

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. Kronrod, 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.**, Reville, 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, RSMs 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 • 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://thekojoonnamdishow.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://thekojoonnamdishow.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§ion=
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_Childrens_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

APPENDIX 1

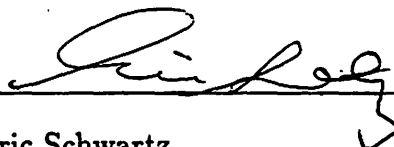
**A Miniature Space-Variant Active Vision System:
Cortex-I**

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: _____



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.

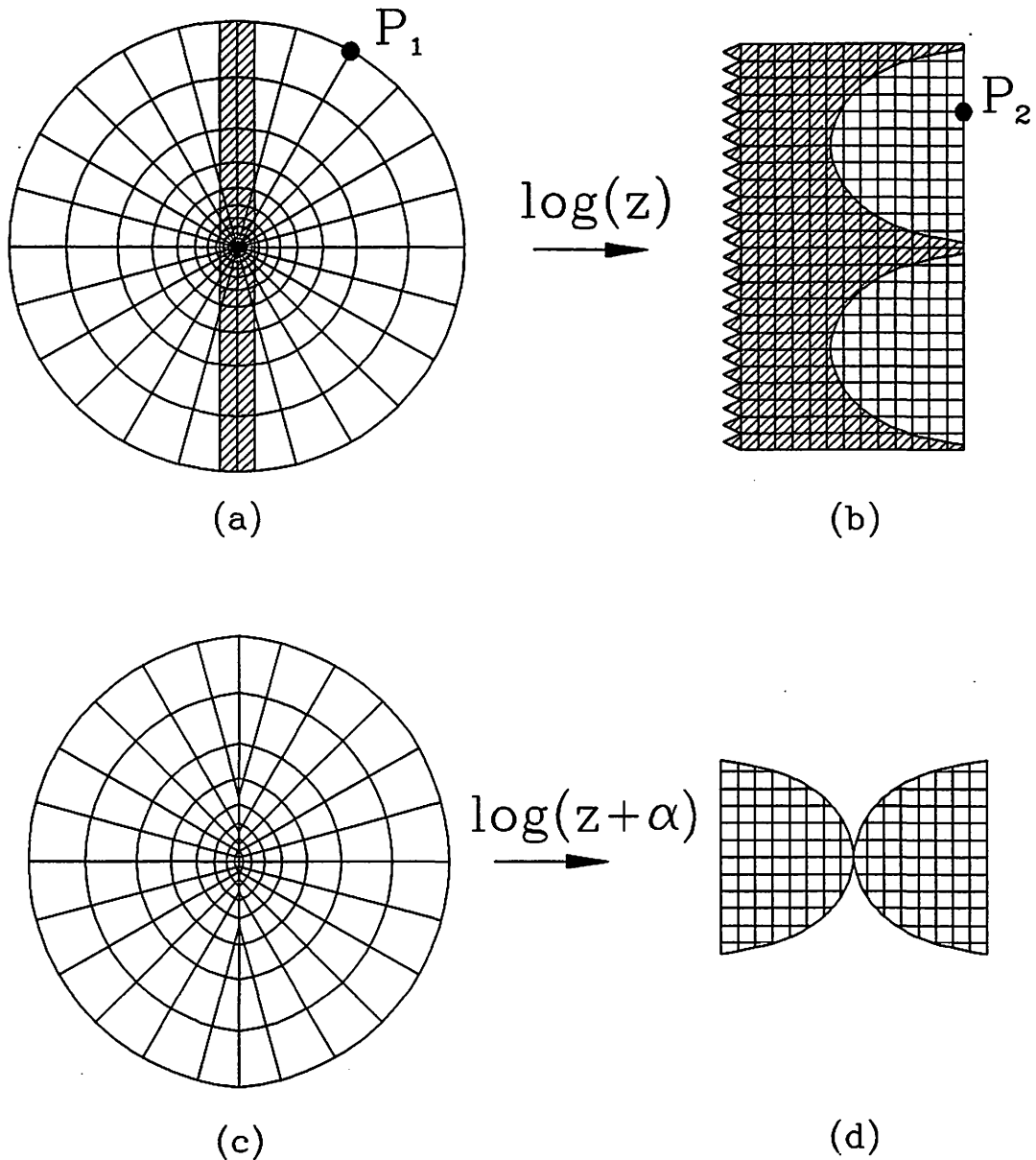


Figure 1.1: This shows the complex logmap, $\log(z)$ in a) and b) and its relative, the mapping, $\log(z + \alpha)$ in c) and d). It can be seen that the $\log(z + \alpha)$ map results from cutting the hatched region out of the $\log(z)$ map. Note that rings centered around the fovea in the domain get 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.

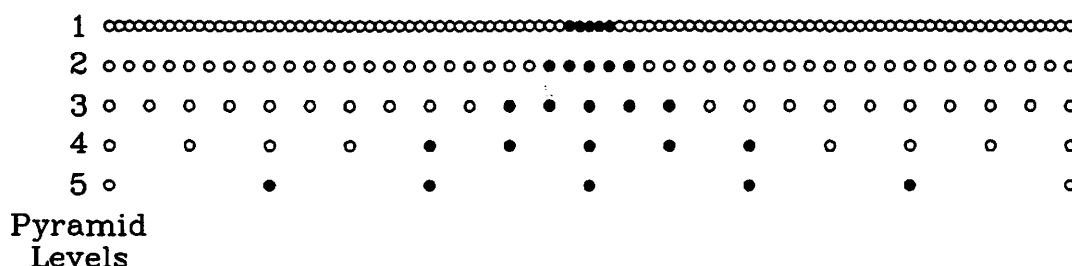


Figure 1.2: This shows the truncated pyramid structure used by Peter Burt. Each successive level represents a lower resolution image. The filled in circles represent pixels present in the pyramid. Notice that there are a constant number of pixels in the pyramid at each level where they cover a larger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

