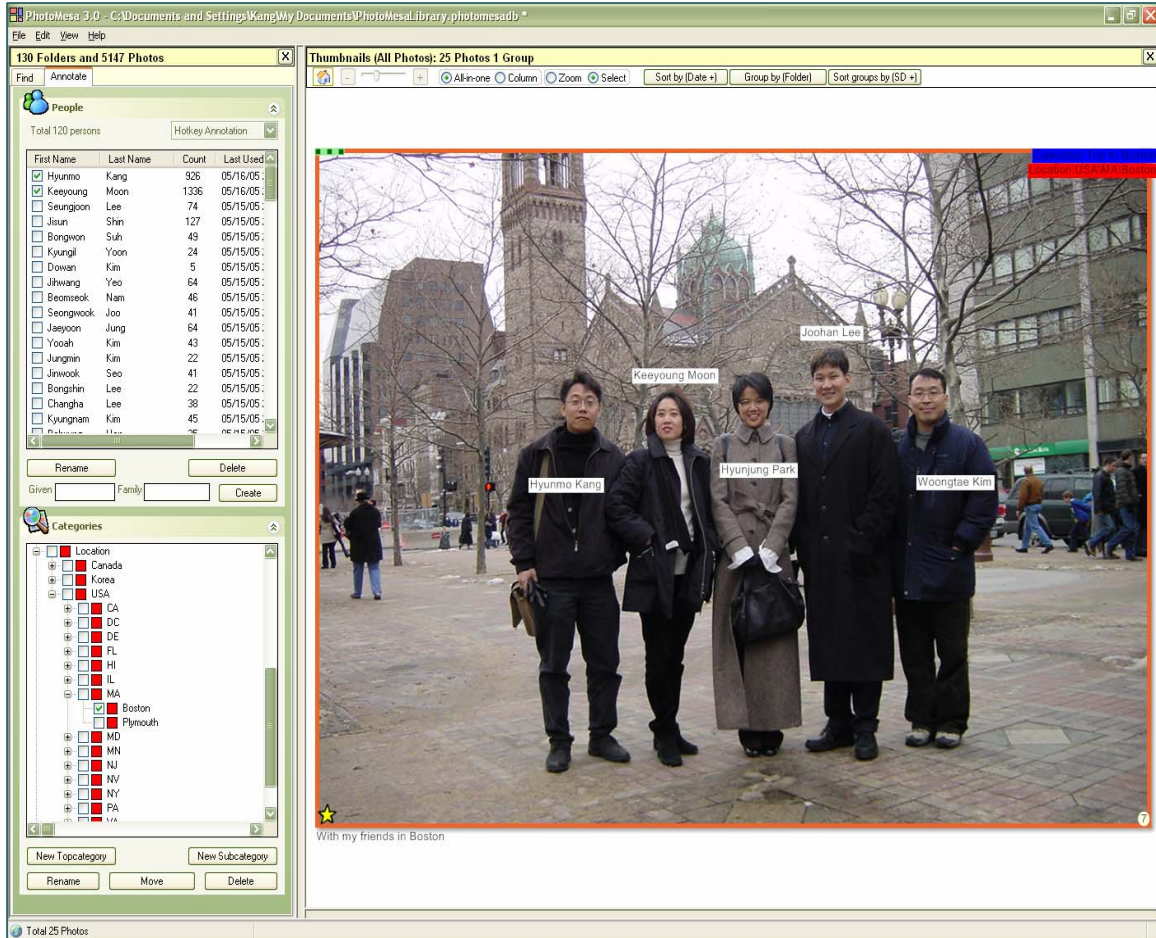


Figure 1: The revised direct annotation mechanisms as implemented in PhotoMesa, the successor to PhotoFinder. Users can add a caption under a photo, or add a particular attribute such as favorite (a yellow star on the bottom left of photo) or hidden. In addition, labels can be dragged from the list of people (on the top left) or from the user-defined category tree (on the bottom left). A label can be directly placed on the photo to represent where the individuals or objects are located within the photo.

2.2 Enhanced Direct Annotation: Bulk Annotation

The direct annotation mechanism was enhanced through a series of design cycles to support more efficient and productive annotation. Perhaps the most notable one was applying the "bulk annotation" technique, which lets users annotate several photos with a single interaction [Kuchinsky et al., 1999]. Bulk annotation is especially useful when users have a set of photos with the same person or group of people involved in the same

events. We thus designed and developed several variations of direct annotation and bulk annotation mechanisms to accelerate the annotation process.

The split menu annotation mechanism (Figure 2(a)) was designed to minimize users' time of scrolling the list to find correct labels to be used for photo annotation. The split menu featured a resizable splitter bar so that the number of the most frequent labels displayed in the top window. The scrollbar was removed from the top window, while the bottom half retained its scrollbar. The split menu raises interesting questions about what kind of automatic algorithms should be used to predict users' future access and facilitate rapid annotation.

Because some annotations get used so much more frequently than others, we designed the interface so that each label (either person's name or categories) can be associated with a "hot key" (Figure 2(b)). After the key is assigned to a particular label, users can press that key whenever the mouse is over a photo. At that point, the spot that the cursor was over will be annotated with the specified label. When the expert user has a good idea about who or what is to be annotated, then the hotkeys can be put to work very efficiently. Instead of having to find the name desired and drag the name over to the position, the user can simply position the mouse and press the hotkey. This method is especially useful when a photo collection has only a few names that appear frequently.

Popup menu annotation (Figure 2(c)) also aims at reducing mouse movement in selecting labels to be used for annotation. This method offers a menu when the right mouse button is pressed on a picture. The menu consists of the currently selected labels in the list so

that users can annotate pictures with a label in the menu at the position the right mouse button was pressed. This mechanism was revised again later as “Label Paint Annotation” so that a photo can be annotated with the currently selected labels whenever users click on a photo without selecting a popup menu.

In addition to the proposed methods for improving the speed in annotating individual photos, we also designed two additional methods for improving the efficiency of bulk annotation as follows:

- **Checkbox Annotation:** Select one or more photos and click the check box next to the label in the list (Figure 1).
- **Drag-drop Annotation:** Select one or more photos and drag and drop the selected labels onto the photo. All the photos that were selected get annotated with the labels.

Another user study [Jung, 2000] was conducted to examine the differences in annotation performance and users’ satisfaction among the enhanced direct annotation methods. The results suggest that while there are some slight differences among the direct annotation methods, for the most part there is not enough of a significant distinguishing factor to proclaim one as the most efficient or the most rewarding. However, a further study of expert users may be necessary to verify this and to determine if there are conditions under which some approaches are more valuable.

If the methods aren’t significant, then the best option may be to include as many methods as possible while allowing a variety of options to let the users customize the methods.

However it raises the question of how much the user can learn at first; if presented by too many options in the beginning, the user may become confused and/or frustrated.

Therefore it may be optimal to use the multi-layered approach [Kang et al, 2003] when designing the initial interface.



(a) Split menu annotation



(b) Hotkey annotation



(c) Popup menu annotation

Figure 2 Enhanced Direct Annotation Approaches within PhotoFinder

2.3 Semi-Automatic Annotation

The performance of direct annotation may be improved by supporting bulk annotation as described in the previous section. However this approach introduces two new challenges. First, unless photos to be annotated with the same labels are clustered together in some way so that they can be selected together, users need to manually select photos one by one before applying bulk annotation. Second, since multiple photos are annotated with a single interaction, the position of the labels in each photo cannot be explicitly specified by users.

To cope with these challenges, we explored a semi-automatic annotation strategy that takes advantage of human and computer strengths [Suh and Bederson, 2007]. The semi-automatic approach enables users to efficiently update automatically obtained metadata interactively and incrementally. Even though automatically identified metadata are compromised with inaccurate recognition errors, the process of correcting inaccurate information can be faster and easier than manually adding new metadata from scratch. In order to facilitate efficient bulk annotation, two clustering algorithms were introduced to generate meaningful photo clusters [Suh and Bederson, 2007]: *Hierarchical event clustering* and *Clothing based person recognition*. The first method clusters photos based on their timestamps so that each group of photos corresponds to a meaningful event. The hierarchical event clustering identifies multiple event levels in personal photo collection and allows users to choose the right event granularity. The second method uses a clothing-based human model to group similar-looking people together. The clothing-

based human model is based on the assumption that people who wear similar clothing and appear in photos taken within relatively short periods of time are very likely to be the same person.

To explore our semi-automatic strategies, we designed and implemented a prototype called SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface). The prototype provides an annotation framework focusing on making bulk annotations on automatically identified photo groups. The two automatic clustering algorithms provide meaningful clusters to users, and make annotation more usable than relying solely on vision techniques such as face recognition or human identification in photos. In addition, the clothing based person recognition automatically detects the positions of people in a photo, which can be used to position annotations. Interestingly, while we found that absolute performance using this approach wasn't much better than manual annotation, the user study showed that the semi automatic annotation was significantly preferred over manual annotation [Suh and Bederson, 2007]. And as computer vision techniques improve, the relative advantage of semi-automated techniques can only improve.

Figure 3 shows a prototype of SAPHARI which shows the clusters of identified people whose upper bodies are cropped from photos of the same event. Since the automatic clustering techniques can also have errors, we designed the GUI so that users can correct errors manually by dragging a photo into the correct group. After error correction, users can annotate multiple photos at once (also with position information) by dragging a label onto the group.

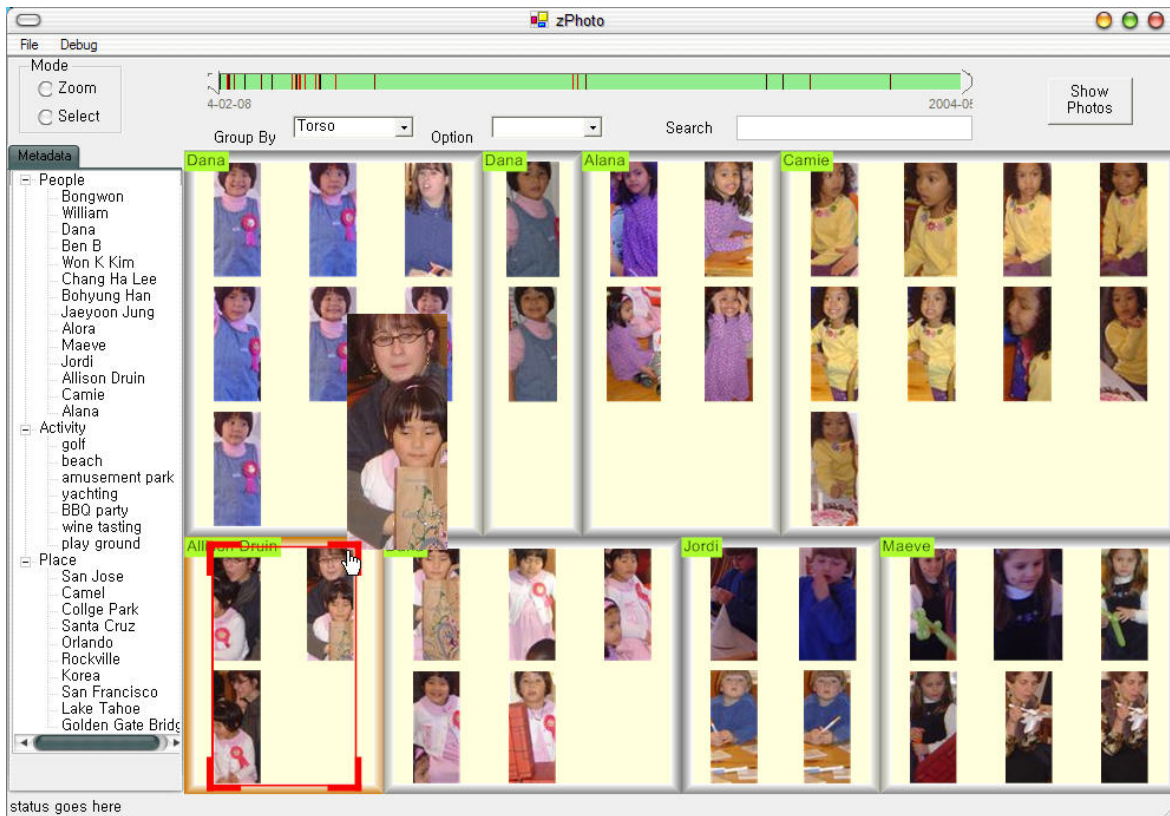


Figure 3: Identified people who are cropped from photos are laid out on the screen in the SAPHARI prototype. Similar looking people are clustered based on their clothing.

2.4 Image Annotation Discussion

There has been significant research on how to acquire useful metadata for images.

Many researchers have focused on automatic extraction of metadata by understanding images. For example, researchers has focused on automatic object detection such as face recognition, and content-based categorization and retrieval [Yang et al. 2002][Yoshitaka and Ichikawa 1999][Zhao et al. 2003]. However, such automatic techniques have achieved limited success so far when applied to personal photo management [Phillips et al. 2003] [Suh and Bederson 2007].

Rather than pure image-based approaches, some researchers used related context information to identify relevant metadata. For example, Naaman et al. [2005] used time and location information to generate label suggestions to identify the people in photos. Lieberman and Liu [2002] used relationships between text and photographs to semi-automatically annotate pictures.

On the other hand, web-based collaborative approaches to collecting metadata become popular. Web pages, online photographs, and Web links have been actively annotated with rich metadata through *tagging* (e.g. Flickr and Del.icio.us). Tagging is really just a simple kind of annotation where keywords can be associated with objects such as photos. However, the simple underlying approach belies the richness and power gained by broad communal access to these tags which enable cross-collection aggregation and searching along with innovative interfaces for creating and presenting those tags (i.e., tag clouds). With Flickr, a user can share personal photos with his/her friends, family and colleagues allowing the invited users to make annotations on shared photos. When users select photos and make them available to others, they seem to be willing to invest more effort in annotation [Shneiderman et al., 2006]. Also, by making them public, they invite others to comment and add annotations. We believe that this “folksonomy” based collaborative annotation has great potential for creating more useful metadata.

On the other hand, folksonomy based tagging system has a set of limitations. Folksomic tagging can be inconsistent because of its vocabulary problem [Furnas et al. 1987]. In folksonomy based systems, there is no bona fide standard for selecting a tag and users can choose to use any word as a tag. For example, one can use a tag, TV, while others

choose to use a tag, television. Furthermore, Furnas et al. [1987] showed that it is not always easy for one to come up with good descriptive keywords that can be shared. Tags, therefore, can be easily idiosyncratic and often causes meta-noise, which decreases the retrieval efficiency of systems.

Going further with collaborative annotation approaches, Von Ahn addressed the challenge by adding gaming to the mix. ESP Game [Von Ahn et al., 2006] combines a leisure game with practical goals. It lets people play an image guessing game that gathers image metadata as a side effect of them playing.

Another ongoing challenge for metadata management is where to store the metadata. It is crucial that users own their metadata just as much as they own their photos. Yet some companies (such as Apple with their iPhoto software) create a custom database that separates metadata from the photos leaving no easy way to get the annotations back for sharing or for use in other programs.

Software developers tend to like centralized metadata stores because they make it easier to write software that perform fast searches (and because they “lock in” customers to their products). However this approach is not necessary. Instead, we suggest storing all annotations and metadata using the “IPTC” standard format⁹ in the “EXIF” header [JEITA, 2002] of JPEG images. Then the application can create a cached copy of the metadata for efficient operations while leaving the “original” metadata with the photo. This is how we implemented PhotoMesa, and appears to be the same approach taken by

⁹ International Press Telecommunications Council <http://www.iptc.org>

some other commercial software such as Picasa and ACDSee¹⁰. However, even this approach has some challenges due to the following limitations:

- Not every image format supports metadata fields in the file headers.
- Metadata standards (i.e., EXIF/IPTC) do not have rich enough fields for objects such as people and hierarchical categories.
- Some operating systems make it difficult and possibly dangerous to modify the image header without changing the file modification date (which users sometimes use to determine when a file was created).

While the issues and challenges relating to metadata standards are significant, they are beyond the scope of this article. We thus raised these few issues to point out the importance of getting it right since it has such a dramatic affect on user experience. We therefore call on the industry to collaborate more openly in developing consistent and rich photo metadata standards addressing at least the above issues.

Technological change in society is rapid. There are new opportunities that we do not explore in this article. For example, the growing number of cell phones and even dedicated cameras that offer voice recording and network connectivity offer new possibilities such as supporting annotation at the moment the photo is captured. On the other hand, collaborative tagging has shown users are willing to make annotations when certain conditions are met. Further research is needed to investigate how the approaches we discuss in this article apply to those new settings.

¹⁰ ACD Systems <http://www.acdsee.com>

Even though the importance of photo metadata is well known, the development of systems to support annotation has not been so successful. That is partly because there have been many technical difficulties in automatic content analysis approaches, and partly because software designers have not always understood that many users are in fact willing to spend a substantial amount of energy annotating photos. In this section, we have demonstrated a few examples that show how novel interface designs based on a careful understanding of users can facilitate photo annotation and encourage users to annotate photos personally or collaboratively. In addition, we have explored issues that should be considered and resolved in order to encourage more researchers and developers to put their efforts in designing innovative GUIs to facilitate the annotation process for users.

3. GUIs for Using Metadata in Image Management and Exploration

Once richer image metadata is available either by automatic image content analysis or by manual annotation, users can make use of it for various image management tasks including not only search, but also organization, meaning extraction, navigation, and distribution.

From our research experience with personal media management systems [Bederson, 2001; Bederson et al., 2002; Kang and Shneiderman, 2000, 2002, 2006; Khella and Bederson, 2004; Kules et al., 2004; Shneiderman and Kang, 2000; Shneiderman et al., 2002; Shneiderman et al., 2006; Suh and Bederson, 2007], we learned that managing personal media objects is a challenge for most users, who may struggle to understand, interpret, arrange, and use them. They wrestle with at least two major problems. First, most tools were designed and developed based on rigid and system-driven organizing metaphors (such as file-folder hierarchy). Second, those tools were not suitably designed to use metadata in exploring the media objects except search.

To address these challenges, we have designed a number of novel GUIs that can improve task performance as well as user satisfaction in exploring and managing personal media data. In this section, we explain how these approaches can help image management tasks and how we tried to deal with those challenges.

3.1 Organization

With rich image metadata, even if users can always find the images they need, they still frequently want to organize them for other reasons such as supporting future serendipitous browsing and to provide the satisfaction of putting their images in order [Balabanovic et al., 2002; Kuchinsky et al., 1999; Rodden and Wood, 2003]. Hence, image organization is one of the most common and important image management tasks for users.

One of the main challenges in designing a novel user interface for organizing images is how to let end-users represent and flexibly apply their conceptual models to their image collections. Users typically understand their data by constructing conceptual models in their minds. There is no unique or right model. Rather, the models are personal, have meaning for the individual who creates them, and are tied to specific tasks. Even in a simple personal photo library, images can be organized by timelines, locations, events, people, or other attributes, depending on users' conceptual models and specific tasks. Despite the diversity of users' conceptual models, the means available for users to organize and customize their information spaces are typically poor and driven mostly by storage and distribution models, not by users' needs.

Ben Shneiderman and the first author of this article tried to resolve this challenge by providing users an environment to customize their information space appropriately for their conceptual models and specific tasks. We introduced a model called *Semantic*

Regions [Kang and Shneiderman 2006] (Figure 4), which are query regions drawn directly on a two-dimensional information space. Users can specify the shapes, sizes, and positions of the regions and thus form the layout of the regions meaningful to them. *Semantic Regions* are spatially positioned and grouped on the 2D space based on personally defined clusters or well known display representations such as a map, tree, timeline, or organization chart. Once the regions are created and arranged, users can specify semantics for each region using image metadata. The specified image metadata are conjunctively joined to form the semantics of a region.

In Figure 4, each region represents a person, and the regions are grouped into 5 clusters to represent different friend groups. When the images are dragged onto the constructed model, they are automatically placed in the appropriate regions based on the annotations. This metaphor is called *fling-and-flock*; that is, users fling the objects and the objects flock to the regions. If photos do not satisfy any of the semantics of regions, they are collected in the *remaining items region* located at the top of the panel.