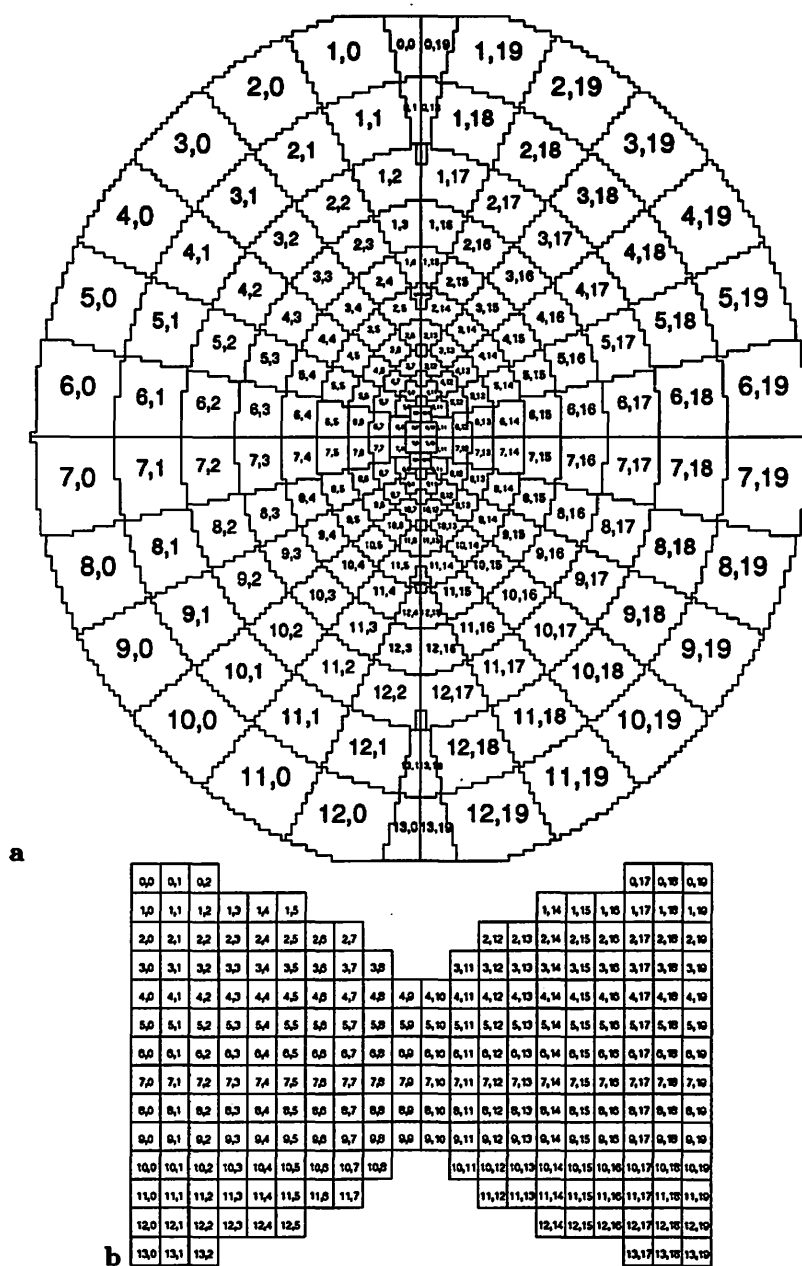


Figure 1.3: a) The lookup tables $S(i, j)$ and $R(i, j)$ combined into one figure. b) The forward map space. The correspondence between inverse and forward map pixels is shown.

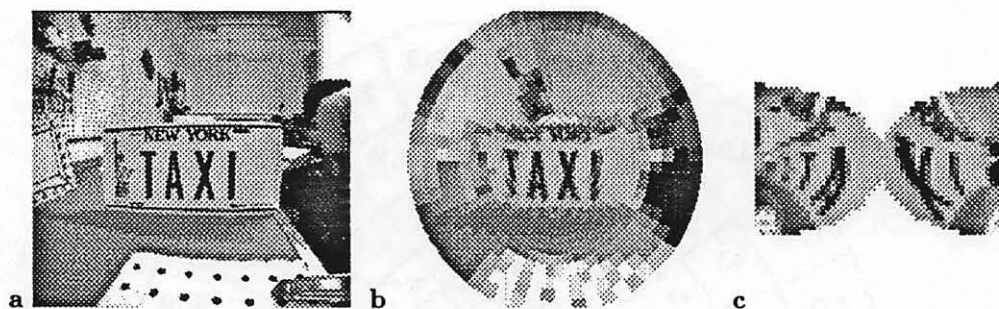


Figure 1.4: A sample of a) a TV image, $I(i, j)$, b) The inverse logmap image $L^{-1}(i, j)$, and c) The forward logmap image $L(u, v)$.