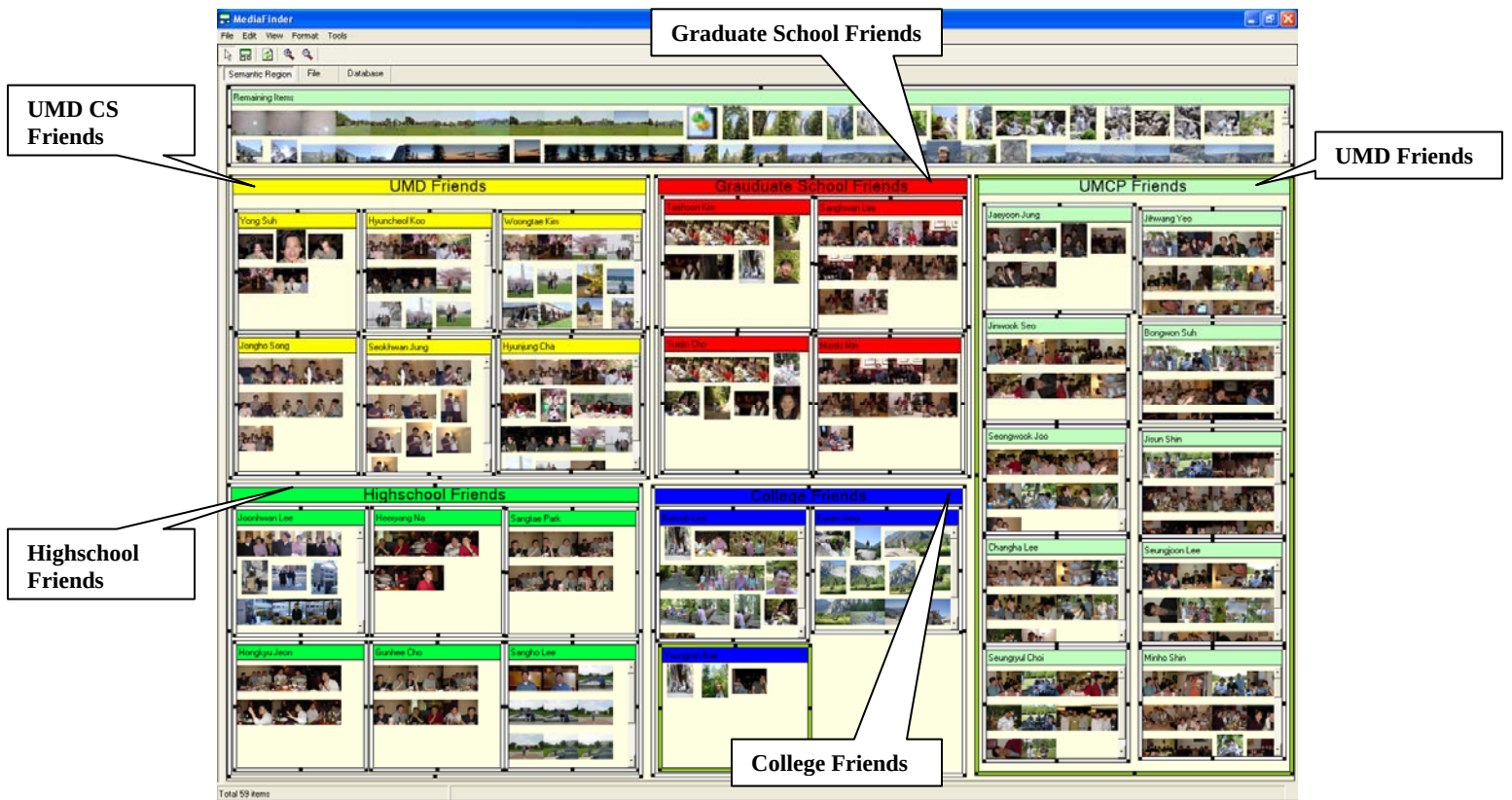


Figure 4: A friend group conceptual model: Each region represents a person and contains all the photos annotated with the name defined in it. The regions are grouped into 5 clusters to represent different friend groups (UMD friends -University of Maryland computer science students-, high school friends, graduate school friends, college friends, and UMCP friends-University of Maryland non-computer science students-). Each group has its own color to represent the different group of friends.

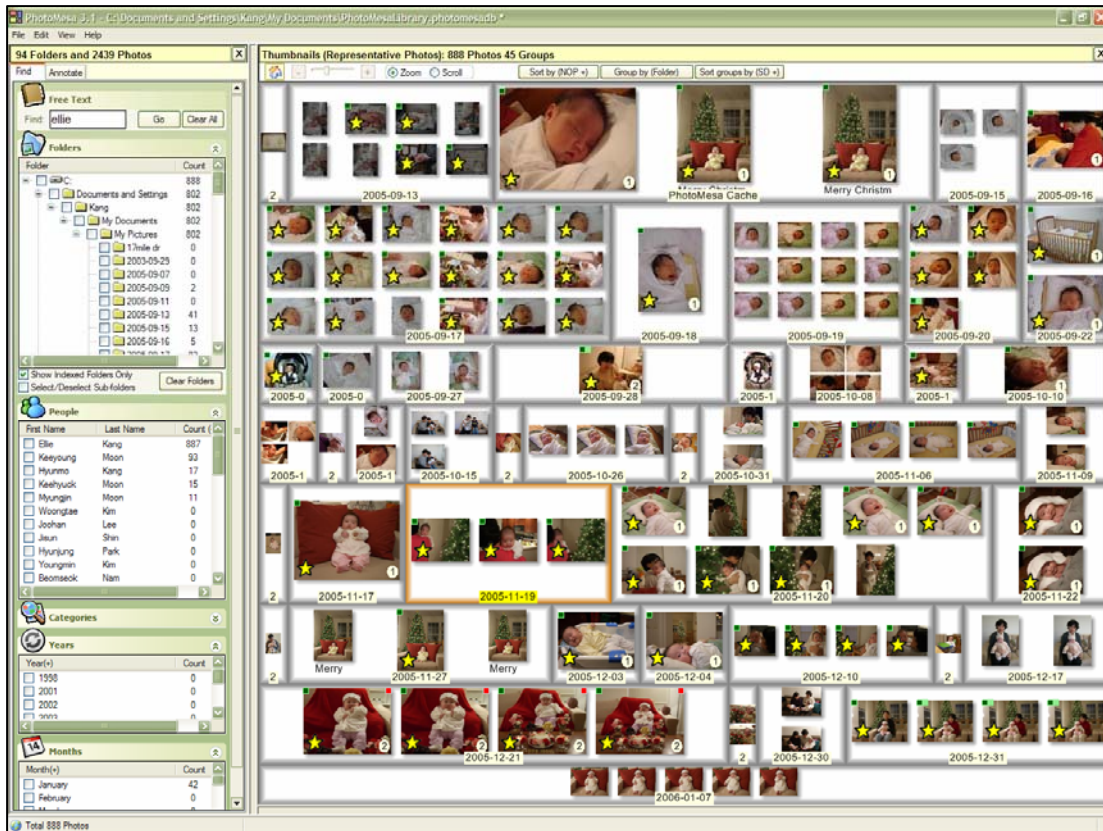
Small usability studies of the *Semantic Region* approach were conducted to discover potential user interface improvements, and to gain a deeper understanding about users' ability to understand, construct, and use semantic regions. The studies mainly focused on qualitative attributes such as user satisfaction rather than task performance. From these studies [Kang and Shneiderman, 2003], we learned that users liked the idea of spatial organization and dynamic grouping but they were not too interested in constructing their own models because it took too much time and effort. Users wanted rich built-in templates rather than user-defined templates, or more automatic features to reduce the steps for region construction and semantic specification. Based on these studies, we

simplified this approach by removing steps from the region construction and semantic specification while supporting automatic grouping and layout based on image metadata.

In a different project, we created an alternative and simplified approach for displaying groups of photos. This tool, called PhotoMesa, was developed as a successor to PhotoFinder by adding a zoomable user interface and the ability to show multiple groups of photos on the screen at the same time [Bederson, 2001; Bederson et al., 2002]. In PhotoMesa, users can dynamically control how photos are grouped based on available metadata (**Figure 5(a)**). For example, photos can be grouped by the folder they belong to, by when the photos were taken (month or year), by people in the photo, or by any user-defined categories (locations, events, objects, etc.) that the photos are annotated with. The groups are automatically placed using quantum treemap space-filling algorithm [Bederson et al., 2002], and the groups can also be sorted by photo metadata within a group such as number of photos in each group, starting date of photos in the group, title of the group, and so on. Each photo can be contained in multiple groups as long as it satisfies the grouping category.

Another approach created by Ben Shneiderman and Jack Kustanowitz is designed for groups of photos that have some relationship among them. In this case, it can be advantageous to show this relationship in the layout, rather than only in textual captions. **Figure 5(b)** shows a novel bi-level hierarchical layout, in which one region is designated for primary content, which can be a single photo, text, graphic, or combination [Kustanowitz and Shneiderman, 2005, 2006]. Adjacent to that primary region, groups of

photos are placed radially in an ordered fashion, such that the relationship of the single primary region to its many secondary regions is immediately apparent. A compelling aspect is the interactive experience in which the layout is dynamically resizable, allowing users to rapidly, incrementally, and reversibly alter the dimensions and content.



(a) PhotoMesa interface enables users to dynamically group photos based on photo metadata. The group layout is automatically generated using space filling algorithm but they can also be sorted by the metadata of group photos. PhotoMesa enables users to control the visible photos in various ways. In this example, PhotoMesa shows only the representative photos of each group.



(b) Bi-level hierarchical layout shows the photo groups taken in 6 locations during a trip by placing the groups clockwise in chronological order(left). A real estate browser with 5 communities and the price ranges of selected homes is shown on the right.

Figure 5: Organization of Personal Photo Collections

3.2 Navigation and Browsing

Image browsing is important for a number of reasons. First of all, no matter what information retrieval system is being used, the user has to browse the results of the search. It is certainly important to build query systems that help users get results that are as close to what is wanted as possible. However, there will always be images that need to be browsed visually to make the final pick. The second reason is that sometimes people browse images just for the pleasure of looking at those images, and they often do it with other people. This is especially true for personal photos. Looking at home photos has a lot of overlap with traditional retrieval systems. People still want to be able to find photos of particular people and events, etc. However, they are less likely to be time pressured to find a particular photo, and more likely to be interested in serendipity (that is, finding photos they weren't looking for) [Balabanovic et al., 2000; Bederson, 2002; Rodden and Wood, 2003, Shneiderman et al., 2006].

We also designed PhotoMesa to support this serendipitous style of use. It uses a zoomable user interface (Figure 6) to enable users to see all the images in a single view and concentrate on the images while browsing photos by zooming in or out without the need for interacting with other UI widgets such as scroll bar or popup window. This design is also based on the characteristics of personal image collections. As users have seen them at least once, they already have rough idea about their images and metadata. Hence, they often want to start browsing from the overview of their whole collection without worrying about missing some photos by accident (because they scrolled outside the currently visible region).

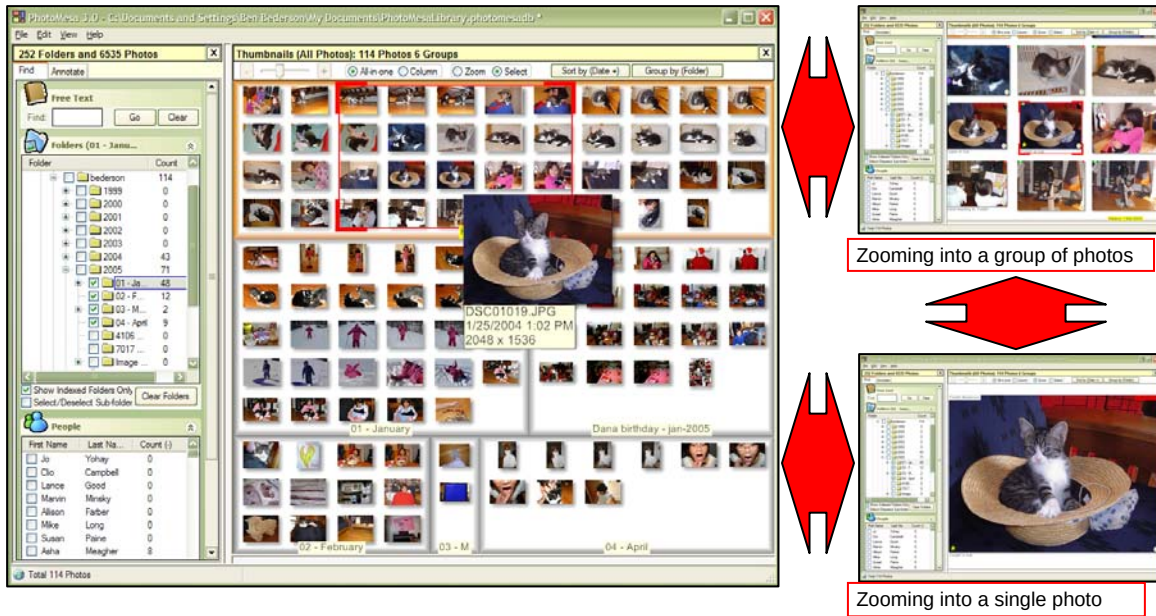


Figure 6: PhotoMesa shows all photos in a single view. The layout of photos depends on how many groups and photos have been loaded. A larger preview of any photo can be obtained by moving the mouse over one of the small thumbnails. PhotoMesa uses a Zoomable User Interface (ZUI) and supports browsing of photos by zooming in or out. The order as well as the group that photos are displayed in can be controlled dynamically by photo metadata.

This zoomable browsing approach was further enhanced in terms of system performance and users' cognitive workload by supporting semantic zooming [Bederson et al., 1996]. In the PhotoMesa interface, if there are too many displayed for users to recognize at once, PhotoMesa has the option of displaying just representative photos (Figure 4a). A layout such as *Time quilt* [Huynh et al., 2005] (Figure 8(a)) showed the improvement in photo browsing performance even more by making use of linear timeline layout on top of the 2D space-filling techniques with semantic zooming. This layout is designed to convey a temporal order while making reasonable use of screen space by packing a timeline layout into a rectangular screen space using a "line break" algorithm. Their user study shows that this combined layout enabled users to complete some photo browsing tasks faster than the quantum treemap layout and the timeline layout.

On the other hand, a new tool called PhotoSynth [Snavely et al, 2006] (Figure 8(b)) uses 3D space for photo browsing. The system takes a large collection of photos of a place or an object, analyzes them for similarities, and displays them in a reconstructed three-dimensional space so that users can interactively browse and explore large collections of photos of a scene using a 3D interface.

In addition to the design of novel browsing interfaces, we also developed a novel thumbnail generation algorithm which aims to reduce users' cognitive workload during browsing [Suh et al., 2003]. Using thumbnails has a tradeoff; bigger thumbnails use more screen space while smaller thumbnails are hard to recognize. Automatic thumbnail cropping is a method to find the optimal part of an image for thumbnail generation. The algorithm finds the informative portion of images based on a saliency model and cuts out the non-salient part of images. Thumbnail images generated from the cropped part of images increases users' recognition and helps users in visual search. On the other hand, the face detection based cropping technique [Suh et al., 2003] shows how semantic information can be used to enhance thumbnail generation further (when that semantic information about an image is known). For example, using face detection can more effectively create thumbnails with the images of people, which was demonstrated to significantly increase users' recognition and visual search performance.

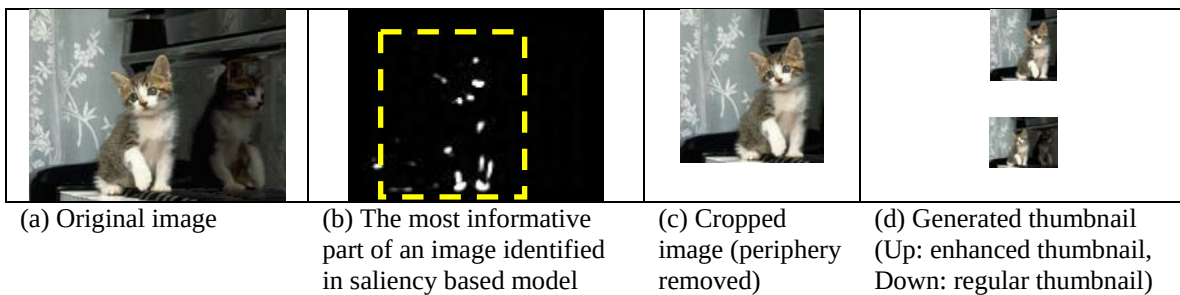
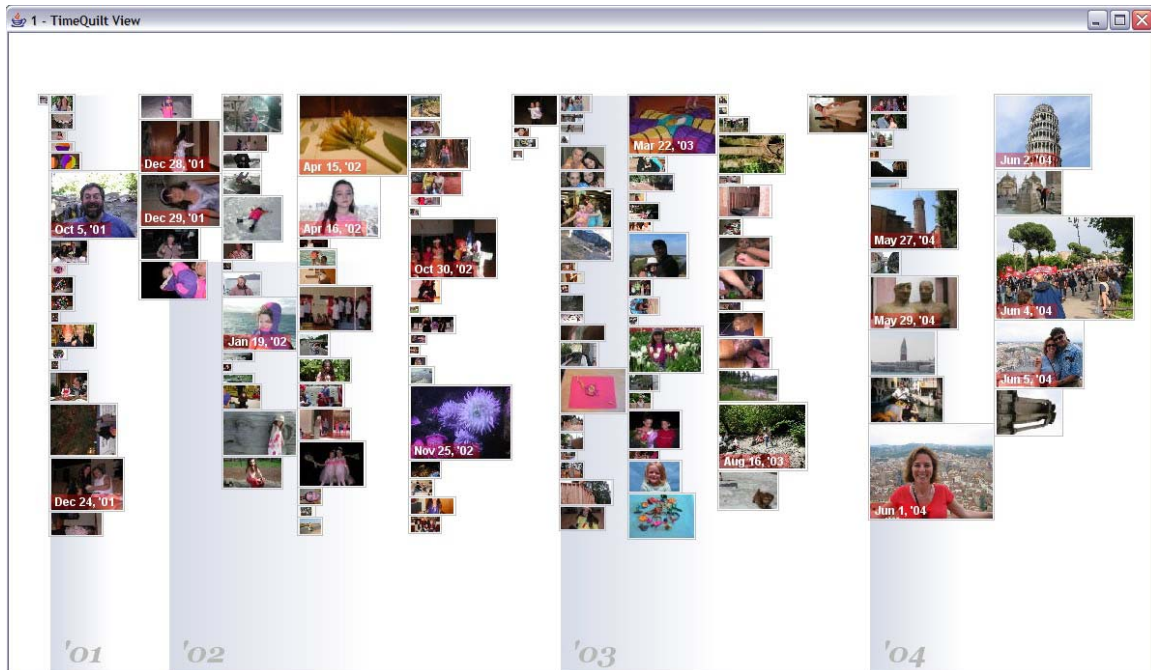


Figure 7 The saliency based model identifies the most informative part of an image. Thumbnail images created out of the core part help users' recognition and visual search performance.



(a) Time quilt – a layout designed to convey temporal order while making better use of screenspace than a timeline, showing approximately 5,500 photos with representative thumbnail overview



(b) PhotoSynth takes a large collection of photos of a place or an object, analyzes them for similarities, and displays them in a reconstructed three-dimensional space.

Figure 8 Photo Browsing Interfaces

3.3 Search and Meaning Extraction

There is no doubt that search is a hugely important photo management task. Users may be looking for photos of their grandmother, their trip to Disney World, or a friend's wedding. Interactive visualization to select and view dozens or hundreds of photos extracted from tens of thousands has become a popular strategy [Shneiderman et al., 2006]. In addition to simple photo search based on annotation, users sometimes want to extract meaning from their photo collections such as “Who else appeared in the pictures of my grandmother and me over 20 years and when and where did I meet her?”, “Who appeared both in my friend's wedding pictures and in my wife's friend's wedding pictures?”, “How many photos did I take for my daughter each year?, Were there any special events since she was born? Were there any seasonal differences in photo distribution?”, and so on. This kind of meaning extraction from personal photo collections becomes more common and popular tasks as users have more and more digital photos and richer metadata. However, little work has been done so far in this area.

One effort in this area, though, was done in the *Semantic Region* project [Kang and Shneiderman, 2006] described earlier in this section provides an interactive visualization technique called *region brushing* to support meaning extraction task by visualizing the relationships among the semantic regions. Region brushing is used to highlight the personal media items contained in multiple regions simultaneously (Figure 9(a)) and is often used for depicting both the intra-relationships and the inter-relationships among models. Since a personal media item can be contained in multiple semantic regions, the existence as well as the amount of the shared personal media items among the regions

well represents their relationships. In Figure 9(b), three models, HCI researcher, US map, and CHI conference calendar are combined together. Since a photo can be contained in multiple regions across the different models, many questions concerning the interrelationships of models can be answered through region brushing. Such questions might be “Find the name of the conferences that all nine people participated in and where was the conference held?”, “What was the name of the conference held in Atlanta, Georgia, and who did not appear in the photos taken at this conference?”, “Find the conferences in which Ben Shneiderman did not appear in the photos and where was the conference held?”, and so on. Furthermore, constructing new semantic regions recursively inside a region (e.g. location regions inside a person region) enables users to dynamically regroup the photos contained in a region and extract meanings.

Simple techniques such as *query preview* [Plaisant et al., 1999] or *faceted search* [Yee et al., 2003] can also help users extract meanings as well as filter down a collection and show potential targets from their personal collections. PhotoMesa applies these approaches by letting users search for photos using the Find tab (left panel in Figure 10) in six ways: with keywords; by the folder a photo is in; by the people in a photo; by the categories a photo was annotated with; and by the year or month a photo was taken. If a new photo set is searched by combining those six ways, each search panel previews the distribution of the searched photos so that users can see how many photos are from each folder, how many people (and who) appear in the searched photos, where the searched photos were taken (or how many photos were from other user-defined categories), when they were taken (month and year). With this feature, users can get information about the searched photos and use it for further exploratory search.



Figure 9: Region brushing: an interaction technique to visualize the shared items among the *Semantic Regions*. (a) regions containing any of the photos in the region the mouse is over (Shneiderman's region) are highlighted (b) visualization of interrelationships of the multiple models. By placing the mouse on Shneiderman's region, users can see which conference and which state his photos were taken.

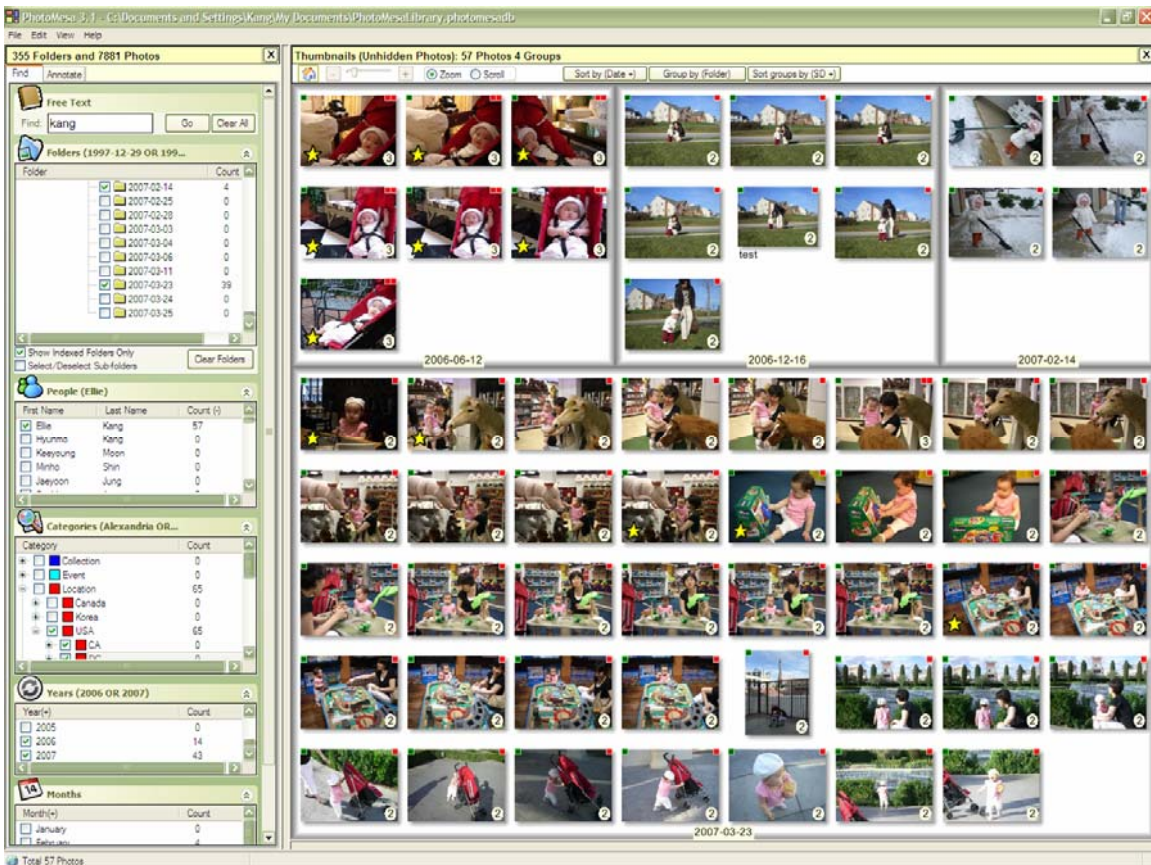


Figure 10. PhotoMesa search panel on the left enables users to search photos in six ways: with keywords; by the folder a photo is in; by the people in a photo; by the categories a photo was annotated with; and by the year or month a photo was taken. The searched photos are displayed in the thumbnail viewer on the right and their distributions are shown in each search panel.

3.4 Sharing and Distribution

Photo sharing is becoming one of the most common activities of digital photo management. Our earlier work provided interesting insights about how individuals and communities share their personal photographs. For example, a client-server model of PhotoFinder, PhotoFinder Kiosk (which was installed at CHI 2001), enabled members of a community of HCI professionals to enjoy, share and annotate photos at an interactive public exhibit [Kules et al., 2004, Shneiderman et al., 2002]. On the other hand, as mentioned earlier, there is a tremendous commercial activity in this area. There are numerous popular photo sharing websites such as Flickr, Picasa Web Albums, etc.

We previously mentioned some of the trade-offs of desktop applications and web-based photo-sharing sites. However, the current bifurcation requires users to pick one approach or the other to manage their photos. We believe that this split is an artifact of business requirements and software development practices, and that with a little more creativity and effort, it is possible to build a hybrid solution that bridges both worlds providing the advantages of each. For example, Sharpcast¹¹ enables users to share, access, and organize their photos either through local file systems and web sites. We also have heard of people using independent file synchronization tools, such as FolderShare¹² to keep multiple copies of their photos synced with metadata between different machines including web servers. All this points to the necessity of a hybrid approach. From a user's perspective, it doesn't make any sense that they should have to choose in advance whether they should commit to storing photos locally or online.

¹¹ Sharpcast, Inc. <http://www.sharpcast.com>

¹² Microsoft Inc. <http://www.foldershare.com>

To assess the usefulness of the hybrid approach, we built a prototype extension to PhotoMesa that hides the distinction between local and web access from the user. Since photos almost always start locally, after being transferred from a camera to a computer, we assume users will open the photos regularly in their local tool. Then the tool continuously synchronizes the local photos with the Flickr website using its public API in the background. In this manner, all the user's photos are also available within their private Flickr account. Furthermore, any changes or annotations to the photos on Flickr are synced back to the desktop application the next time they are accessed locally. In addition, full searches of Flickr's remote database can be performed within PhotoMesa the same way local searches are made, and remote photos are shown grouped by whatever metadata Flickr makes available.

There is a largely unmet need of users to be freed from the shackles of tightly controlled applications that limit access to their photos and metadata. In the increasingly decentralized world of information, that iron-fisted approach is just not sustainable any more.

4. Conclusion

The world of digital photo management has come a long way in the past few years, and we are pleased to see that the basic needs of most personal photo users are now being met. However we aren't done yet. There remain large gaps in meeting users' fuller needs such as recording richer annotations that are more accurate and broadly available. And it is crucial to support seamless sharing of photos and their stories in a way that lets multiple parties participate without losing ownership of the photos and metadata.

It is not clear how well current approaches will scale up. With the ever-shrinking cost of storage at the same time that cameras are becoming ubiquitous, we can expect that over an individual's life, it will be common to manage tens or even hundreds of thousands of photos.

The challenge of applying automated or human-assisted analysis techniques to these photos in a way that remains understandable and usable is ongoing. It also remains a challenge to find the right balance between local and web access that enables people to have reliable long-term access to their full resolution photos in a way that doesn't lock them into any particular vendor's system. People must also be able to maintain their privacy and control over photos while sharing and collaboratively editing the photos and stories about them. And when we envision the addition of audio and video to these collections which is already happening, all of these challenges grow even further.

The transition to digital photography has already largely happened. This implies that consumers are satisfied enough with current solutions – or at least prefer digital to film. However that doesn't mean that their long-term problems are solved, or that things are good as they could and should be. It is our job as designers and developers of these systems to address these challenges.

In this article, we described a range of projects that attempt to improve some of the problems already described. By integrating better annotation tools, visual presentation and query systems, and through the use of more open metadata storage approaches, we have the opportunity to take the growing digital photo market and make it really explode – while providing better systems for all of us to use with our own dear personal photos. Some directions not covered in this article, such as integrating annotations into other life activities (such as blogging, diary writing, email, etc.) also offer promise. In the end, we expect a rich ecosystem of photo-related technologies will continue to evolve that support the diverse set of users that take photos.

5. Reference

1. Balabanovic, M., Chu, L.L., and Wolff, G.J., Story-telling With Digital Photographs, *In Proceeding of Human Factors in Computing Systems (CHI 2000)*, ACM Press, pp. 564-571, 2000.
2. Bederson, B.B., PhotoMesa: A Zoomable Image Browser using Quantum Treemaps and Bubblemaps, *In Proceeding of ACM Conference on User Interface and Software Technology (UIST 2001)*, ACM Press, pp. 71-80, 2001.
3. Bederson, B.B., Shneiderman, B., and Wattenberg, M., Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies, *ACM Transactions on Graphics*, 21, 4, pp. 833-854. *
4. Chalfen, R., Snapshot Versions of Life, Bowling Green State University Popular Press, Ohio, 1987.
5. Chellappa, R., Wilson, C.L., and Sirohey, S., Human and Machine Recognition of Faces: A Survey, *In Proceeding of the IEEE*, Vol. 83, pp. 705-740, 1995.
6. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T., The Vocabulary Problem in Human-System Communication, *Communication of ACM*, 30, pp. 964-971, 1987.
7. Goto, Y., Jung, J., Ma, K., and McCaslin, O., The Effect of Direct Annotation on Speed and Satisfaction, SHORE, UMD, 2000.
<http://www.otal.umd.edu/SHORE2000/annotation/>

8. Huynh, D., Drucker, S., Baudisch, P., Wong, C., Time Quilt: Scaling up Zoomable Photo Browsers for Large, Unstructured Photo Collections. *In Extended Abstracts of Human Factors in Computing Systems (CHI 2005)* , Portland, OR, pp. 1937-1940, 2005.
9. JEITA (Japan electronic and information technology industries association), Exchangeable image file format for digital still cameras: Exif Version 2.2, 2002.
<http://www.exif.org>
10. Jung, J., Empirical Comparison of Four Accelerators for Direct Annotation of Photos, 2000. <http://www.cs.umd.edu/hcil/photolib/paper/cmssc498paper.doc>
11. Kang, H. and Shneiderman, B., Dynamic Layout Management in a Multimedia Bulletin Board, *In Proceeding of IEEE International Symposium on Human-Centric Computing Languages and Environments (HCC 2002)*, 2002. *
12. Kang, H., Plaisant, C., and Shneiderman B., Helping Users Get Started with Visual Interfaces: Multi-Layered Interfaces, Integrated Guidance and Video Demonstrations *In Proceedings of Digital Government Conference (DG2003)*, 2003. *
13. Kang, H., Shneiderman, B., Exploring Personal Media: A Spatial Interface Supporting User-Defined Semantic Regions, *Journal of Visual Language and Computing*, volume 17, issue 3, pp. 254-283, 2006. *
14. Kang, H., Shneiderman, B., Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder, *In Proceeding of IEEE International Conference on Multimedia and Expo (ICME2000)*, 2000. *

15. Khella, A., Bederson, B.B., Pocket PhotoMesa: A Zoomable Image Browser for PDAs., *In Proceeding of Mobile and Ubiquitous Multimedia (MUM 2004)*, ACM Press, pp. 19-24, 2004.
16. Kuchinsky, A., Pering, C., Creech, M.L., Freeze, D., Serra, B., Gwizdka, J., FotoFile: A Consumer Multimedia Organization and Retrieval System, *In Proceeding of ACM Conference on Human Factors in Computing Systems (CHI99)*, pp. 496-503, 1999.
17. Kules, B., Kang, H., Plaisant, P., Rose, A., Shneiderman, B., Immediate Usability: A Case Study of Public Access Design for a Community Photo Library, *Journal of Interacting With Computers*, volume 16, num 6, pp. 1171-1193, Elsevier, 2004. *
18. Kustanowitz, J. and Shneiderman, B., Meaningful presentations of photo libraries: Rationale and applications of bi-level radial quantum layouts, *In Proceeding of Joint Conference on Digital Libraries*, ACM Press, 2005. *
19. Kustanowitz, J., Shneiderman, B., Hierarchical layouts for photo libraries, *IEEE MultiMedia*, 13, 4, pp. 62-72, 2006. *
20. Lieberman, H. and Liu, H., Adaptive linking between text and photos using common sense reasoning, *In Proceeding of Adaptive Hypermedia and Adaptive Web Systems*, Malaga, Spain, 2002.
21. Naaman, M., Yeh, R. B., Garcia-Molina, H., Paepcke, A., Leveraging Context to Resolve Identity in Photo Albums, *In Proceeding of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2005)*, 2005.
22. Phillips, P. J., Grother, P. J., Michaels, R. J., Blackburn, D. M., Tabassi, E., Bone, J. M., Face recognition vendor test 2002: Evaluation report., NISTIR 6965, 2003.

23. Plaisant, C., Venkatraman, M., Ngamkajornwiwat, K., Barth, R., Harberts, B., Feng, W., Refining query previews techniques for data with multivalued attributes: The case of NASA EOSDIS, *In Proceeding of Advanced Digital Libraries 99 (ADL'99)*, IEEE Computer Society Press, pp. 50-59, 1999.
24. Rodden, K. and Wood, K., How do People Manage Their Digital Photographs?, *In Proceeding of ACM Conference on Human Factors in Computing Systems (CHI 2003)*, Fort Lauderdale, 2003.
25. Shneiderman, B., Bederson, B.B., and Drucker, S.M., Find that photo!: interface strategies to annotate, browse, and share., *Communications Of the ACM*, 49, 4, pp. 69-71, 2006. *
26. Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th Edition, *Addison Wesley Longman*, Reading, MA, 2005. *
27. Shneiderman, B., Direct manipulation: a step beyond programming languages, *IEEE Computer* 16(8), pp. 57-69, 1983. *
28. Shneiderman, B., Kang, H., Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos, *In Proceeding of International Conference Information Visualization (IV2000)*. IEEE Computer Society, pp. 88-95, 2000. *
29. Shneiderman, B., Kang, H., Kules, B., Plaisant, C., Rose, A., and Ruchir, R., A photo history of SIGCHI: evolution of design from personal to public, *ACM Interactions*, Volume 9, Issue 3, pp. 17-23, 2002. *
30. Snavely, N., Seitz, S. M., Szeliski, R., Photo tourism: exploring photo collections in 3D, *ACM Transactions on Graphics (TOG)*, Volume 25 Issue 3, pp. 835-846, 2006.

31. Suh, B., Bederson, B.B., Semi-Automatic Image Annotation Strategies Using Event Based Clustering and Clothing Based Person Recognition, *Journal of Interacting with Computers*, Elsevier, 2007.
32. Suh, B., Ling, H., Bederson, B.B., Jacobs, D., Automatic Thumbnail Cropping and its Effectiveness, *In Proceeding of ACM Conference on User Interface and Software Technolgy* (UIST 2003), pp. 95-104, 2003
33. Von Ahn, L. and Dabbish, L., Labeling Images with a Computer Game, *In Proceeding of ACM Conference on Human Factors in Computing Systems* (CHI 2004), ACM Press, pp. 319-326, 2004.
34. White, R., Kules, B., Bederson, B.B., Exploratory Search Interfaces: Categorization, Clustering and Beyond, *SIGIR Forum*, volume 39, issue 2
35. Yang, M., Kriegman, D., and Ahuja, N., Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Mach Intelligence*, 24(1), pp. 34-58, 2002.
36. Yee, K-P., Swearingen, K., Li, K., and Hearst, M., Faceted Metadata for Image Search and Browsing, *In Proceeding of ACM Conference on Human Factors in Computing Systems* (CHI 2003), ACM Press, 2003.
37. Yoshitaka, A., and Ichikawa, T., A Survey on Content-Based Retrieval for Multimedia Databases., *IEEE Trans on Knowledge and Data Engineering*, 11(1): pp. 81-93, 1999.
38. Zhao, W., Chellappa, R., Philips, P.J., Rosenfeld, A., Face Recognition: A Literature Survey, *ACM Computing Surveys*, Vol 35(4), pp. 399-458, 2003.

DateLens: A Fisheye Calendar Interface for PDAs

BENJAMIN B. BEDERSON and AARON CLAMAGE

University of Maryland

and

MARY P. CZERWINSKI and GEORGE G. ROBERTSON

Microsoft Research

Calendar applications for small handheld devices are growing in popularity. This led us to develop DateLens, a novel calendar interface for PDAs designed to support complex tasks. It uses a fisheye representation coupled with compact overviews to give the big picture in a small space. The interface also gives users control over the visible time period, as well as supporting integrated search to discover patterns and outliers. Designed with device scalability in mind, DateLens currently runs on desktop computers as well as PDAs. Two user studies were conducted to examine the viability of DateLens as a replacement for traditional calendar visualizations. In the first study, non-PDA users performed complex tasks significantly faster with DateLens than with the Microsoft Pocket PC 2002™ calendar (using a PDA emulator). In addition, they rated DateLens as being easier to use than the default calendar application for a majority of the tasks. In the second study, the participants were expert Pocket PC users and the software was run on their own devices. Again, DateLens performed significantly faster for the complex tasks, and there were satisfaction differences favoring each calendar for different kinds of tasks. From these studies, it is clear that DateLens is superior for more complex tasks such as those associated with longer time periods. For daily event tracking, users familiar with the default Pocket PC calendar strongly preferred its daily view and behaviors.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*, *Interaction styles, screen design*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Fisheye distortion interfaces, information visualization, calendar interfaces, PDAs, animation, graphics

The portion of this work performed at the University of Maryland was funded in part by a generous gift from Microsoft Research.

Authors' addresses: B. B. Bederson and A. Clamage, Human-Computer Interaction Laboratory, Computer Science Department, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742; email: {bederson, aclamage}@cs.umd.edu; M. P. Czerwinski and G. G. Robertson, Microsoft Research, one Microsoft Way, Redmond, WA 98052; email: {marycz, ggr}@microsoft.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.

© 2004 ACM 1073-0616/04/0300-0090 \$5.00

1. INTRODUCTION

More and more people carry small Personal Digital Assistants (PDAs) with them to help manage day-to-day information. While these devices can be helpful for retrieving relevant information when it is needed, our informal polling of colleagues tells us that they are less helpful for more complex tasks, such as those involving larger time spans. This is not surprising since these devices have limited screen space, forcing users to jump around through multiple screens, making it harder to relate disparate pieces of information.

We designed DateLens,¹ a new calendar interface for PDAs, to better support these more complex tasks such as picking a good weekend to go camping, counting the number of Mondays in November, finding the start and end dates of a trip. These are instances of tasks that we classify more generally as scheduling, navigating and counting, and searching, respectively.

Our secondary goal was to design a calendar interface that would scale down to smaller devices such as mobile phones, and up to larger devices such as desktop displays. This is important since individuals are likely to access their calendar information from these and other devices. Offering a single interface across devices would give users a consistent experience, and, eventually, the ability to more readily switch between devices using whichever one is more accessible.

The DateLens design addresses these goals by using a fisheye distortion technique coupled with carefully designed visualizations and interactions appropriate for a pen-based device and a small display [Bederson et al. 2003]. Figure 1 shows DateLens running on a Pocket PC device.

The basic approach used by DateLens is to start with an overview of a large time period using a graphical representation of each day's activities. Tapping on any day expands the area representing that day, and reveals the list of appointments in context. Users may change focus days, zoom in further for a full day view, search for appointments, and reconfigure the viewable space.

This interface shows varying time span displays within the same framework using animated transitions between view changes, and thus, may improve users' ability to maintain a sense of where they are. This paper describes the interface along with the results of two user studies comparing DateLens to the standard Microsoft Pocket PC 2002TM calendar interface. Evidence from these studies supports our hypothesis and provides many useful ideas for improving the DateLens user interface design.

1.1 Related Work

Fisheye distortion techniques, initially called bifocal displays, were introduced by Spence and Apperly [1982] 20 years ago. At that time, the basic concept was to distort the information space so focus items were enlarged while peripheral items were shrunk. A few years later, Furnas [1986] generalized this approach by suggesting a "degree-of-interest" function, which calculates the relevance of

¹DateLens is available for download at www.cs.umd.edu/hcil/datelens.

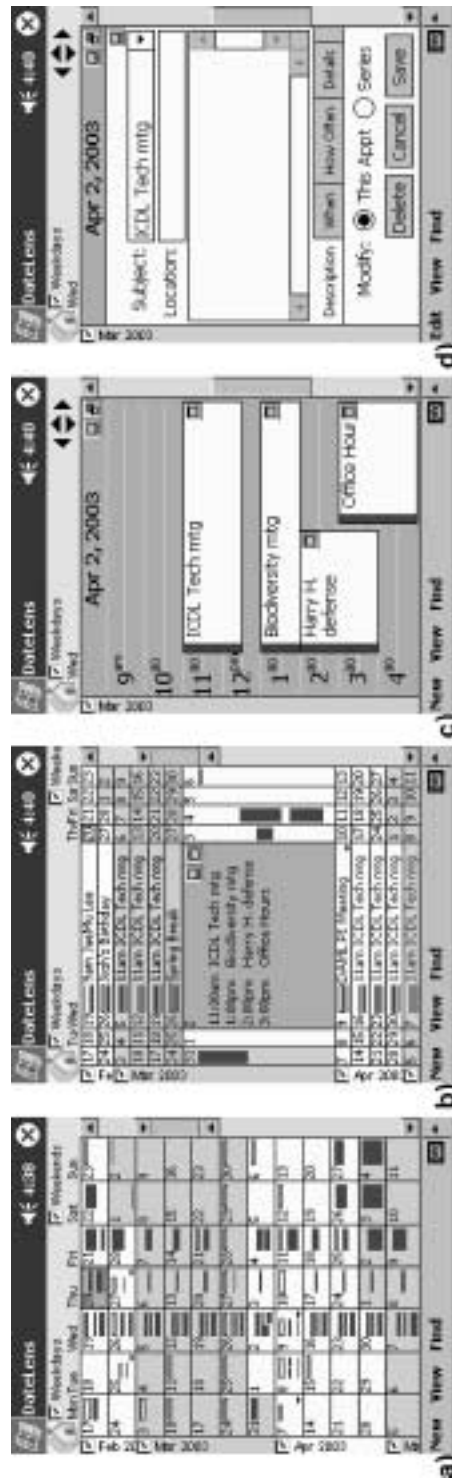


Fig. 1. The DateLens interface with the view configured to show 12 weeks and consecutive levels of detail as the view is: a) shown as an overview; b) zoomed into one day; c) focused on that day; and d) zoomed into an appointment. All transitions between views are animated.

each item in the information space. This result is then used to calculate the size and visibility of that item.

Fisheye distortion techniques have been applied to a number of domains, from graphs [Sarkar and Brown 1992] to trees [Lamping et al. 1995] to menus [Bederson 2000], among others. Their effectiveness has been mixed, but in at least some cases, such as hierarchically clustered networks [Schaffer et al. 1997], fisheye interfaces have been shown to benefit users. The common theme has been that fisheye views are appropriate when users need to see details of some specific items in the context of a large information space.

The idea of using fisheye distortion to view calendars is not new. It was first suggested over ten years ago by Furnas [1991], where he described a textual Lisp-based calendar program. We followed the basic approach Furnas created at that time. A tabular display shows days in the calendar, and clicking on individual days causes the amount of space allocated to that day to be increased. Furnas' calendar used varying amounts of space to show different days, so that the focus day was largest, and other days were sized in inverse proportion to the distance from the focus day (although days in the past were always tiny because the assumption was that users were more interested in the future.) This program, while impressive for its time, did not support graphical representations of appointments, animation, searching, or full screen views, and did not have an interface to control which and how many weeks to display. It was not designed with small displays in mind. In addition, it was not evaluated with users, and was not pursued past the publication of the above-mentioned technical report.