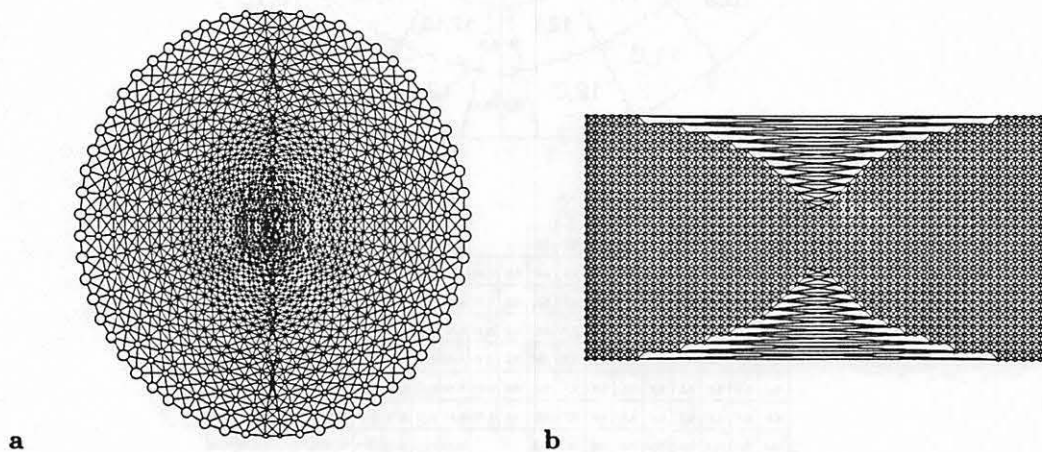


Figure 1.5: The connectivity graph for a $\log(z + \alpha)$ sensor geometry. Circles stand for graph vertices and lines stand for graph edges. The diameter of the circles in the inverse map is proportional to the area of the pixel. The graph explicitly represents the neighbor relations across the vertical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

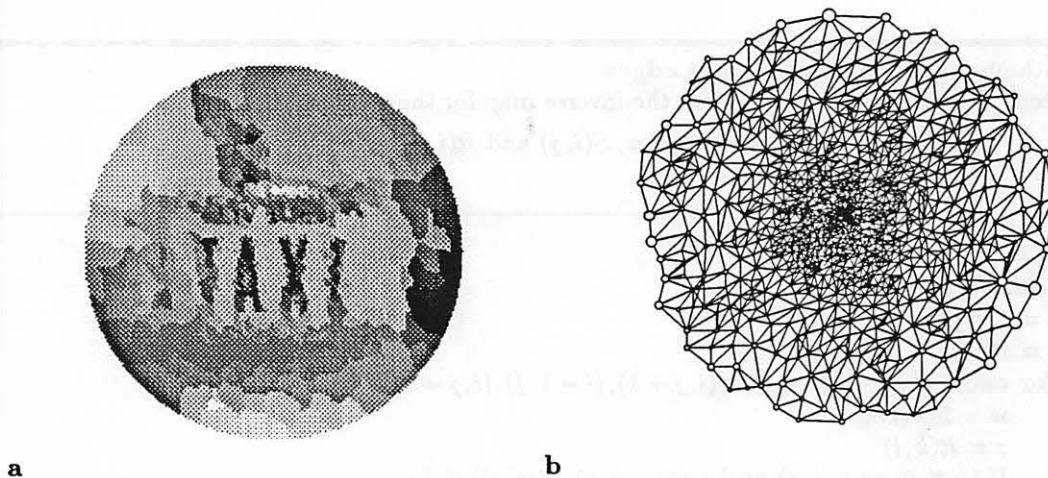


Figure 1.6: a) The image from a sensor having a random pixel geometry. b) The connectivity graph for this sensor.

$(u(p), v(p))$ 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

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

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 $L_{t+1}(p) = L(p)$, otherwise,

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

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 ($1K \times 1K$ or $2K \times 2K$ 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 $8 \times 8 \times 10$ 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 $7.5 \times 7.5 \times 12$ cm and can move at rotational velocities up to $100^\circ/\text{sec}$.

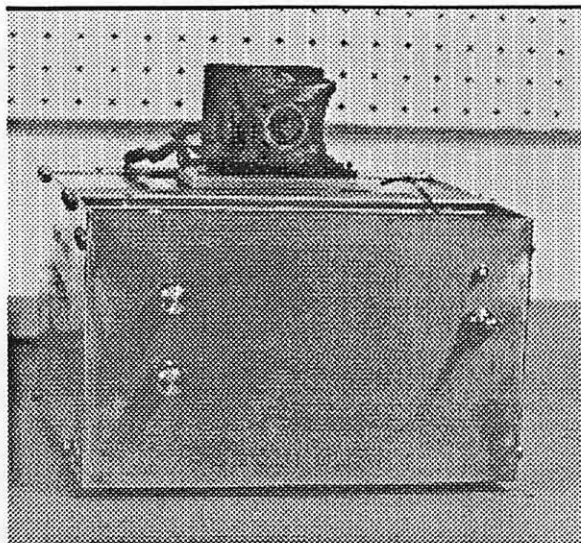


Figure 1.7: **Cortex-I**. Its dimensions are $14 \times 22 \times 22$ cm.