Fisheye visualizations have also been used successfully to view and interact with generic tabular information. The best known example of this is probably Table Lens, which presents an interface for tabular data [Rao and Card 1994]. This visualization approach was designed for tables with many rows, but a modest number of columns. It represents each cell with a horizontal bar whose length is proportional to the value of the cell for numerical data, and whose position represents categorical data. The height of each row is scaled to fit the available space. Users may then focus on individual or multiple cells (or rows or columns). In addition, users can sort rows to help see relationships within the data. While this approach is somewhat similar to the present work in that it uses a fisheye distortion to view tabular data, it is not directly applicable to calendar information as it is designed for spreadsheet style information that has one item per cell, rather than the multiple and possibly conflicting appointments of calendars. In addition, it does not support the searching or navigation that calendar users require. Nevertheless, the acceptance of this technique (as demonstrated by its successful commercialization [Inxight 2003]) gives hope that users will be able to understand and navigate calendar information in a tabular format using a fisheye view.

Researchers have also developed other techniques to visualize and interact with calendar information. Plaisant and Shneiderman [1992] were among the first to develop small visual representations of calendar information. Mackinlay et al. [1994] developed a 3D "spiral calendar" visualization. This

approach, while not suitable for small devices since it displays several visual representations simultaneously, does have a fisheye-like quality in that it displays detailed appointment information with visual links back to larger scale calendars. So, users can see what week an appointment comes from, what month that week is in, what year that month is in, and so forth.

Perhaps surprisingly, fisheye techniques have rarely been used for interfaces for PDAs and other devices with small displays. One use was by Björk et al. [1999] who used “flip zooming” to display web pages and then personal information including calendar data [Björk et al. 2000] on a PDA as demonstrated within their PowerView application. Flip zooming consists of presenting one medium-sized focus page and several tiny pages in the periphery that can be used for navigation. At a high level, the basic approach of flip zooming is similar to DateLens. However, DateLens differs from flip zooming in that flip zooming is designed to support hierarchical textual data while DateLens supports tabular data with a natural visual abstraction. Furthermore, flip zooming has system-defined viewpoints while DateLens allows users to define views. Finally, DateLens adds two important new features: integrated search and animated transitions.

2. DATELENS

DateLens is the fisheye calendar interface we designed for use primarily on a PDA (Figure 1). It was designed and built at the University of Maryland and Microsoft Research then joined the project to run the experiments. DateLens runs on both Pocket PC devices as well as on full-sized Windows machines.

As described in the related work section, DateLens draws its design from a range of earlier work. While the individual features of DateLens represent variations of existing approaches, our primary contribution is the integration of a host of techniques to create a novel application that is both usable and useful in an important domain. In addition, we present two studies that show how this design works in comparison to a more traditional calendar design for small devices. We hope that DateLens illustrates how existing techniques can be applied in new ways to new domains, and in doing so, advance the state of the art.

DateLens was built to target currently available devices running the Microsoft Pocket PC operating system. These devices are small enough to fit comfortably in a hand, have high quality 240×320 pixel screens, and fast enough processors to support modest animation.

Since DateLens was designed for a pen-based PDA, we have been careful to design the interaction so that it requires minimal text entry and simple interaction. The entire interface can be accessed with single taps, although dragging offers some modest extra features—including access to tool-tips and fast scrollbar usage.

The rest of this section describes the DateLens interface in detail, including a description of its navigation capabilities, the visualizations that represent calendar information at different sizes, and how search capabilities are integrated into the interface.

2.1 Navigation

A fundamental characteristic of DateLens is its ability to support users in easily customizing their view of the calendar. Most commercial calendar applications provide mechanisms to directly switch between day, week, month, and year views, and to change which range of dates are visible with each view. However, the different views are disconnected and inflexible. One goal of DateLens was to offer the same functionality in terms of a range of views, but to do so in an integrated and flexible fashion. Using animation and fisheye distortion, users can see the relationship between the range of dates they are viewing and the previous view. As such, users should not have to expend as much mental effort to manage context and figure out “where they are.”

The basic organization of the display is tabular (Figure 1). Each row represents one week, with columns representing the days of the week. The number of visible rows can be changed from one, representing a single week, to 52, representing an entire year. When a day is tapped, the view is gradually changed to expand the day that was tapped on (Figure 1a, 1b). This and all view transitions are animated over a user-defined period that defaults to about 250 milliseconds. The animations are rendered by linearly interpolating the position of each grid line for the in-between frames. The result is that each view smoothly transitions from one view to the next.

The view can be changed through direct manipulation by interacting with the calendar itself, by manipulating widgets in the periphery of the display, or by using special hardware button shortcuts. One of the challenges was to make it extremely easy to configure the view. The final design only uses interaction mechanisms that most users are familiar with, including tapping on an item that they want more information about, and manipulating familiar buttons and widgets.

Direct Manipulation. DateLens was designed to take advantage of user familiarity with clicking on hyperlinks to find more detailed information about the item they clicked on. It allows users to tap anywhere on a day to focus on that day, minimizing other days (Figure 1a).

Within a focused day (Figure 1b), users can tap on the background, or tap on the maximize button to zoom in to a full day view. Or, users can tap on the minimize button to go back to original view with no days focused.

Within the full day view (Figure 1c), users can tap on an appointment's background or the appointment's maximize button to view the appointment details. Tapping on the day's minimize button returns to the original view, and tapping on the overlapping-windows button returns to the focus day view.

Within the full appointment view (Figure 1d), scrolling shows the full contents of the appointment. Tapping on the minimize button returns to the full day view.

2.1.1 Peripheral Widgets. The “range slider” widget on the right side of the display controls how many weeks are visible at a time. It acts like a traditional scrollbar, but the thumb has two additional buttons that are used to manually set the start and end dates of the current view. The view dynamically changes as

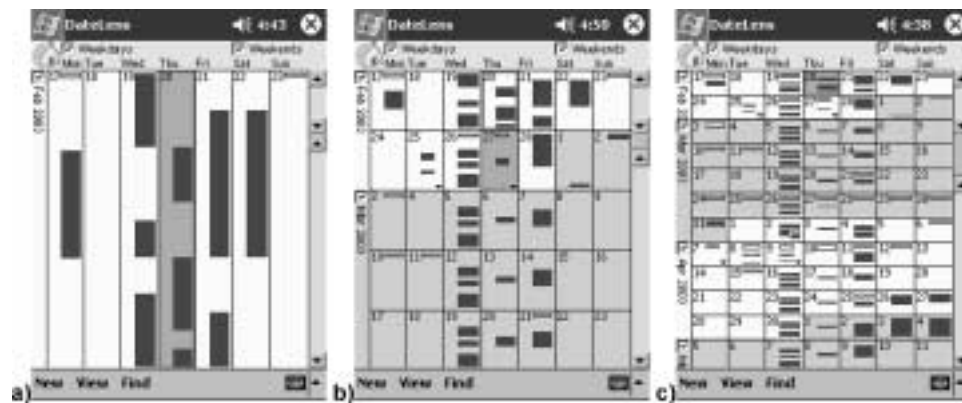


Fig. 2. A series of views as the bottom of the scrollbar thumb is dragged downwards to show: a) 1 week; b) 1 month; c) and 3 months.



Fig. 3. The views resulting from a) unchecking the weekend checkbox; and b) leaving just weekends in March checked.

the scrollbar is manipulated, but for animation efficiency, appointments within days are only shown when the scrollbar is released. Figure 2 shows a range of views as controlled by the range slider.

Another way to configure the space is to manipulate checkboxes on the top and left sides of the display. These checkboxes specify whether space gets allocated fully to the correlated set of items, or if those items are minimized. The left side of the display has one checkbox for each month. The top side of the display has one checkbox for weekdays and one checkbox for weekends. Figure 3 shows the result of two different configurations of checkboxes.

There is also a “home” button in the top-left corner of the display that resets all navigation settings to a user-configurable state.

2.1.2 Hardware Buttons. On desktop computers, graphical user interfaces typically offer keyboard shortcuts so that expert users can quickly

access commonly used functions. On PDAs, there is no keyboard, but there are special hardware buttons that applications can use for a similar purpose.

When DateLens runs a Pocket PC device, the “calendar” button can be used to cycle between the preset views of one day, one week, one month, and three months. The “joystick” (a small 4-way rocker switch) offers motion in four directions, and we use that to move the “active” day, which is indicated to the user by a dark blue highlight. Pressing the center of the joystick focuses on that day (or maximizes it if it was already focused). The joystick can also be used when a day is focused or maximized. On a desktop PC, the keyboard can be used to navigate through the calendar by using the arrow keys to change dates, the enter key to zoom in, the escape key to zoom out, and the space bar to switch between preset views.

2.2 Visual Representations

A crucial aspect of the design of DateLens is the visual representation of the calendar for different configurations. We decided to use a “semantic zooming” approach that we developed from our prior work with Zoomable User Interfaces [Bederson et al. 2000]. Semantic zooming means that objects are visually represented differently depending on how much space is available to display them. Using this technique, there are no explicit view modes. Rather, the fisheye distortion algorithms first allocate space, then each cell renders itself using a view that is appropriate to the available space. The graphical views are scaled to fit the available space, while the textual views use a constant-sized font, and the text is wrapped to fit in the available space.

The four available views are:

- *Tiny View*. This shows a graphical representation of the day’s appointments. It includes depictions of all-day appointments with a white rectangle at the top of the day rectangle. It uses color to represent different appointment types, and depicts appointment conflicts using multiple columns. The pen can be dragged across appointments to show tool tips with textual information about the appointment under the pen. In large-scale views, where each row is thinner than a threshold, the black lines separating rows are removed to make the display less “heavy” (Figure 1a).
- *Agenda View*. This shows a textual list of appointments in order by time. There are actually two representations in this view. If there is a smaller amount of space available, a smaller font is used, and the appointment times are not listed. If there is more space available, a larger font is used, and the appointment times are listed (Figure 1b).
- *Full Day View*. This shows a traditional full day view with a schedule of the entire day, and appointments positioned at the appropriate times. It shows all-day appointments and conflicting appointments, and uses color in the same way as the tiny view (Figure 1c).
- *Appointment Detail*. Traditional widgets are used to show the details of a particular appointment (Figure 1d).

The decision of which view to show is, by default, made by DateLens depending on how much space is available in a given cell. There is a threshold cell size for each view that determines what to display.

While the initial design of DateLens was motivated by small displays, one of the attractions of its design is that it scales nicely to larger displays as well. Since the decision of which representation to use is made based on how much space is available for a particular cell, the same layout algorithm and rendering code works on larger displays as well. Figure 4 shows DateLens running on a desktop PC at 1400×1050 resolution.

2.3 Search

The last primary component of DateLens is search. Search is important because it lets users identify patterns and outliers within a large time span. When users search in DateLens, the days that contain an appointment matching the search criteria are highlighted. The highlights are maintained while users continue to operate DateLens normally so the space can be explored to understand the results of the search.

In addition to highlighting the visible days within the current view, “attribute mapped scrollbars” [Hill and Hollan, 1994] show which days are highlighted in both the past and future (Figure 5). The scrollbar shows indicators representing which days are highlighted within and outside of the current view. This mapping of search results to the scrollbar is fixed to this single attribute, and is not under user control.

While it is natural to support searching for arbitrary user-entered text strings, that is somewhat problematic because it is notoriously difficult and slow to enter text on PDAs. So, while DateLens supports free text search, it also provides two search mechanisms that do not require text entry: pre-built searches and searches based on existing appointments.

2.3.1 Free Text Search. To search manually, users enter text in the text box at the top of the display. A somewhat tricky issue is how to deal with search strings that consist of multiple words. Should the search consist of the conjunction or disjunction of the words, or the actual search string? No one of those approaches worked for all of the experimental tasks in our studies. Instead, DateLens operates like many current Web search engines, using a simulated “vector” based search [Baeza-Yates and Ribeiro-Neto, 1999, pp. 27–30].

Vector searches work by using a number of characteristics of the search to rank the order in which the results are shown. This results in an ordering that usually matches user expectations. Exact string matches are typically listed first, conjunctions (where all the words match) are listed next, and disjunctions (where not all the words match) are listed last.

DateLens is a little different since it does not present an ordered list of search results, but instead highlights whichever days match. Rather than ordering search results, DateLens only presents highly ranked search results. It works by first performing an exact string match, and if there are any results, they alone

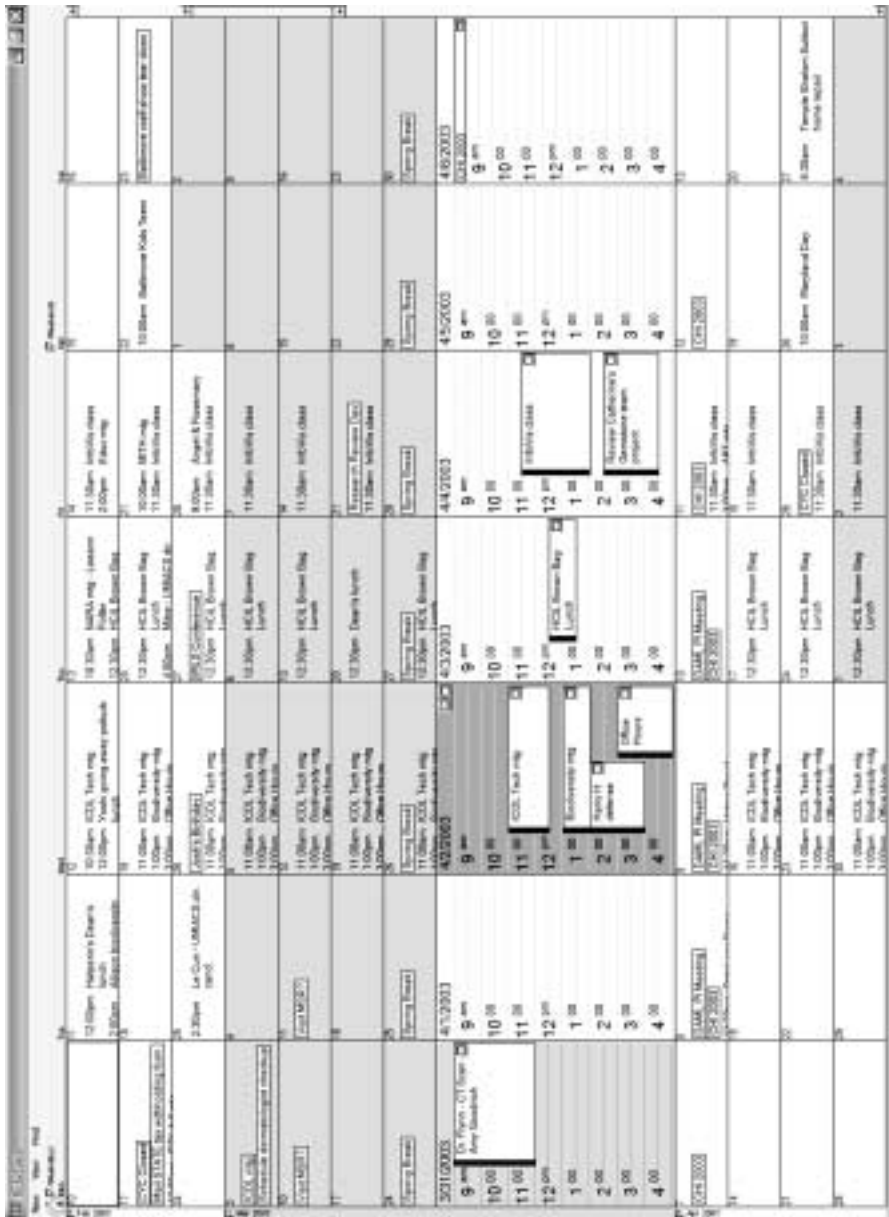


Fig. 4. DateLens scaled up to 1400 × 1050 pixels on a desktop computer. There is room to display an entire week with the full day representation, even while showing three months of the calendar.

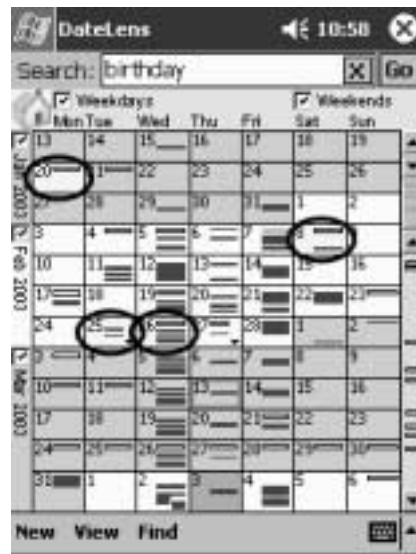


Fig. 5. DateLens showing the results of searching for birthdays (colored highlights are circled for black-and-white printing clarity). A few individual birthdays are highlighted. In addition, the scrollbar shows several days in the future that are highlighted, which also have birthdays.

are shown. If there are no results, then it searches for days with appointments matching all the words in the search string, and highlights those days. If there are still no matches, it then searches for days with appointments that match any of the words in the search string and highlights those. This combination of search strategies mimics the main effect of vector searches, and seems to work well in practice.

2.3.2 Predefined Searches. Since it seems likely that many searches by a particular user will be for the same thing, we added support for predefined searches. The goal is to make it even easier to search for commonly sought events, such as travel, meetings, doctor appointments, or holidays.

A simple approach is to search on appointment metadata which is supported by Pocket PC as well as other calendar systems. The problem with this approach is that most users do not annotate each appointment with categories.

Rather than force users to do something they do not want to do, DateLens takes advantage of what information is already available—the appointment text itself. While there are no guarantees that a user will enter a similar event the same way every time, we have found through informal polling of our colleagues, that people often do represent similar events with similar textual descriptions—although they vary significantly from one user to another.

We built support for predefined searches where each search would actually look for a match within any of a set of search strings. For example, searching for “Doctor Appointments” actually searches for “doctor”, “dr.”, or “dr appt”. While these predefined searches are currently hard-coded, our intention is for users to be able to modify them, or define their own.

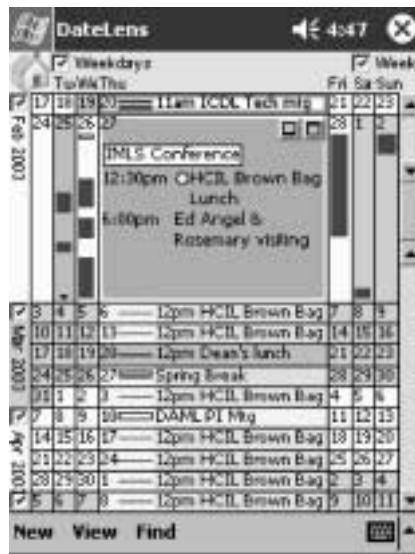


Fig. 6. The results of clicking on the appointment “HCIL Brown Bag Lunch”. Most Thursdays are highlighted in yellow which shows a recurring appointment. In addition, several days are highlighted in orange which match either “HCIL”, “Brown”, “Bag”, or “Lunch”.

This approach has been tested on the authors’ calendar data, and it works quite well except for a few idiosyncrasies that we discovered. For instance, one of the authors uses textual graphics such as “— >” to indicate travel. Some of these are searchable as a text string, but some are not because they span multiple lines.

Nevertheless, this approach still appears practical. Since having good quality predefined searches is so useful, some users are likely to adapt the way they write appointments to be more consistent given an easy-to-use search tool. While the general idea of requiring users to adapt to system requirements is undesirable, this is better than the current solution, which requires manual annotation of each appointment with categories.

2.3.3 Existing Appointment Search. Since it is quite common for people to create recurring appointments, where the same appointment happens at regular time intervals, it seems natural to have a simple way to support finding and visualizing those recurrences. DateLens does this by offering a way to find all appointments matching an existing appointment. This works by tapping on any appointment. All other days with exactly matching appointments are highlighted in yellow, just as if the text of the appointment subject had been typed into the search box.

We noticed, however, that sometimes users had similar appointments that were not exact matches. Based on the implementation of free text searching, we also search for days with appointments that partially match the specified appointment, and we highlight those in orange (Figure 6). This is a simple

solution that can be readily ignored by users if they are only interested in the exact recurrence, but offers more information if desired.

2.4 Implementation

The implementation of DateLens consists of about 20,000 lines of C# code, and runs on whatever platforms the Microsoft Common Language Runtime (CLR) is available on. Currently, the CLR is available on the Pocket PC platform and all desktop versions of Windows except Windows 95. DateLens uses the standard Pocket Outlook database on the Pocket PC platform and the Outlook database on desktop platforms, so users can switch between commercial calendars and DateLens while using the same data. All features described in this article are fully implemented.

The most complex part of the implementation is the layout algorithm used to allocate space for each calendar day. The layout algorithm takes as input the number of days in a week, number of weeks displayed, the checkbox states, the focus day, and the size of the window. The subtle part of the layout algorithm relates to the large set of configurations of the space for which it must work. Specifically, there must be a balance between the minimum size of unfocused cells and the maximum size of focused cells. That is, we have found it makes most sense for each day to stay within a range of sizes whenever possible. So, DateLens defines preferred minimum and maximum sizes for unfocused and focused cells, and allocates space within those ranges whenever possible.

The other subtle part of the DateLens implementation is performance. To make DateLens respond to user interaction rapidly, and to animate transitions smoothly, the overall structure had to be carefully designed. The primary things taken into account that contribute to its performance are:

- Custom rendering loop. Rather than use a toolkit, which might have been easier in some respects, DateLens uses a custom data structure, rendering loop, and “picking” implementation. This was particularly appropriate since the basic data structure is a table, and is easily handled as a two dimensional array.
- Space vs. time tradeoff. Things are precomputed and stored, rather than being computed on the fly. The most obvious place this occurs is in the layout of the days.
- Render only what is needed during transitions such as the basic grid structure. However, some visual aspects, such as highlighted days have to be shown during scrolling since users sometimes look for that while scrolling. In addition, we found that by rendering the appointment details of just the cell that the user tapped on, users were largely unaware of the lower rendering quality during animations. This is because the user’s eye is almost always looking at the point of interaction, and is not aware of modest changes in the periphery.

All transitions in DateLens are animated with simple linear interpolation that occurs over 250 milliseconds by default. We picked such a short animation

time because the visual changes are quite small (usually not changing by more than a few centimeters).

In order to get DateLens to run on both Pocket PCs and desktops, we largely relied on the multi-platform capabilities of Microsoft's .NET platform. That worked for all of the core functionality of DateLens including animation and rendering. However, in order to take advantage of any device-specific components such as hardware buttons on the Pocket PC and keyboard input on the desktop, we had to write some platform-specific code. In addition, we wrote platform-specific code to access the Outlook database, which uses a slightly different API on the different platforms. In terms of performance, we designed for the slower system (Pocket PC), which then worked well on the desktop. The only significant difference in terms of performance is that on the Pocket PC, we did not render appointment details during animation or scrolling because of the limited processing power; while on the desktop version, we render all details all the time.

3. BENCHMARK STUDY

We performed two studies comparing DateLens to the current shipping user interface of Microsoft's Pocket PC 2002™ calendar (Figure 7). The first was an early benchmark study that we ran before DateLens was running on actual Pocket PC devices, so we performed the study on a desktop computer using a Pocket PC emulator and a mouse. For this first study, we looked at novice users who had not used Pocket PCs before. Once DateLens was running well on Pocket PC devices, we ran a second study with Pocket PC experts from Microsoft using the actual devices with pen input.

The goals of the first study were to examine the initial design ideas behind DateLens, in order to see if the user interface design could be improved, and to compare its overall usability against an existing product.

3.1 Participants

We gathered eleven knowledge workers (five females) who were all experienced Microsoft Windows and Office users, as confirmed through an in-house validated recruiting screener questionnaire. Participants were screened to be between 25–50 years of age (average age of 39.2). In addition, the participants fit some broad characteristics of being target end users of personal digital assistants (PDA), but were purposefully chosen to not own or use one at the current time. We thought this aspect of the user group would be especially interesting since for some reason these users had avoided buying a PDA, and perhaps the presentation of PDA information on a small screen was a primary issue for them.

3.2 Materials

The user's display was an LCD set to 1024×768 resolution with 16-bit color. Each calendar occupied a 240×320 pixel window centered on the display (standard Pocket PC resolution). All participants were run singly in a usability lab, on a Dell Pentium 450 MHz computer running Windows XP. An MS

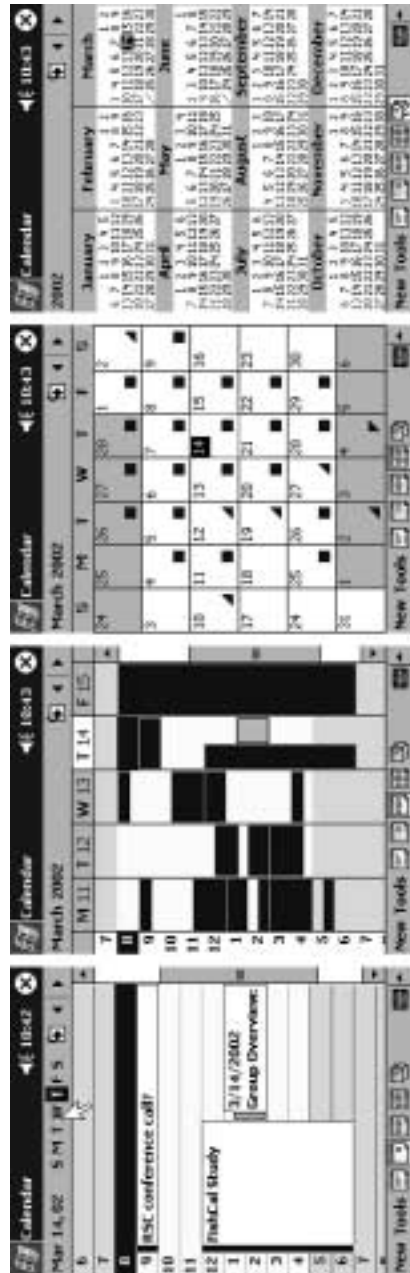


Fig. 7. Screen shots of the Microsoft Pocket PC Calendar program that was used in the study showing day, week, month, and year views.

Natural keyboard and an MS IntelliMouse were used as input devices, though the “wheel” was not functional with the calendars.

3.3 Procedure

Brief (approximately 5 minutes) tutorials were provided to participants prior to each set of tasks on each calendar. The tutorials consisted of a one page sheet of instructions on operating the two interfaces, and the participants then tried each of the described mechanisms. The tutorial focused on the features and functions of each calendar that were necessary for completing the experimental tasks. However, 2 additional minutes were provided for the user to explore the calendar as he or she saw fit prior to starting. The participants performed an isomorphic set of 11 tasks using each calendar (example tasks are listed below). The order of calendar use and task set for the calendar were both counterbalanced in order to minimize the effects of training, or the possibility of one task set being slightly more difficult than the other. However, tasks within a set were not randomized.

We worked with the Pocket PC usability engineers to devise a task list that reflected real tasks carried out frequently by Pocket PC users with their calendars. In addition, based on our own use of the Pocket PC, we added some tasks that were a bit more complicated, requiring more navigation than simply entering an appointment. Therefore, participants completed three different categories of calendar tasks: Searching (e.g., finding the start and/or end dates of appointments), Navigation and Counting (e.g., navigating to particular appointments or monthly views and counting things like the number of conflicts, or the number of Mondays in a month) and Scheduling (e.g., scheduling a meeting, a vacation or a fun night out). The navigation and counting, and the scheduling tasks were the most complex, in general, in that they required users to keep information in short-term memory while navigating or scrolling through timeline views. The navigation and counting category of tasks may not be performed as often by real users as the searching and scheduling tasks, but we added them in an effort to more deeply probe the value of the DateLens visualization, as well as to generalize our findings for the information visualization community, where navigation can be a key issue.

All tasks were given a deadline of two minutes to complete in order to keep the session under two hours (and because a two minute deadline seemed reasonable for being able to discover information from one's PDA calendar.) Task times and success rates, verbal protocols, and user satisfaction and preference questionnaire data were collected throughout the session. Sessions lasted approximately one and one half hours. The complete tutorials and task lists for this study are available in Appendix A in the ACM Digital Library.

3.4 Results

3.4.1 Task Times. Task times for one participant were unavailable, as his session expired before he was able to get to the 4th task using DateLens. A tape jam prevented us from obtaining the task times for one other participant for the Pocket PC, and both participants' data had to be ignored for the task

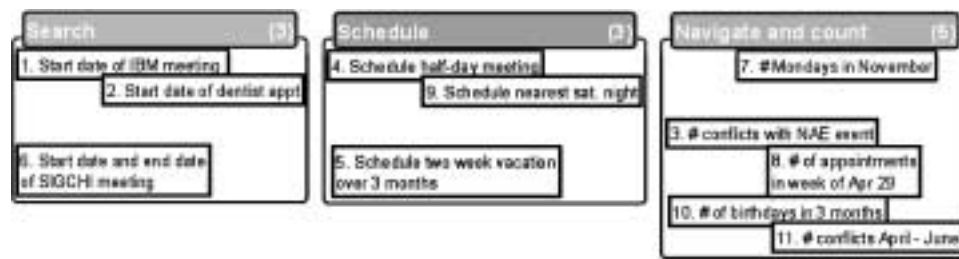


Fig. 8. Summary of tasks grouped by type and complexity. In each group, the top tasks are the “simple” ones, and the bottom ones are the “complex” ones.

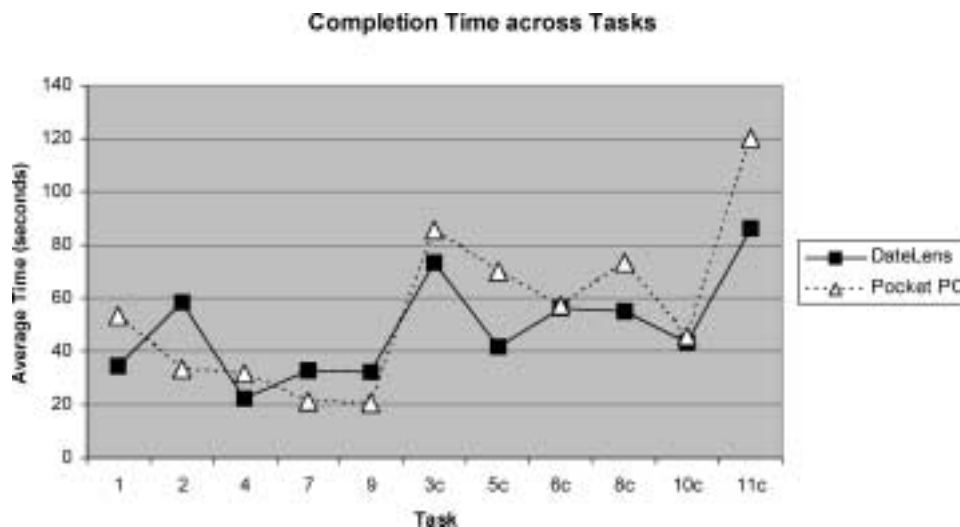


Fig. 9. The time spent by study participants to complete each task using the DateLens and standard Pocket PC calendar interfaces. Tasks are listed with simple tasks on the left, and complex (marked with a ‘c’ in the task number) on the right.

time analysis. A 2 (calendar type) $\times$ 11 (Task) repeated measures Analysis of Variance (RM-ANOVA) was carried out on the completion times for the tasks. Tasks were performed faster using DateLens (49 seconds versus 55.8 seconds for the Pocket PC, on average), but this result did not reach significance, $F(1, 8) = 3.5$, $p = .08$. There was also a significant main effect of task, $F(10, 80) = 12.9$, $p < .01$, and a significant calendar $\times$ task interaction, $F(10, 80) = 2.05$, $p = .04$. Of particular interest was the fact that, as the tasks became more complex (tasks 3, 5, 6, 8, 10 and 11), the DateLens task time advantage grew reliably. We divided the tasks into “simple” and “complex” by calling the five fastest tasks the “simple” ones, and the six slowest tasks the “complex” ones, and found that DateLens performed 12% faster for the complex tasks. We measured the task speed for this definition as the fastest of the two average completion times (one for each interface). Figure 8 shows a summary of the tasks, grouped by type and complexity. The time that participants spent on each task can be seen in Figure 9.

We decided to carry out another analysis based on our significant effect of task, comparing the tasks we observed to be simpler to those that were relatively more complex (again, task 3, 5, 6, 8, 10 and 11) to explore this result further. A 2 (calendar type) $\times$ 2 (simple v. complex task average) RM-ANOVA was performed on the task times. Again, no significant effect of calendar was obtained, but there was a significant effect of task complexity, $F(1, 9) = 79.55$, $p < .001$, and a significant interaction between calendar type and task complexity, $F(1, 9) = 8.4$, $p = .017$. The post-hoc analyses revealed that the Pocket PC calendar provided a significant advantage for simpler tasks, while the DateLens application provided a significant advantage for the more complex tasks (all p-values less than .05 using Bonferroni adjustments for multiple tests). Our observations of users in the study led us to believe this latter finding was primarily due to the fact that DateLens allowed flexible views across time in a user-defined manner during complex tasks. By allowing the users to view as little or as much information as needed for a particular task and by using animation to help users more clearly see view transitions, short-term memory load was reduced in the DateLens application. In the Pocket PC application, rigid views (day, week or month) meant that multi-view questions required retaining search results in short-term memory while switching through the required views for the complex tasks. In addition, the integrated search mechanism and its resultant views in DateLens made finding particular sets of events via keyword matching quite effective. Unfortunately, it is not possible to tease out exactly how often users used the search mechanism as opposed to adjusting the views in this study. The searching tasks were the likely candidates for using search, but sometimes users tried search and then quickly jumped to adjusting the view or vice versa, so in many cases a combination of both was involved. Since the Pocket PC showed a day view by default, simpler tasks were well supported via a familiar, Outlook-style view, and this may have led to significant performance gains for those tasks. These results can be seen in Figure 10.

3.4.2 Success Analysis. A 2 (calendar type) $\times$ 11 (Task) repeated measures Analysis of Variance (RM-ANOVA) was carried out on the success rates for the tasks. Overall, tasks were successfully completed significantly more often using DateLens (on average, an 88.2% success rate, versus 76.3% for the Pocket PC), $F(1, 9) = 37.1$, $p < .001$. In addition, there was once again a significant main effect of task, $F(10, 90) = 12.9$, $p < .001$. The interaction, however, was not significant. These data are shown in Figure 11, where it becomes clear that certain tasks (3, 5, 6, 8, 10 and 11) were successfully completed more often with DateLens, on average, but the result was not significant. For the most difficult task (#11), no participant using the Pocket PC completed the task successfully.

3.4.3 Satisfaction and Preference. Users completed “ease of use” ratings on a scale of 1 to 5 (1 = very difficult, 5 = very easy) after every task. DateLens was rated higher across a majority of the tasks, especially the most difficult task (task 11—how many conflicts are there for the next 3 months?—had the lowest success rate for both calendars). A 2 (calendar type) $\times$ 11 (Task) RM-ANOVA was carried out on the satisfaction data. DateLens was rated higher than the

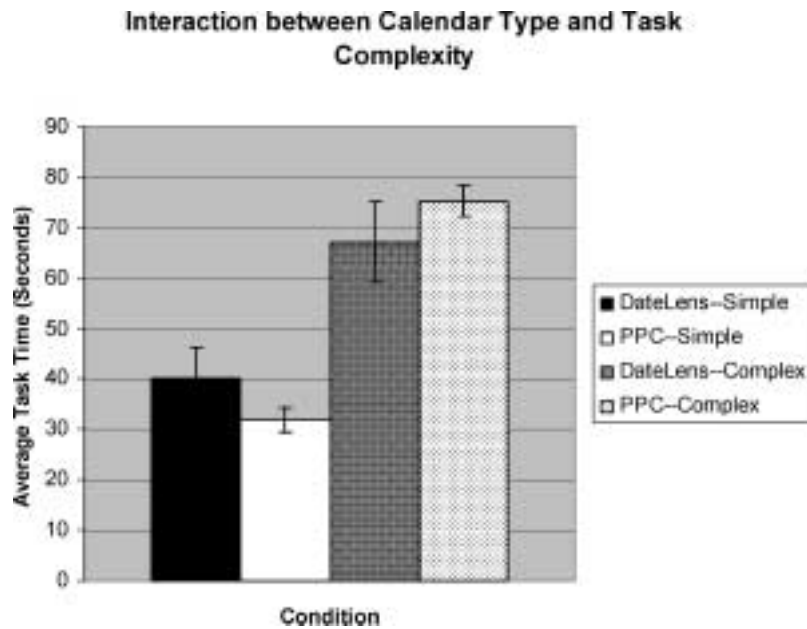


Fig. 10. Effects of calendar type by task complexity on average task times.

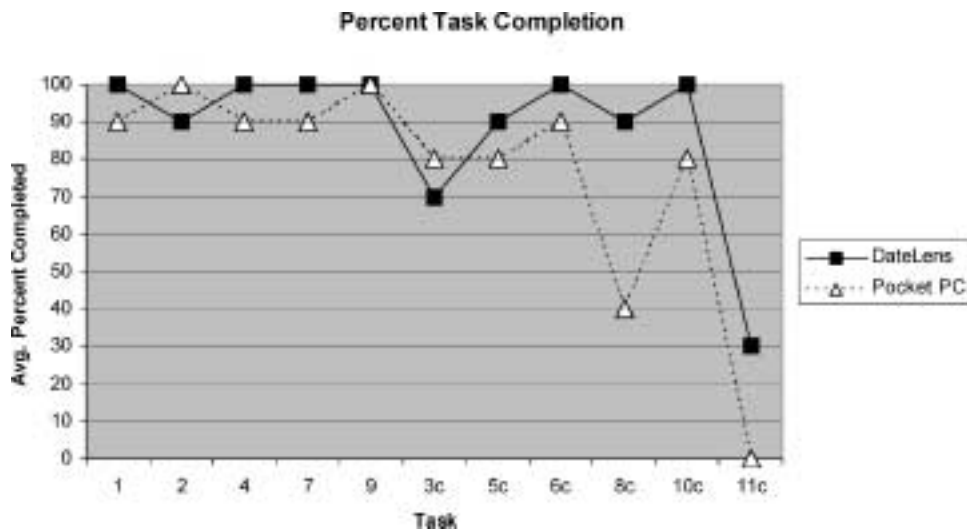


Fig. 11. The percent of tasks completed by study participants for each task. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

Pocket PC in terms of task satisfaction, on average, $F(1, 9) = 4.37$, $p = .06$, a borderline significant result. The average task ratings are shown in Figure 12. As can be seen in the Figure, only tasks 2, 3 and 7 were rated highest for Pocket PC. This only correlates partially with success rate, since tasks 2 and 3 were

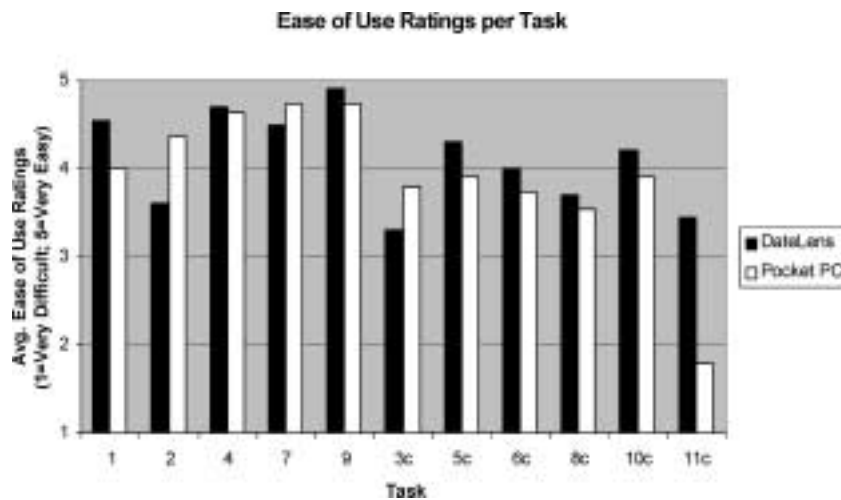


Fig. 12. The ease of use rating for each task by study participants. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

carried out more successfully with that calendar but task 7 was carried out more successfully with the DateLens application.

Participants completed an overall preference questionnaire at the end of the session. No significant differences emerged in this preference data, though the Pocket PC was slightly more preferred overall (6 out of 11 participants chose the Pocket PC Calendar; one participant abstained and stated that she wanted features of both calendars in the ideal calendar; 4 participants chose DateLens). Most participants said that they would prefer a combination of features from each of the two calendars during the post-session debriefing. The most often cited reason for choosing the Pocket PC calendar was the participants' familiarity with the Outlook XP calendar, which is similar in many ways. The overall preference data is shown in Figure 13.

3.4.4 Usability Issues. Many usability issues were observed with this initial version of DateLens, as well as the Pocket PC calendar, and good design feedback was received from the participants about how best to move toward redesign. We summarize those issues pertaining to DateLens here.

Many users disliked the view of the calendar when more than 6 months were shown at once, claiming that the individual days were simply too small at that point to be useful. In addition, users wanted to see all 24 hours of a day's full view, but the prototype was limited in functionality to simply show a 9-5 view for this iteration of user testing. More importantly, a visualization of search results tried to show as many "hits" in the calendar as possible without making the view so crowded as to be useless. If the result a user was looking for was scrolled out of view (into the future), there was no visual indicator as such (the attribute mapped scrollbar that shows search results was added after the study was run.) Users voiced strong concerns about the readability of text, and being able to set their own default views according to their individual eyesight needs.

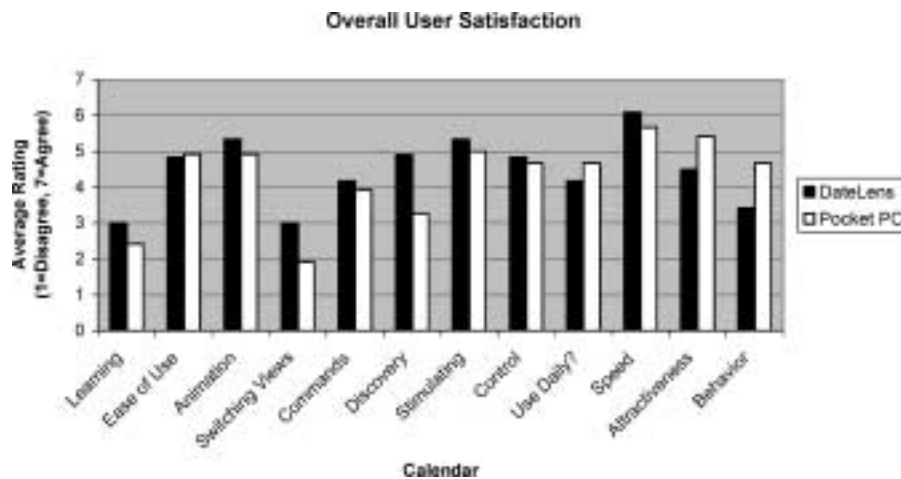


Fig. 13. Overall user satisfaction.