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

CHAPTER 1. INTRODUCTION

17

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

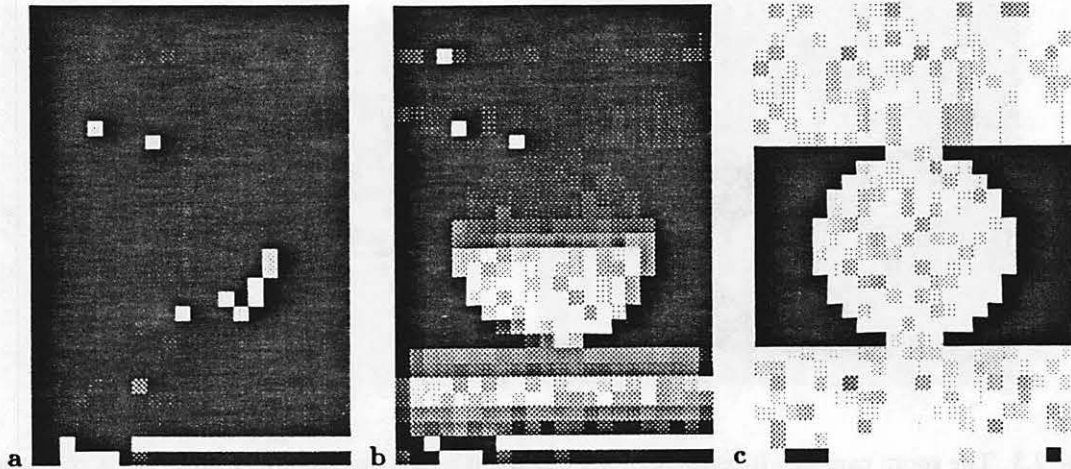


Figure 2.1: Images from our custom logmap sensor chip. They are the logmap images from the chip, and have not been mapped back to scene coordinates. They are scenes of a) an all black scene, b) a horizontal edge with the top half black and the bottom half white, and c) an all white scene. The circle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

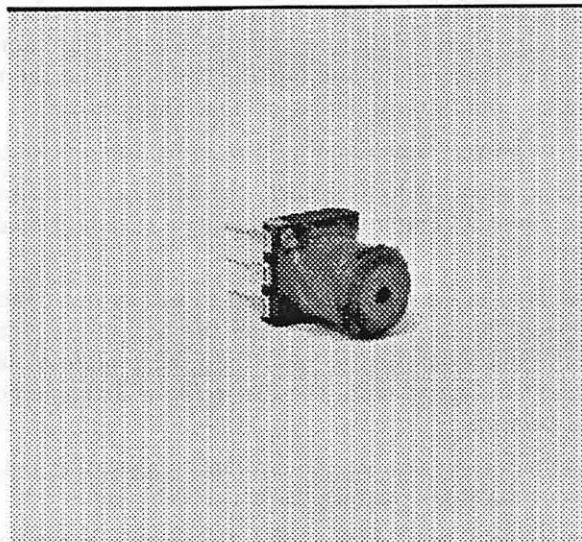


Figure 2.2: The fixed field of view camera. It consists of a four element lens assembly attached directly to a CCD sensor.