Users also wanted more control over how their weeks were viewed (e.g., should the week start with Sunday or Monday?). Finally, users wanted better visual indicators of conflicts for both calendars, for example, red highlights and/or a “conflicts” filter. These usability issues were of a generic nature and were seen to affect the use of DateLens pretty much during every task, though the search issues obviously strongly affected the tasks that required direct searching. As a result, many users attempted to use search to target known events in a task, but often abandoned this effort half-way through or adjusted the view in addition to attempting to search. Improvements in text legibility and adding an attribute-mapped scrollbar might enable better utilization of the search mechanism in future versions of DateLens. Users in this study did not choose to browse when completing tasks.

In summary, DateLens performed quite well despite its novelty and this being its first iteration of user testing, especially for tasks that turned out to be complex and that required custom views or large amounts of navigation. The responsiveness to direct user manipulation, the ability to easily create custom views on the fly, its clear presentation of conflicts, and integrated search utility were all design innovations that participants thought would be valuable to any calendar used daily for planning and reviewing one’s schedule. The Pocket PC calendar was seen by participants to be consistent with other Microsoft calendar products, and this was viewed positively. Several participants wanted to see a combination of the two calendars taking advantage of the good features of both in a final product.

4. FOLLOW-UP USER STUDY

Our second user study also compared DateLens to the Pocket PC 2002 Calendar with the following differences in the software and study design:

- Both conditions were run on the Pocket PC device with a stylus for input instead of using a desktop with mouse.

- The DateLens software was improved to respond to several of the biggest usability concerns—including the ability to see appointments outside 9am–5pm, and the ability to see search hits outside the current view (through marks on the scrollbar).
- We performed a combined longitudinal and lab study. We wanted to examine the usefulness of the visualization over a longer time period than the typical lab situation would allow, in the user's own work environment with their personal data. We believed that we would learn much more about the usefulness of both calendars if users used them in situ and as part of their daily routine. However, we also wanted the accuracy and control afforded by the laboratory experience. We decided to combine both methods for an acceptable evaluation. After deploying DateLens to our users, we had them use it for three entire days before we performed a comparative evaluation of it and the Pocket PC calendar. Users were still more familiar with the Pocket PC calendar in the end, but at least they got a chance to experience the DateLens software for a number of days before the experimental session occurred.
- We chose participants who were experienced Pocket PC users. This had the benefit of giving us feedback from real target end users who were familiar with PDA calendaring. Since DateLens is a research project, and personal calendar data is very important to individuals, we decided to include only willing participants from Microsoft due to privacy concerns that might be an issue with external participants. This introduced a potential bias, but it is likely to be in favor of the Microsoft product. Because of this, we feel that any advantage seen in DateLens is likely to be more pronounced among a general user population.

4.1 Participants

Eight Microsoft employees (1 female) who use Pocket PC heavily every day were recruited for participation. The participants ranged in age from 28 to 36 years. All were expert PC users and had been using the Pocket PC for over 2 years. Their job titles varied from usability engineer, designer, and researcher to group manager. Participants were given vouchers for food or coffee from the cafeteria as a gratuity.

4.2 Materials

Participants were sent email study instructions that included download and installation information for DateLens, a brief tutorial on how to use the key features in the DateLens software and a survey that participants were to fill out after using the software for 3 days. Participants used their own Pocket PC devices, which varied widely in terms of the amount of memory included, form factor and even version of Windows CE (some of the participants were running a beta version of the next release not available to the public). However, in order to standardize the study to some degree, we required all users to drop back to using the 2002 version of their Pocket PC software. Users filled out another questionnaire after performing an experimental task session with both the

Table I. Average Satisfaction Ratings After Using the DateLens Software for 3 Days
(1 = Disagree, 7 = Agree)

This calendar was difficult to learn how to use:	4.57
This calendar was easy to use for the tasks you performed today:	3.86
The animations allowed me to stay better oriented as I switched to different calendar views in the user interface:	3.71
It was difficult to switch to the proper view in the calendar:	4.00
I sometimes wondered if I was using the right command:	4.00
I was able to discover more about a schedule using this calendar compared to existing tools:	3.57
Working with this calendar was mentally stimulating:	4.33
I felt in control of this calendar when I was using it:	3.00
I would like to use this calendar every day:	3.00
The speed of this calendar is about right:	3.71
This calendar has a very attractive presentation:	2.29
This calendar sometimes behaves in a way I don't understand:	4.86

DateLens and Pocket PC software. The design of this second questionnaire is based on the IBM Post-Study System Usability Questionnaire [Lewis 1995].

4.3 Procedure

Participants were asked to use DateLens to view their own personal calendar data for 3 days. After they had done so, they filled out a survey about their initial reactions. Upon receiving this survey, we asked each participant to perform two sets of isomorphic tasks, once using DateLens and next using the Pocket PC calendar. The order of presentation of each of the two calendars was counterbalanced, as was the task set. The task set was replicated from those tasks used in the first study, however some abstraction and/or changes were required due to the fact that we were using the participants' real data. The entire task set for the second study is available in Appendix B. If a participant did not have any data in their calendar corresponding to a task, the task was either skipped (this occurred only twice) or a very close alternative was used in the target's place (e.g., find your next dentist appointment was allowed if there was no doctor's appointment on the participant's calendar). All tasks were timed using an electronic timing program on the experimenter's laptop. In addition, the experimenter recorded comments, errors and observations provided by the participants for both calendar visualizations while performing the tasks. Finally, after completing all tasks on each calendar, the participant filled out a standardized satisfaction questionnaire.

4.4 Results

After using DateLens for three days, users rated its initial ease of use. Mean ratings for DateLens are shown in Table I, based on a Likert scale rating system of 1 = Disagree, 7 = Agree). Users felt that the DateLens calendar needed a more attractive design, and users indicated that they did not feel in control when they used the calendar, and that it behaved unpredictably at times. For these reasons, their likelihood of using the software daily was rated lower, on average. However, these ratings really point to remaining usability problems

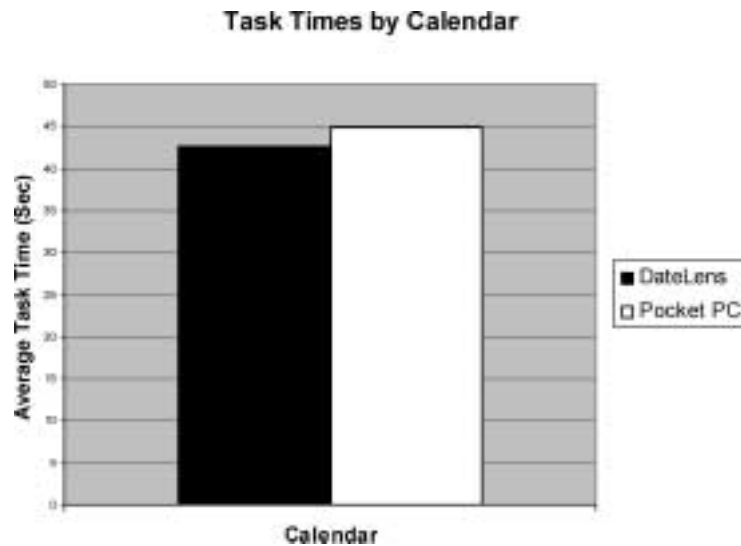


Fig. 14. Overall average task times for each calendar.

(more on this is detailed below) and are not necessarily indicative of flaws in the concept of the fisheye calendar. They do point to the challenge of making novel visualizations work effectively in everyday tasks especially with users who are already quite familiar with existing tools. These ratings also provide a good barometer for what aspects of the software should be focused on for further improvements to the design.

The results from the experimental session provided task time results allowing us to compare the two calendars quantitatively. A 2 (calendar) $\times$ 11 (task) RM-ANOVA was performed on the task times. No main effect was observed for calendar, $F(1, 5) = 0.36$, $p = .58$, but a main effect of task was observed, $F(10, 50) = 4.0$, $p < .001$. No interaction was obtained. This data is shown in Figure 14. However, based on our complexity analysis from the first study, we suspected it was the complexity of the task that might be driving results in this second study as well. Therefore, the same analysis was carried out again. A 2 (calendar type) $\times$ 2 (complexity of task) RM-ANOVA was carried out on the average task time data. Results were identical to the first study, in that no significant main effect emerged for calendar type, but a significant main effect for complexity was revealed, $F(1, 7) = 29.83$, $p < .001$, and a significant interaction between calendar type and complexity was observed, $F(1, 7) = 8.9$, $p = .02$. Post-hoc tests showed that once again the Pocket PC was reliably better for the simple task set, while the DateLens calendar provided significant benefits for the more complex task set, with a 17% average time improvement. These results can be seen in Figure 15.

The post-experimental task satisfaction questionnaire results were also analyzed. DateLens received a slightly worse average satisfaction rating, as can be seen in Figure 16. This difference did not reach statistical significance in a 2 (calendar) $\times$ 13 (question) ANOVA, $F(1, 7) = 1.34$, $p = .28$. The only significant

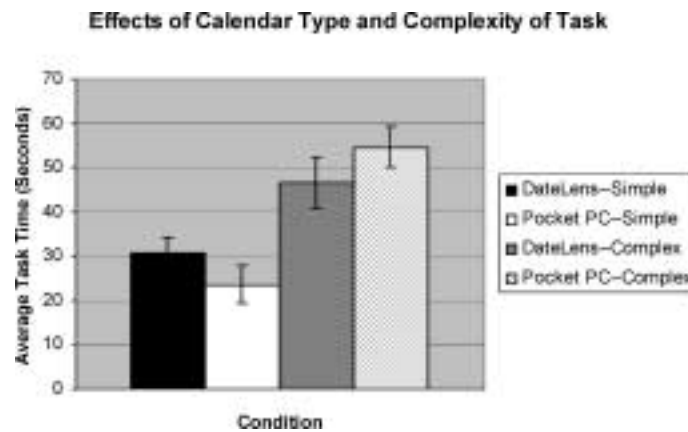


Fig. 15. Effects of calendar type and task complexity on overall average task times in study 2.

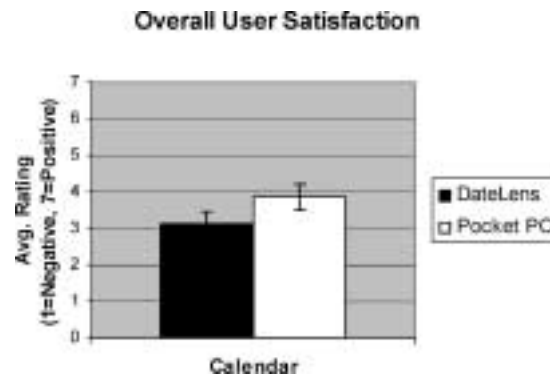


Fig. 16. Overall average satisfaction for each calendar based on the experimental tasks.

effect observed in the analysis was that of the interaction between calendar and questionnaire item, $F(10, 70) = 3.07$, $p = .002$. Post-hoc t-tests showed only one significant difference among the items at the $p = .05$ level. Question 4 asked participants whether they could effectively complete the experimental tasks using each system, with DateLens rated significantly higher than Pocket PC, $t(7) = -2.9$, $p = .02$, two-tailed. There were two borderline significant differences in favor of the Pocket PC: Question 7 asked about user comfort while using the systems, and not surprisingly the Pocket PC was rated borderline significantly more comfortable, $t(7) = 2.2$, $p = .06$, two-tailed. Question 8 asked users how easy it was to learn to use each system, with Pocket PC rated borderline significantly easier to learn, $t(7) = 2.2$, $p = .06$. All of the satisfaction ratings can be seen in Table II. Note that the actual questionnaire recorded satisfaction with low values being positive, but we are reporting the data here flipped to see the results more clearly.

Several usability issues were observed for both calendars in this study, although for DateLens the usability issues that were observed were now quite

Table II. Average Post-Task Satisfaction Ratings for Each Calendar (Note That Lower Ratings Are Better). DL = DateLens, PPC = Pocket PC

Questionnaire Item	DL	PPC
Overall, I am satisfied with this system.	3.9	3.3
Overall, I am satisfied with how easy it is to use this system.	4.4	2.6
It was simple to use this system.	4.6	2.9
I could effectively complete the tasks and scenarios using this system.	2.8	4.5
I was able to complete the tasks and scenarios quickly using this system.	3.3	3.6
I was able to efficiently complete the tasks and scenarios using this system.	3.9	4.0
I felt comfortable using this system.	4.0	2.0
It was easy to learn to use this system.	4.8	2.6
The “look” of this system was pleasant.	4.0	3.3
I liked using this system.	3.6	2.8
The organization of information in this system was clear.	3.5	3.0

different from the first iteration because of changes we made to the design in the interim. Therefore, we will focus on the usability issues we observed for the DateLens visualization, with an eye toward ideas for further design improvements. DateLens was in a period of rapid development, and we ran the second study at a point before it was finished. So, some problems were observed that have since been fixed. The most serious issues we observed concerned tap targets in the user interface simply being too small. Experienced Pocket PC users often use their fingers to tap on targets in the user interface, either because they have misplaced their stylus, they don't have time to grab the stylus, or because the unit is docked and finger tapping is preferred over stylus tapping in the vertical orientation. Even with the stylus, users often invoked incorrect behaviors and actions accidentally when attempting to scroll or make UI selections in DateLens. Future versions of the UI could enlarge the size of frequently used targets, especially the scrolling controls and checkboxes. In addition, the scrollbar had two sets of arrows, which were not distinguishable except for by their spatial positioning. Users wanted to see a new design for the specialized scrollbar, in which the different arrows had discriminating features so that the user could tell which feature mapped to which function. We are experimenting with alternative designs to support this, such as replacing the arrows on the inner buttons with hash marks. Finally, some issues remained surrounding the visualization of search results. While users really enjoyed the ability to search with keywords and some pre-defined defaults (something that the Pocket PC does not do in an integrated fashion in the calendar), search results sometimes were only viewable off screen, and users did not necessarily see the “hits” feedback in the elevator of the scrollbar, indicating off-screen results. This would lead to users not knowing if a search had completed or if it had ever been invoked at all. Adding an hour-glass style piece of feedback or scrolling the user to the most temporally proximal search result would alleviate user confusion in these instances. Clearly these usability issues affected most tasks, though again the search usability issues were more prone to detract from the usefulness of that functionality in DateLens. And, once again, because of issues with search, participants often chose to use a combination approach of searching plus view adjustments, or moved to using more view adjustments to navigate

rather than relying solely on search. In general, this set of users did not browse to complete their tasks.

5. CONCLUSION

From these two studies, we see that DateLens has strong potential. It offers better performance for the more complex tasks that we looked at, many users preferred it over the traditional static calendar design, and it offers the potential of having a single design that scales across device sizes. Furthermore, these positive outcomes came in comparison to a refined shipping product. DateLens is still rough around the edges, being a research prototype. Based on the observed usability problems, we are confident that DateLens could perform better than it did in the studies. In fact, there is no conceptual reason that DateLens could not support the “simple” tasks as well as the Microsoft calendar, yielding a solution that is better in all cases. In fact, after we ran the studies, we modified DateLens to better support the style of use we observed. For example, we added support for making the single day view the default, we added toolbar buttons at the bottom to jump directly to day, week, month, or 3 month views, and we added a navigation bar at the top to directly jump to different days of the week.

Given our experience with DateLens, what can we say are DateLens’ strengths? They appear to be what we hypothesized—the support of complex tasks such as those associated with longer lengths of time. One of the primary reasons for the difficulty of these tasks is that they require comparison between multiple dates, and an understanding of a large expanse of time. In order to do these things, traditional calendar designs require users to switch between the different views (day, week, month, and year)—keeping the relationship between dates *in their heads*. That is, we believe that traditional calendar designs require greater use of short term memory to solve more complex tasks.

By showing a visual representation of dates in context with animated transitions, users appear to be able to compare multiple dates across larger expanses of time with reduced memory load. This translates into shorter task times and subjective preference for some users. While we saw a smaller benefit in the second study, it is possible that this is due to the extreme familiarity with the traditional static interface among the study participants. In addition, by performing a longitudinal study, we relied on the participants to use their own devices—which had varying resources, and some had older machines on which DateLens performed more slowly. Finally, the 2nd study participants lived primarily by the “day” view, and claimed to not use their Pocket PC much for scheduling tasks. This may be partially because Microsoft employees work in a very technology rich environment, and when scheduling was required, a full size computer was frequently available. Thus, we may have tapped into a user population that might not be served well by the added flexibility in DateLens. The first study participants, on the other hand, reported needing this kind of support for performing their daily calendar tasks.

Since managing one’s calendar is so important, many users will be cautious about adopting non-traditional interfaces. Thus, one of the biggest remaining challenges is to refine DateLens so that it is appreciated by a broad spectrum

of users. This is a significant lesson of DateLens: for end-user applications, it is not enough to use visualization to improve performance on some tasks as performance must not be hurt on other tasks. Given the complexity and novelty of visualizations, this is a daunting goal, but one that we all must strive for.

Based on our experience and the results of the user studies we performed, we offer the following general design guidelines for builders of calendar interfaces:

- Support complex tasks by reducing the user's short-term memory load. DateLens does this by using a fisheye view that enables users to see larger and more custom sets of dates, as well as detailed date information in context of the surrounding calendar.
- Help users understand view transitions by using animation. While we have not formally studied the effectiveness of animation in DateLens, our experience points to its utility, and our previous work has shown the benefit of animation in similar situations [Bederson and Boltman 1999].
- Offer an easy way to switch between views and the ability to "live" in different views. Many of our users preferred to keep DateLens in the single day view between uses, but others liked to keep it in the month view and navigate from there.
- Offer users the ability to configure the interface to control the visualization. While we attempted to design DateLens to present information as clearly as possible, we observed users with different appointment characteristics than we had designed for.
- Offer the option of showing search results integrated into the calendar view. We received many positive comments about this ability in DateLens.
- For PDAs, create interaction targets that are as large as possible so that users can use the calendar with their fingers instead of a stylus. This remains a difficult but important challenge when the goal of the visualization is to show as much information as possible. A corollary to this guideline is to allow the user to adjust the font size to be appropriate in novel visualizations as well.

What is perhaps the most interesting question is also the most difficult. That is, can we now offer any advice about where fisheye distortion displays are likely to be useful more generally—outside of the domain of calendars? We can answer this only with our intuition, and do as follows. We found that key characteristics of our solution that made DateLens work were:

- A simple navigation mechanism. All nodes are visible, and a tap always expands a node.
- A simple representational structure. The familiar tabular representation of calendars made a grid a natural representation for us to follow. This rectilinear structure seems to be easy for people to understand. There have been reports that fisheye distortion of other structures can be disorienting to users. We didn't hear a single comment of this nature from either user study.
- Clear small representations of data. A basic characteristic of fisheye views is that sometimes, data must be presented in a very small space. Thus, it is crucial that all data has some meaningful representation that can fit in a

tiny space. We found calendar data to work well in this regard, and we found several natural and meaningful small representations. The smallest of which was just the date, which fit in a 12×12 pixel box.

While it is certainly possible that fisheye applications that don't have these characteristics will also be useful, we believe that applications that do focus on these traits have a higher likelihood of success.

6. FUTURE WORK

There are several areas of future work for DateLens. While technically, we are pleased with DateLens' performance and robustness, there is clearly room for design improvement. We found a number of usability issues during the user studies that must be addressed. Naturally, there is also a long list of features that users have asked for that must be looked into as well, such as support for faster data entry. Understanding how these changes affect users and keeping DateLens easy enough for novice users to feel comfortable with will be an ongoing and crucial challenge.

More user studies must be run after the next round of usability improvements have been made to develop a deeper understanding of the potential effectiveness of fisheye visualizations. There clearly are a number of usability issues in the current implementation of DateLens which got in the way of its effectiveness in the current studies. In addition, we would like to design a specific study to tease apart the individual influences of integrated search and the flexible, fisheye visualization to complex tasks. We would also like to know more about the frequency that users typically perform the more complex tasks that DateLens supports well.

Finally, further design issues are likely to come up when the DateLens interface is applied to smaller devices (such as cell phones) and larger ones (such as tiled displays). While the basic paradigm scales nicely, there are likely to be specific details that need to be changed for different sized displays.

ACKNOWLEDGMENTS

We greatly appreciate the comments of our colleagues through the many revisions of DateLens. We particularly appreciate John SanGiovanni who has been a tireless supporter of this project at Microsoft Research and has helped us through many iterations of the software. We also appreciate the efforts of Neema Moraveji, a University of Maryland student who helped us to initially integrate DateLens with Outlook.

Finally, we thank Susan Wilhite for her help in running the first user study, along with Ben Shneiderman, Catherine Plaisant and Hilary Hutchinson for their comments on drafts of this paper.

REFERENCES

- BAEZA-YATES, R. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. New York: ACM Press.
- BEDERSON, B. B. 2000. Fisheye Menus. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2, 2, 217–225.

- BEDERSON, B. B. AND BOLTMAN, A. 1999. Does Animation Help Users Build Mental Maps of Spatial Information? In *Proceedings of Information Visualization Symposium (InfoVis 99)* New York: IEEE, 28–35.
- BEDERSON, B. B., CLAMAGE, A., CZERWINSKI, M. P., AND ROBERTSON, G. G. 2003. A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2003)*. ACM Press, 618–619.
- BEDERSON, B. B., MEYER, J., AND GOOD, L. 2000. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters* 2, 2, 171–180.
- BJÖRK, S., HOLMQUIST, L. E., LJUNGSTRAND, P., AND REDSTRÖM, J. 2000. PowerView: Structured Access to Integrated Information on Small Screens. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2000)*. ACM Press, 265–266.
- BJÖRK, S., HOLMQUIST, L. E., REDSTRÖM, J., BRETAN, I., DANIELSSON, R., KARLGREN, J., AND FRANZÉN, K. 1999. WEST: A Web Browser for Small Terminals. *UIST 99, ACM Symposium on User Interface Software and Technology, CHI Letters* 1, 1, 187–196.
- FURNAS, G. W. 1986. Generalized Fisheye Views. In *Proceedings of Human Factors in Computing Systems (CHI 86)*. ACM Press, 16–23.
- FURNAS, G. W. 1991. *The Fisheye Calendar System*. (Report No. TM-ARH-020558). Bellcore, Morristown, NJ.
- HILL, W. AND HOLLAN, J. 1994. History-Enriched Digital Objects: Prototypes and Policy Issues. *The Information Society* 10, 2, 139–145.
- INXIGHT. 2003. <http://www.inxight.com>.
- LAMPING, J., RAO, R., AND PIROLI, P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of Human Factors in Computing Systems (CHI 95)*. ACM Press, 401–408.
- LEWIS, J. R. 1995. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instruction for Use. *Int. J. Hum.-Comput. Inter* 7, 1, 57–78.
- MACKINLAY, J. D., ROBERTSON, G. G., AND DELINE, R. 1994. Developing Calendar Visualizers for the Information Visualizer. In *Proceedings of User Interface and Software Technology (UIST 94)*. ACM Press, 109–118.
- PLAISANT, C., AND SHNEIDERMAN, B. 1992. Scheduling Home Control Devices: Design Issues and Usability Evaluation of Four Touchscreen Interfaces. *Int. J. Man-Machine Studies* 36, 375–393.
- RAO, R. AND CARD, S. K. 1994. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proceedings of Human Factors in Computing Systems (CHI 94)*. ACM Press, 318–322.
- SARKAR, M., AND BROWN, M. H. 1992. Graphical Fisheye Views of Graphs. In *Proceedings of Human Factors in Computing Systems (CHI 92)*. ACM Press, 83–91.
- SCHAFER, D., ZUO, Z., BARTRAM, L., DILL, J., DUBS, S., GREENBERG, S., AND ROSEMAN, M. 1997. Comparing Fisheye and Full-Zoom Techniques for Navigation of Hierarchically Clustered Networks. In *Proceedings of Graphics Interface (GI 97)*. Canadian Information Processing Society, 87–96.
- SPENCE, R. AND APPERLEY, M. 1982. Data Base Navigation: an Office Environment for the Professional. *Behaviour & Information Technology* 1, 1, 43–54.

Received March 2003; revised August 2003; accepted October 2003

AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices

Amy K. Karlson, Benjamin B. Bederson

Human-Computer Interaction Lab
Computer Science Department
Univ. of Maryland, College Park, MD, 20742
{akk, bederson}@cs.umd.edu

John SanGiovanni

Microsoft Research
One Microsoft Way, Redmond, WA 98052
johnsang@microsoft.com

ABSTRACT

We present two interfaces to support one-handed thumb use for PDAs and cell phones. Both use Scalable User Interface (ScUI) techniques to support multiple devices with different resolutions and aspect ratios. The designs use variations of zooming interface techniques to provide multiple views of application data: AppLens uses tabular fisheye to access nine applications, while LaunchTile uses pure zoom to access thirty-six applications. We introduce two sets of thumb gestures, each representing different philosophies for one-handed interaction. We conducted two studies to evaluate our designs. In the first study, we explored whether users could learn and execute the AppLens gesture set with minimal training. Participants performed more accurately and efficiently using gestures for directional navigation than using gestures for object interaction. In the second study, we gathered user reactions to each interface, as well as comparative preferences. With minimal exposure to each design, most users favored AppLens's tabular fisheye interface.

Author Keywords

One-handed, mobile devices, gestures, notification, Piccolo, thumb navigation, Zoomable User Interfaces (ZUIs).

ACM Classification Keywords

H.5.2. User Interfaces: Input Devices and Strategies, Interaction Styles, Screen Design; D.2.2 User Interfaces; I.3.6 Interaction Techniques

INTRODUCTION

The current generation of mobile computing hardware features a variety of devices for interaction with the system software. One such class of devices, typically referred to as “smartphones”, features a numeric keypad or miniature thumb keyboard for input together with a screen for display

output. These devices allow for single-handed interaction, which provides users with the ability to place calls and access information when one hand is otherwise occupied. However, because smartphones lack a touch-sensitive display, user interaction is constrained to a discrete set of keys, and thus the options for interaction design are limited to keypad-mapped functions and directional navigation. Another design approach, typically classified as a “Personal Digital Assistant” (PDA) features a touch-sensitive display surface designed primarily to be used with an included stylus. This design offers greater software design flexibility, but many of the small targets designed for a stylus are too small for fingertip actuation, making one-handed use difficult or impossible.

Our goal is to create a new single-handed interaction system for both smartphone and PDA devices. In this work, we have focused on the problem navigating device applications, and have adopted two design strategies. The first leverages Scalable User Interface (ScUI) techniques [2,3] to allow the system to adapt to different screen resolutions as well as support both portrait and landscape device rotation. This architecture allows developers to create a single application that can target a wide variety of screen resolutions, and provides users with a consistent interface and interaction model across devices. We accomplish this using the University of Maryland's Piccolo.NET development toolkit for zoomable and scalable user interfaces [5,18].

Our second design strategy provides access to rich notification information from multiple applications. Most current PDA interfaces are designed for focused interaction with a single task or application, with limited consideration or display real estate allocated for notifications (e.g., email, appointment reminders) or monitoring of ambient information streams (e.g., stocks, sport scores). In our proposed designs, each application has a dynamic launch tile in the place of a static launch icon. This feature offers high-value at-a-glance information for several applications at once, as well as on-demand application launch when users desire more detailed information.

In the work presented here, however, we limit our discussion of scalability and notification in favor of emphasizing the design features relevant to one-handed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.
Copyright 2005 ACM 1-58113-998-5/05/0004...\$5.00.

interaction. We proceed by describing two designs: AppLens (characterized by zoom+fisheye) and LaunchTile (characterized by zoom+pan). The two approaches employ variations of zooming interface techniques [1] to overview several notification tiles, each roughly the size of a postage stamp. AppLens uses a tabular fisheye approach to provide integrated access to and notification for nine applications. LaunchTile uses pure zooming within a landscape of thirty-six applications to accomplish the same goals.

Fisheye and pure zoomable techniques both offer promise, but there are no clear guidelines as to when each approach should be used. Our approach in this work, therefore, has been to design and build the best interfaces we could with roughly the same functionality, and then compare and contrast the results. In this way, we hope to understand which design direction makes the most sense for this domain and to learn something about the trade-offs between the two approaches.

For device interaction when using a touch-sensitive screen, both designs utilize a gestural system for navigation within the application's zoomspace. While our designs do not directly address one-handed text entry, they are compatible with a variety of existing single-handed text input techniques, including single- and multi-tap alphanumeric keypad input, as well as miniature thumb keyboards and unistroke input systems executed with a thumb (e.g., Graffiti [6], Quikwriting [17]).

RELATED WORK

Gestures have proven a popular interaction alternative when hardware alone fails to effectively support user tasks, typical of many nontraditional devices, from mobile computers to wall-sized displays [9]. Gestures can be very efficient, combining both command and operand in a single motion, and are space-conserving, reducing the need for software buttons and menus. However, the invisible nature of gestures can make them hard to remember, and recognition errors can negatively impact user satisfaction [15]. Recent research efforts pairing gestures with PDA-sized devices have emphasized gestures based on changes in device position or orientation [10,21,28]. However, our work more closely resembles the onscreen gestures that have played a prominent role in stylus-based mobile computing (e.g., Power Browser [7]).

Our thumb-as-stylus designs support usage scenarios in which only one hand is available for device operation, such as talking on the phone or carrying a briefcase. Although existing stylus-based gesture systems do not preclude the use of the thumb, we are not aware of any systems that have been specifically designed for the limited precision and extent of the thumb. The EdgeWrite [32] gesture-based stylus text entry system is particularly suited to mobile usage scenarios due to its use of physical barriers. EdgeWrite adaptation to one-handed mobile computing is compelling, but would require expanding the dedicated input area to accommodate thumb-resolution gestures.

One handed device interaction has typically focused on text entry techniques, but beyond a numeric keypad, most one-handed text entry systems require specialized hardware, such as accelerometers [25,30] or customized keyboards [16]. We instead address the more general task of system navigation and interaction, and restrict our designs to existing hardware.

Commercial products have also emerged with related design goals. The Jackito PDA [14] supports thumb-only interaction, but presumes two handed use and is not gesture oriented. Lastly, Apple's iPod is an elegant one-handed interaction solution for audio play and storage [13], but is currently not designed for the type of generalized application interaction we propose.

THE ZOOM+FISHEYE APPROACH: APPLENS

AppLens provides one-handed access to nine applications, and is strongly motivated by DateLens, a tabular fisheye calendar [3]. We refer to AppLens as a "shell" application for its role in organizing and managing access to other applications.

Generalized Data Access Using Tabular Fisheyes

Spence and Apperley [27] introduced the "bifocal display" as one of the first examples of fisheye distortion applied to computer interfaces. Furnas extended the bifocal display to include cognitive perceptual strategies and introduced a set of analytical functions to automatically compute generalized fisheye effects [8]. Since then, fisheye distortion has been applied with mixed success across a variety of domains, including graphs [24], networks [26], spreadsheets [20], and documents [12,23].

Bederson et al. [3] drew upon that work in developing DateLens, a space-conserving calendar for PDAs, which was shown to perform favorably for long-term scheduling and planning tasks when compared to a traditional calendar implementation. One of the strengths of DateLens was the pairing of distortion and scalability, which allowed the details of a single day to be viewed in the context of up to several months of appointment data. Also important was the design of effective representations for the variety of cell sizes and aspect ratios that resulted from tabular distortion. One drawback of DateLens, however, was that it required two-handed stylus interaction to actuate the small interface widgets. Our design extends the principles of DateLens to include one-handed thumb access and generalizes the approach for use across a variety of domains.

We developed a scalable framework that provides a grid, tabular layout, and default views for cell contents at a variety of sizes and aspect ratios. We also developed a general API to make it simple for applications to be built within this framework; developers need only to replace a small number of cell views with representations that are meaningful within the target domain.



Figure 1. AppLens Zoom Levels: (a) Notification, (b) Full, (c, d) Context.

AppLens Zoom Levels

The AppLens shell (Figure 1) has been implemented within our generalized tabular fisheye framework, using a 3x3 grid, and assigning one of nine applications to each cell. The support for tabular layout includes representations at 3 zoom levels: *Notification*, *Context* and *Full*.

Notification zoom distributes the available screen real estate equally among the 9 application tiles (Figure 1a). One tile (shown centered) remains reserved for settings, which can be used to configure the selection of applications which occupy the other 8 notification tiles. Generally, tiles at *Notification* size display high level static and/or dynamic application-specific notification information.

Context zoom (Figure 1c) allocates roughly half the available display area to a single *focus* tile, compressing the remaining tiles according to a tabular fisheye distortion technique [3,20]. A tile at *Context* size typically appears much like a fully functional application, but selectively displays features to accommodate display constraints, and is not interactive. Tiles on the periphery of a *Context* zoom, called *peripheral*

tiles, may be rendered quite differently depending on their position relative to the focus tile, which dictates whether the peripheral tile is a square, a wide-flat rectangle, or a narrow-tall rectangle. To reduce visual overload, peripheral tiles are displayed at 40% transparency. The contents of distorted peripheral tiles are not themselves distorted, but rather change representation to provide the most meaning in the space available.

The third and final *Full* zoom level expands a tile to a fully interactive application that occupies 100% of the display (Figure 1b).

Gesture-Based Cursor Navigation

Existing application designs for PDAs are often inappropriate for one-handed use due to their reliance on screen regions that typically cannot be reached while maintaining control of the device (e.g., accessing the Start menu in the upper left-hand corner of a display while holding the device in the right hand), and the use of standard widgets that are too small for reliable thumb use (e.g., radio buttons, checkboxes, and on-screen keyboards). In support of traditional designs, AppLens uses an object cursor to identify the on-screen object that is the current interaction target. The cursor is depicted as a dynamically-sized rectangular orange border that users move from one on-screen object to another via command gestures. Cursors are not new to PDA interface design: the WebThumb [31] web browser includes a similar notion of cursor, but which is controlled via directional hardware, and others [29] have explored device tilting to manipulate PDA cursors.

Neither the cursor nor gestures interfere with the most common stylus interactions of tap and tap+hold. Although gestures do overlap stylus drag commands, dragging is rarely used in handheld applications and could be distinguished from gestures by explicitly setting a gesture input mode.

We established a core set of commands that would allow users to navigate applications using only the input cursor. Our command language supports directional navigation (UP, DOWN, LEFT, RIGHT) as well as two widget interaction commands: one equivalent to a stylus tap (ACTIVATE), and the other which negates widget activation (CANCEL), equivalent to tapping the stylus outside the target widget. We also include the convenience commands FORWARD and BACKWARD, equivalent to TAB and SHIFT-TAB on Windows PCs.

Command Gestures

Our use of gestures is motivated by Hirota's observation that the button positions on cell phones require interaction using a low, unstable grip [11]. PDA joysticks face a similar drawback in that they are typically located along the lower perimeter of the device. AppLens avoids this problem since its gestures can be issued anywhere on the screen. Each AppLens gesture is uniquely defined by a slope and direction, or vector, which allows gestures to be robust and highly personalizable; users can issue gestures of any length (beyond an activation threshold of 20 pixels) anywhere on

the touch-sensitive surface. This flexibility lets users interact with our designs using the grasp that provides maximum stability in a mobile scenario.



Figure 2. Screen area accessible with one hand.

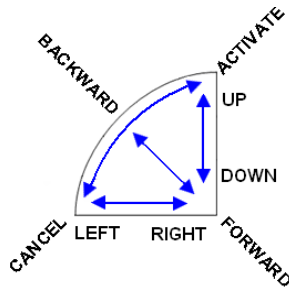


Figure 3. The gesture set.

We based the gesture set on the limited motion range of thumbs (Figure 2), with the goal of creating a gesture language that could be learned with minimal training. After informally experimenting with a variety of gestures, we developed a simple set of gestures with the aim of maximizing memorability and robustness of execution (Figure 3). We assigned the directional commands UP, DOWN, LEFT and RIGHT to spatial gestures that map directly to their function. We assigned ACTIVATE and CANCEL to the two gestures defined by pivoting the thumb from bottom to top and top to bottom respectively. We made these assignments both to reinforce their opposing relationship as well as for ergonomic ease in issuing common commands. Finally, we assigned the upper-left to lower-right diagonal to FORWARD due to its relative forward nature, and by similar reasoning, the reverse gesture to BACKWARD.

The eight gesture commands can also be activated with a numeric keypad by mapping each to the key corresponding to the gesture endpoint: 1-BACKWARD, 3-ACTIVATE, 7-CANCEL, and 9-FORWARD. Since smartphones have a joystick that can be used for directional navigation, the keypad-to-command mapping is not necessary, but can be assigned as follows: 2-UP, 4-LEFT, 6-RIGHT and 8-DOWN.

Using Command Gestures within AppLens

Users navigate between AppLens zoom levels using ACTIVATE and CANCEL gestures. As a rule, the ACTIVATE gesture behaves as a stylus tap on the current cursor target, thus its effects are target-specific. Since application tiles are non-interactive at *Notification* and *Context* zoom levels, the ACTIVATE gesture simply zooms in, animating the layout first from *Notification* to *Context* zoom, and then to *Full* zoom. Once at *Full* zoom, the input cursor transitions to the objects *within* the application, at which point the command gestures affect the current target widget. The CANCEL command negates the effects of the ACTIVATE command. At *Full* zoom, the effects of the CANCEL command depend on the location of the cursor and the state of its target. The CANCEL command will first deactivate an active target. If the current target is not in an active state, CANCEL will cause the

application tile to animate from *Full* zoom to *Context* zoom, and if issued again, to *Notification* zoom.



Figure 4. Three LaunchTile zoom levels: (a, b) Zone, (c) World, (d) Application

THE ZOOM+PAN APPROACH: LAUNCHTILE

Our second design, LaunchTile proposes another way to interact with a grid of notification tiles. The primary shell of the LaunchTile design is an interactive zoomspace consisting of 36 application tiles, divided into 9 zones of 4 tiles each (Figure 4c). The 36 tile configuration is an exploration of the maximum number of applications that can reasonably fit within the display space. Since the design is fundamentally scalable, however, it can display any number of tiles *up to* 36, and fewer may even be preferable. As a central design element, LaunchTile uses a large blue onscreen button, called Blue (Figure 4a), to unify the shell and applications with a consistent visual metaphor. Blue provides a consistent point of reference for zooming and panning navigation, with onscreen tiles, menus, and functions maintaining a consistent relative position to Blue. The LaunchTile zoomspace consists of 3 zoom levels: World (36 tiles, Figure 4c), Zone (4 tiles, Figures 4a and b) and Application (1 tile, Figure 4d).

Zone View

The Zone view of LaunchTile divides the screen area into 4 equally-sized application tiles (Figure 4a). To view other tiles, the user pans the zoomspace to neighboring 4-tile clusters, called *zones*. The zones are arranged as a 3x3 grid, each associated with a numerical designator from 1 to 9 as on a conventional telephone keypad. Zone 5 is the center zone, which defines the Home screen, and shows the 4 highest priority notification tiles, as configured by the user.

Panning Techniques

To support various input hardware and styles of interaction, there are several ways to pan the zoomspace within Zone view. If the device has a multidirectional control joystick, the user can push it in the direction of the targeted zone. From the Home screen, the 16 tiles in zones 2, 4, 6, and 8 are only a single tap away. As many directional pads do not support diagonal action, the 16 additional tiles in the corner zones 1, 3, 7, and 9 are two taps away. Alternatively, if the device has a touch-sensitive display, the user can use his or her thumb directly to drag the zoomspace. Dragging is performed “on rails”, permitting the user to drag vertically and horizontally, but not diagonally. Although the zoomspace moves with the thumb during dragging, Blue remains centered and stationary. Because only one instance of Blue exists within Zone view, each zone is depicted with an empty center hub during dragging. Upon thumb release, the zoomspace animates to align Blue with the closest zone’s empty hub. The visual and automated guidance ensures the user is never caught between zones.

Within each 4-tile zone, indicator widgets communicate the user’s relative location within the zoomspace. The indicator has two components. First, directional arrows show where other zones are. If users only see indicators pointing up and right, they know they are in zone 7. Small blue dots featured with the arrows represent the location of all zones not currently in view. The blue dots could also be used to indicate an alert or status change in a neighboring zone, though this feature was not implemented in our prototype. The final way to pan the zoomspace is to tap the directional indicator itself. An oversized hit target ensures that the user can easily hit the indicator without using a stylus.

Zooming Out to the World View

From any one of the nine 4-tile Zone view zones, the user may tap Blue (or press the 5 key) to zoom out to view the entire 36-tile zoomspace (Figure 4c). Since all 36 tiles are visible at once, this view reduces each tile to a small icon. From this World view, the user can easily see the absolute location of each tile, as well as monitor all applications at once. In the World view, the display is divided into a grid of 9 hit targets, each mapping to a 4-tile zone. Single-tapping a zone animates to Zone view, displaying the zone’s 4 notification tiles.

Zooming In to an Application

The user taps any of the 4 notification tiles within Zone view to launch the corresponding application. An animated zoom draws the zoomspace toward the user until the target application fills the entire display (Figure 4d). If the device has only a numeric keypad (no touchscreen), the user presses the numeric key in the corner that corresponds to the zone. Pressing 1 launches the upper left tile, 3 launches the upper right, 7 launches the lower left, and 9 launches the lower right. This technique provides quick, single-tap access to each visible tile, and was inspired by ZoneZoom of Robbins et al. [22].

As the system zooms, Blue stays in view and retains its function as a central point of reference (Figure 4d). Application menu commands are represented as on-screen buttons clustered around Blue, now positioned at the bottom of the display. Each menu button displays a numeric label, so that mobile phone users may activate each menu by pressing the corresponding number on the keypad. A visual indicator to the left of the zoomspace animates during interaction to reflect the user’s current absolute zoom level within the LaunchTile environment.

Zoom Control

Pressing Blue typically moves the user deeper into the object hierarchy, while a dedicated Back button moves the user up the hierarchy. In the Zone view however, Blue toggles between Zone view (4 tiles) and World view (36 tiles). Once an application is launched, three dedicated software buttons along the top edge of the screen support inter- and intra-application navigation (Figure 4d). A green Home button immediately zooms the view out to the Home screen. There is also a Back button on the upper right edge of the screen, and another global command key. The placeholder for this function in our prototype is an icon for voice command and control. On a non-touchscreen device, Back and Home commands are executed with dedicated physical buttons, such as those provided on a smartphone.

Application-Level Interaction

Although the original focus of our designs was on the application “shell”, we extended the LaunchTile interaction philosophy to the application level, where we sought to make interaction consistent with navigation among the application tiles. Several gestures have been designed to specifically support one-handed navigation and item selection within applications. Previously, others have demonstrated that for direct-manipulation interfaces, a grounded tap-drag-select-release technique is more accurate than a tap-to-select [19]. We therefore made all LaunchTile tap-to-select targets large enough for thumb activation. In cases when limited display real estate necessitates smaller targets, the central Blue widget serves as a moveable tool glass which can be positioned over the target object (e.g., email header, message text). The large thumb-friendly drag target is offset below the selection area to prevent the user’s thumb from occluding the target. Alternatively, the user can drag the application

contents, such as to pan a map, scroll a list of email, or navigate to text outside the view area. Together, these two interaction techniques permit the user to address a large application content space, yet bring focus to a target much smaller than a finger. Alternatively, a non-touchscreen user uses the multidirectional joystick to position the selection area on the screen. Keys 2 and 8 are used for page control up and down, and 4 and 6 can be either menu items or horizontal page control, as appropriate.

Once the targeted item is in the tool glass selection area, a tap on Blue (or pressing the 5 key) activates the item, which may either draw the user further into the application (e.g., open an email), or may replace the application menus with a context-sensitive menu clustered immediately around Blue. As with the application menus, keys on a numeric keypad execute context menus according to their positions relative to Blue. At this point a second tap on Blue sends the user to the deepest level of the zoomspace, text input. At this level, a modular text input object called an InputTile appears around the selection area for alphanumeric input. Alternatively a mini qwerty keyboard may be used for directly entering text. With this design, a double tap on Blue while text is selected will bypass the context menu and take the user directly into text edit mode.

IMPLEMENTATION

The AppLens and LaunchTile prototypes have been built using the University of Maryland's PocketPiccolo.NET development toolkit for Zoomable User Interfaces (ZUIs) [5,18]. Although the primary development and test platform has been the HP iPAQ PocketPC running Microsoft Windows Mobile 2003, both run unmodified on the iMate Smartphone II running Windows Mobile Smartphone 2003 (Figures 1d and 4b respectively).

Although the core architecture and gesture recognition for each shell has been implemented, applications have been simulated with images. This has allowed us to put designs that are faithful to the look and feel of each shell in the hands of users for early feedback, but falls short of full interactivity. However, because LaunchTile design principles extend to the applications themselves, we have prototyped email and mapping as two interactive examples within LaunchTile.

STUDIES

We conducted two studies to inform future research directions. The first study explored whether users could learn and perform the AppLens gesture set with only minimal training. The second was a formative evaluation which gathered user reactions to each shell design, as well as users' comparative preferences.

APPLENS GESTURE STUDY

A significant difference between the gesture designs of each shell is that all gestures in LaunchTile are direct manipulation, while all gestures in AppLens are remote-control commands. In LaunchTile, users physically drag objects with their thumbs, be it the zoomspace, a tool glass, a

map, or a list of email headers. In AppLens, gestures only affect the cursor target, rather than the objects over which the gesture is performed. The success of AppLens therefore depends as much on the utility of the tabular design as the ability for users to execute the gesture command set. Participants were tested on gesture execution performance and efficiency in navigating an information space.

PARTICIPANTS: 20 participants (12 male, 8 female) were recruited from the general population with the only selection criteria being that participants were right-handed. The median participant age was 30, and although 12 of the participants considered themselves advanced computer users, only 6 regularly used a PDA.



Figure 5. Example gesture study environment states: (a) the top level, (b) the third level after ACTIVATE is performed on 6.5, (c) highlighting within a cell.

MATERIALS: The study was run on an HP iPAQ measuring 4.5x2.8x0.5 inches with a 3.5 inch screen. We constructed a software test environment modeled after AppLens: a hierarchical tabular information space, with each level in the hierarchy represented by a 3x3 grid of equal-sized cells, numbered 1-9 like a phone keypad. For the experiment, the hierarchy was limited to 5 levels and *Context* zoom was eliminated so that activating a cell animated the display to the next level. Cell contents served as navigational landmarks, each labeled with a hierarchical address constructed by prepending the cell's local logical number (1-9) with its parent's label, separated by a dot (Figure 5b and 5c). We reserved an area at the top of the screen for task instructions, and disabled tapping to restrict input to gestures alone. Gestures retained their meaning from AppLens: Navigational gestures LEFT, RIGHT, UP and DOWN controlled the movement of an orange rectangular cursor within a level; ACTIVATE zoomed in to the next level and CANCEL zoomed out to the previous level. Within the context of the test environment, FORWARD and BACKWARD were used for intra-cell selection, allowing participants to cycle a highlight either forward or backward through cell label digits (Figure 5c). Highlighted digits could then be bolded or "activated" using the ACTIVATE gesture or un-bolded or "deactivated" using the CANCEL gesture. Participants were provided with a reference sheet that described the hierarchical information space and labeling scheme, as well as the eight gestures to be used for navigation and interaction.

TASKS: Participants performed two types of tasks. Gesture tasks required users to perform a gesture when presented with the associated command name. Navigation tasks required participants to navigate to a particular cell in the hierarchy, and/or to activate or deactivate a cell label digit.

MEASURES: Application logs recorded task time, the gestures performed for each task, and whether the task was completed successfully. Participants were instructed to press a hardware button between tasks, which served to record task time and advance to the next task. Due to a bug in our logging software, task time was recorded at second rather than millisecond resolution. However, since our goal was to identify performance trends rather than comparison to another input method, one second resolution was sufficient. Participants rated their experience using five nine-point Likert scales: frustrating – satisfying, hard to learn – easy to learn, hard to use – easy to use, slow – fast, and boring – fun.

PROCEDURE: After reading a description of the test environment and gestures, participants performed 16 practice tasks similar in nature to those in the navigation task phase. Participants practiced for 5-10 minutes.

After the practice phase, participants performed gesture tasks: presented with a command name, participants performed the associated gesture and pressed a hardware button to advance to the next task. Command names were presented to participants in random order, four times for each of the eight commands. The gesture reference sheet was placed face down so that the administrator could record the number of times it was referred to.

The second test phase required participants to perform goal-directed tasks within the information space. These included (N)avigation (“Navigate to 6.5.4”), (A)ctivation (“Activate the 5 in 6.5.4”), (NA)vigation+activation (“Activate the 2 in 4.3.2”), and (C)ancellation (“De-activate all digits in 4.3.2”) tasks. Participants then recorded their subjective ratings of the interaction experience. Some participants completed the study in as little as 15 minutes, most within 30 minutes, and none required more than 40 minutes.

Results

In the gesture phase of the study, participants correctly performed gestures an average of 87% of the time. By gesture type, participants correctly performed directional gestures 93% of the time, but had more difficulty with diagonal gestures ACTIVATE and CANCEL at 88% and 85% respectively. BACKWARD and FORWARD had the worst success rate, at 70% and 64% of the time respectively. Time to perform gestures followed a similar trend. On average, participants required 2.4 seconds to perform each gesture: 1.5 – 1.7 seconds for directional gestures, 2.6 – 2.8 seconds for ACTIVATE and CANCEL gestures, and 3.6 – 3.7 seconds for BACKWARD and FORWARD gestures respectively. Neither measure correlated with computer or PDA experience. Although we tallied the number of times that participants looked at a reference sheet, we assumed that the acts of page-

flipping and answer-searching have been reflected in the task time, and thus did not analyze this data further.

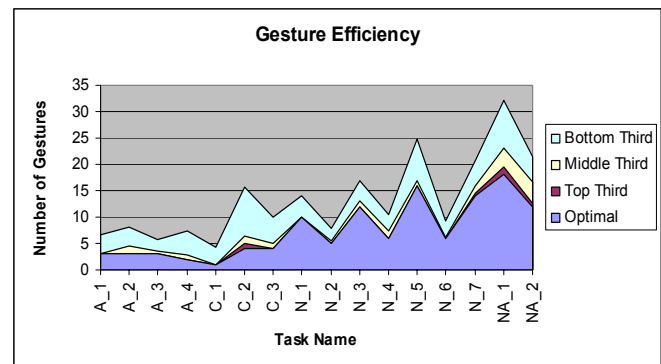


Figure 6. Average number of gestures per task, by performance tier, compared with the optimal.

The second test phase evaluated accuracy and efficiency for goal-directed navigation tasks. The average task success rate was 95%. While both navigation and navigation+activation tasks were performed with 98% accuracy, activation was close behind at 96%. Cancellation tasks, however, were only completed correctly 83% of the time. On average, participants performed 2.4 additional steps than the optimal number to complete a task. Breaking participants into three performance groups for each task, it is clear that one third of participants performed nearly optimally on all but a few tasks (Figure 6). The Middle third of participants performed comparably to the Top third, but the gaps were wider on the tasks the Top third had trouble with. That is, the tasks for which the Middle third deviated farthest from Optimal, were those for which the Top-tier also deviated. Thus, the most significant difficulties in performing the navigation phase tasks were experienced by only a third of the participants.

The average subjective rating for each of our 5 satisfaction measures, on a scale of 1-9 where 9 was positive, fell within a 1 point span of one another, between 5.9 (satisfying) and 6.75 (fun).

Analysis

Because the gesture phase of the study did not distinguish between errors of recall and execution, we could not classify the reasons for error. Analyzing the log files, it is safe to say that errors of both types occurred. The low error and speed measures for directional navigation support our hypothesis that the directional gestures have an intrinsic spatial mapping. Presumably, this mapping contributes to better learnability, more reliable execution, and lower cognitive demand. The positive jump in execution speed for the other four gestures is unsurprising when we consider that the mappings between gesture and command are more abstract than the directional mappings, and likely require more cognitive effort to perform the mental transformation. This alone does not explain the associated increase in error rate, but insights from Long’s work on gesture similarity [15] suggest that users may perceive the diagonal gestures as similar and therefore more difficult to learn.

The difference in performance data between ACTIVATE and CANCEL vs. BACKWARD and FORWARD may be attributed to the more physically challenging nature of the latter two, but may also be due to disproportionate practice time received, considering a single navigation task provided more opportunities to issue ACTIVATE and CANCEL gestures than intra-cell activation tasks provided for FORWARD and BACKWARD. These intuitions also help explain the efficiency results – users were more successful and efficient in pure navigation tasks which contained a proportionally large number of directional gestures compared with intra-cell activation/cancellation tasks. The relative complexity of the gesture navigation environment may have confounded the results by inflating the efficiency measures, but we feel that it also made the results more relevant to the AppLens design.

APPLENS AND LAUNCHTILE FORMATIVE STUDY

Due to the early nature of our interfaces, their shared design principles, and our goal of distilling a unified design, we conducted a formative study to understand usability issues for new users of each shell design and to elicit general reactions and comparative preferences.

PARTICIPANTS: We recruited ten participants (8 male, 2 female) from a local private scientific research center. Three participants were in their 20s, five in their 30s, and two were 40 or older. While all participants considered themselves advanced computer users, four used PDAs regularly, and four had never used a PDA.

MEASURES: Participants provided subjective design-specific and comparative reactions to AppLens and LaunchTile through think aloud and verbal questionnaires.

MATERIALS: The shell prototypes were run on the same hardware used in the first study. A one-page document described the AppLens design and gestures set, followed by a list of eight associated tasks. A two-page document described the LaunchTile design and methods for navigation and interaction, followed by a list of 11 tasks.

TASKS: For each interface, participants performed tasks which were designed to exercise the full range of navigation and interaction features. For example, LaunchTile tasks included navigating to specific zones, finding specific applications, and opening and editing an email message. AppLens tasks included navigating to specific application tiles, and answering questions about application content.

PROCEDURE: Study participants were introduced to each design by reading its design document and performing each of the related tasks. During the tasks, the test administrator recorded think aloud reactions and usability observations. After task completion, the administrator recorded answers to open-ended questions related to the interaction, such as likes and dislikes, features that were easy or hard to use or learn, and comfort level. The same procedure was repeated for the second interface. The administrator balanced the order of the interfaces among participants. After interacting with both interfaces, participants were asked comparative preference

questions. We limited each user session to 45 minutes, allotting roughly fifteen minutes to each interface, and the final fifteen for comparative feedback.

Results

Reactions to AppLens were quite consistent across participants. Because only one question in the study focused on a specific interface design feature (*Context* view), we regard commonality in participant responses indicative of the highest-impact interface characteristics. We report here on the strongest trends in opinion. Half the participants commented that they liked the *Notification* view and the ability to access all nine application tiles within both *Notification* and *Context* views. Even though two participants found nothing redeeming about *Context* view, all others found it useful at least some of the time. Seven participants found application navigation easy and enjoyable, but performed the majority of navigation using tapping rather than gestures. Even so, participants were required to use gestures to zoom out from *Full* zoom, and two participants particularly liked the gestures. Five participants agreed that the gestures were the most difficult aspect of the interface, but disagreed on why, citing confusion over the gestures for zoom-in vs. zoom-out, difficulty performing the ACTIVATE gesture, difficulty with directional navigation, and frustration due to misrecognition. All participants found AppLens both easy to learn and effective for navigating among applications, all but one found one-handed use comfortable, and six out of seven participants stated they would prefer AppLens over their most familiar PDA operating system.

Perhaps due to the richness of the LaunchTile environment, reactions were mixed and more complex. Nearly half the participants reacted positively to the aesthetics of LaunchTile, and specifically to Blue. Seven participants appreciated the volume of information available through the two zoom perspectives World and Zone, while six thought that zooming between those perspectives was one of the easiest aspects of the interface. The majority (7) of participants felt comfortable using one hand to perform the tasks, and eight felt they were able to effectively navigate among applications, primarily by tapping via World view.

Surprisingly, a majority (7) of participants had difficulty panning by dragging within LaunchTile, commenting most often that it was unintuitive, but also that it was slow. This subset of participants was slightly skewed toward participants who used the LaunchTile interface first, and thus would not have been biased by experience with the AppLens directional gestures. Six participants struggled with the multi-modal nature of Blue, unsure about its role in different contexts, especially within applications. A related problem was that of differentiating between the roles of the Home, Back, and Blue buttons from within an application. Most of the participants were tentative and had difficulty performing tasks within the email application. Ultimately, participants were evenly divided (3 vs. 3) about whether they would

choose to use LaunchTile for their own application management - half as many as chose AppLens.

Comparing the two interfaces, most participants recognized the trade off between the number of applications that could be viewed at once and the amount of information conveyed for each, yet seven out of nine participants thought AppLens provided better at-a-glance value. Although nearly half the participants were reluctant to compare the speed of information access between the two interfaces due to the differing amounts of information available, seven out of nine participants thought AppLens supported faster data access, presumably due to a better balance between the number of applications and the presentation space available for each. Additionally, AppLens was considered easier to use (7 out of 9), and 8 out of 9 would prefer AppLens for use on their own device. In response to our general question about the utility of one-handed use, 7 participants thought one-handed interaction would be useful at least some of the time, with 3 of those stating an a priori preference for one-handed use in all situations. However 2 participants expressed that they would never want to use a PDA with one hand, regardless of the interface design.

DISCUSSION

Two notable themes emerged from the comparative study. The first was the participants' reluctance to use gestures. Results from the first study suggested that directional gestures can be learned quickly, yet for both interfaces, users favored tapping targets over issuing gestures. While this trend may simply be a training issue, it may also be an early warning sign for the adoptability of overlay gesture systems - those that share the same real estate as the interface objects themselves. Second, a large number of users commented on the perceived utility of the simultaneous display of high-value content from multiple applications, bolstering our intuition that flexible notification-based designs may provide an effective balance between functionality and content within the real estate constraints of handheld computing.

While AppLens appeared to be the preferred design, we must take care in assigning reasons for the preference. First, AppLens is a simpler design and a shallower prototype than LaunchTile. Its appeal may have been that users felt proficient and better able to manage 9 applications (versus 36) with minimal use. Its simplicity also made AppLens less prone to the performance limitations of the target hardware, which noticeably impacted LaunchTile zooming quality. However, more experience with the two designs might have tipped the scales the other way, as Bederson has pointed out in [4] that even complex interfaces have the potential to be highly satisfying after users have expended the effort to become experts. A vocal minority of expert PDA users who much preferred LaunchTile supported this possibility, citing the large number of applications and configurable layout as very attractive features. Thus a different participant population may have offered different opinions. While we do not consider LaunchTile in this class of expert interface,

clearly 15 minutes is not sufficient for users to become proficient with the variety of interaction techniques supported by the interface.

In [4] Bederson hypothesizes that user satisfaction is related to how well an interface supports "flow", which correlates inversely to the degree to which an interface gets in the way of, or interrupts, user tasks. Blue is an example of a LaunchTile feature that interrupts flow: it performs different functions in different contexts, requiring users to keep track of the system state to predict the outcome of tapping Blue. This type of functional overloading is a well-known design issue, but is commonly used nonetheless. For example, both the Microsoft and Apple desktop media players use a single button for both Play and Pause. Just as with LaunchTile, these designs compromise simpler mappings in favor of a visually simpler design. The difference, however, is that both media players change the button icon to reflect the current state (i.e., the function the button will perform when pressed) so that users don't have to remember the state or deduce the state from less obvious cues. A similar adaptation for Blue may reduce or even eliminate user confusion in the LaunchTile design.

CONCLUSION

Based on our participants' strong interest in one-handed PDA use, and generally positive reactions to their interaction experiences, we are convinced of the value of research in one-handed designs, and believe notification plays an important role in effective utilization of limited real estate. We have less evidence of the utility of design scalability beyond being an engineering convenience. Although we have demonstrated the feasibility of transferring both interfaces to smartphones, we do not know whether the designs support smartphone usage scenarios. With respect to command gestures, we are encouraged by the modest yet positively skewed satisfaction ratings for gesture interaction as well as what we consider very reasonable performance for both directional gesture execution and navigation tasks. It's clear, however, that the introduction of two additional diagonal gestures degrades performance and confuses users. We will need to explore whether AppLens can be an effective interface without these additional commands, or whether a different mapping of commands to gestures or on-screen cues can make the full set of gestures as reliable and learnable as directional gestures seem to be. Finally, we anticipate that extended usage studies with wider populations will unearth more subtle usability issues. With refinement, we hope a single design will emerge to provide a consistent, flexible environment designed for single-handed use.

ACKNOWLEDGMENTS

We appreciate François Guimbretière's early suggestion of considering the ergonomics of human thumbs, and thank Aaron Clamage for his rapid efforts in porting Piccolo.NET to small devices so we could build these prototypes.

REFERENCES

1. Bederson, B. B., Meyer, J. and Good, L. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *Proc. UIST* (2000), 171-180.
2. Bederson, B. B. PhotoMesa: A Zoomable Image Browser using Quatnum Treemaps and Bubblemaps. *Proc. UIST*, ACM Press (2001), 71-80.
3. Bederson, B. B., Clamage, A., Czerwinski, M. and Robertson, G. DateLens: A Fisheye Calendar Interface for PDAs. *ACM Trans. Comput.-Hum. Interact.* 10, 4 (2003).
4. Bederson, B. B. Interfaces for staying in the flow. *Ubiquity* 5, 7 (2004).
5. Bederson, B. B., Grosjean, J. and Meyer, J. Toolkit Design for Interactive Structured Graphics. *IEEE Trans. Soft-Eng.* 30, 8 (2004), 535-546.
6. Blickenstorfer, C. H. Graffiti: Wow! *Pen Computing Magazine* (1995), 30-31.
7. Buyukkokten, O., Garcia-Molina, H., Paepcke, A. and Winograd, T. Power browser: efficient Web browsing for PDAs. *Proc. CHI*, ACM Press (2000), 430-437.
8. Furnas, G. W. Generalized fisheye views. *Proc. CHI*, ACM Press (1986), 16-23.
9. Guimbretiere, F., Stone, M. and Winograd, T. Fluid interaction with high-resolution wall-size displays. *Proc. UIST*, ACM Press (2001), 21-30.
10. Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing techniques for mobile interaction. *Proc. UIST*, ACM Press (2000), 91-100.
11. Hirotaka, N. Reassessing current cell phone designs: using thumb input effectively. *Extended Abstracts CHI*, ACM Press (2003), 938-939.
12. Hornbæk, K. and Frøkjær, E. Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. *Proc. CHI*, ACM Press (2001), 293-300.
13. iPod. www.apple.com/ipod/, Apple Computer.
14. Jackito PDA. <http://www.jackito-pda.com/>.
15. Long, A. C., Landay, J. A., Lawrence, R. A. and Michiels, J. Visual similarity of pen gestures. *Proc. CHI*, ACM Press (2000), 360-367.
16. Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A. and Looney, E. W. Twiddler typing: one-handed chording text entry for mobile phones. *Proc. CHI*, ACM Press (2004), 671-678.
17. Perlin, K. Quikwriting: continuous stylus-based text entry. *Proc. UIST*, ACM Press (1998), 215-216.
18. Piccolo.Net. <http://www.cs.umd.edu/hcil/piccolo/>, Univ. of Maryland Human-Computer Interaction Lab.
19. Potter, R. L., Weldon, L. J. and Shneiderman, B. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI* (1988), 27-32.
20. Rao, R. and Card, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. *Proc. CHI*, ACM Press (1994), 318-322.
21. Rekimoto, J. Tilting operations for small screen interfaces. *Proc. UIST*, ACM Press (1996), 167-168.
22. Robbins, D. C., Cutrell, E., Sarin, R. and Horvitz, E. Zone Zoom: Map Navigation for Smartphones with Recursive View Segmentation. *Proc. AVI*, ACM Press (2004), 231-234.
23. Robertson, G. G. and Mackinlay, J. D. The Document Lens. *Proc. UIST*, ACM Press (1993), 101-108.
24. Sarkar, M. and Brown, M. H. Graphical Fisheye Views of Graphs. *Proc. CHI*, ACM Press (1992), 83-91.
25. Sazawal, V., Want, R. and Borriello, G. The unigesture approach. *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*, Springer-Verlag (2002), 256-270.
26. Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S. and Roseman, M. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans. Comput.-Hum. Interact.* 3, 2 (1996), 162-188.
27. Spence, R. and Apperley, M. D. An Office Environment for the Professional. *Behaviour & Information Technology* 1, 1 (1982), 43-54.
28. Strachan, S., Murray-Smith, R., Oakley, I. and Angelesleva, J. Dynamic Primitives for Gestural Interaction. *Proc. Mobile HCI*, Springer Verlag (2004).
29. Weberg, L., Torbjorn, B. and Hansson, A. W. A piece of butter on the PDA display. *Extended Abstracts CHI* (2001), 435-436.
30. Wigdor, D. and Balakrishnan, R. TiltText: using tilt for text input to mobile phones. *Proc. UIST*, ACM Press (2003), 81-90.
31. Wobbrock, J. O., Forlizzi, J., Hudson, S. E. and Myers, B. A. WebThumb: interaction techniques for small-screen browsers. *Proc. UIST*, ACM Press (2002).
32. Wobbrock, J. O., Myers, B. A. and Kembell, J. A. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. *Proc. UIST*, ACM Press (2003), 61-70.