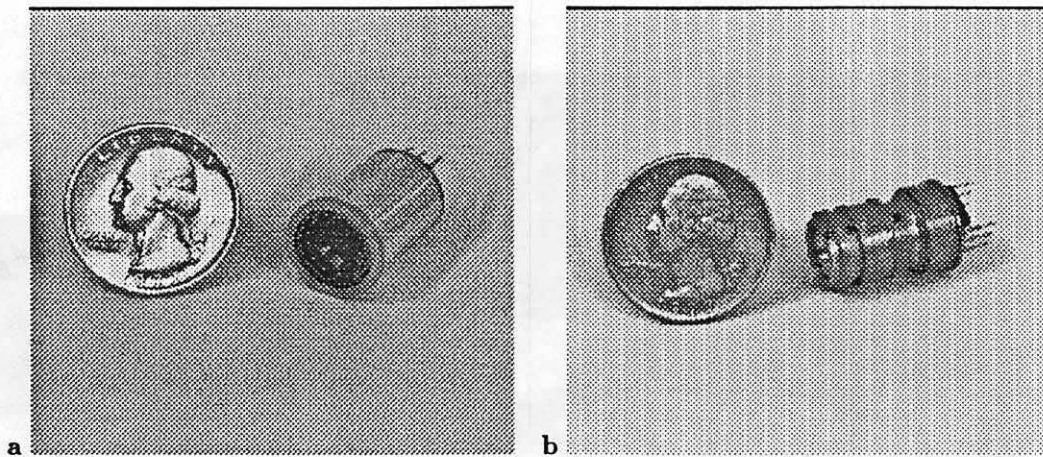


Figure 2.3: The zoom camera. It consists of a six element zoom lens assembly including a coil to actuate the zoom lens. It is attached directly to the CCD sensor. It is shown a) with and b) without the light baffle.

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.

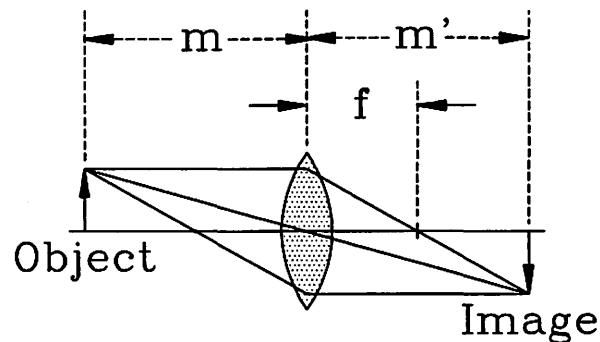


Figure 2.4: Illustration of imaging an object with a simple biconvex lens.

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

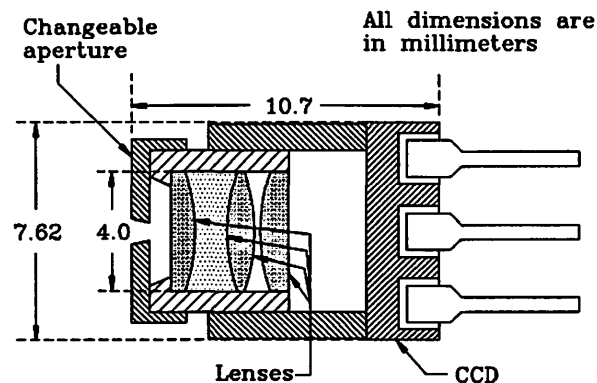


Figure 2.5: Mechanical drawing of the fixed field of view camera. The 4 element lens is a 4 mm orthoscopic lens that is corrected for chromatic aberrations.

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

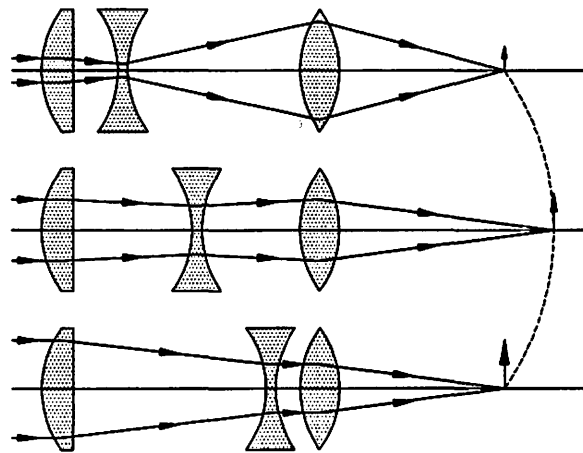


Figure 2.6: Illustration of zoom lens. The image plane moves as the magnification of the lens changes. (Adapted from [51, P. 275])

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

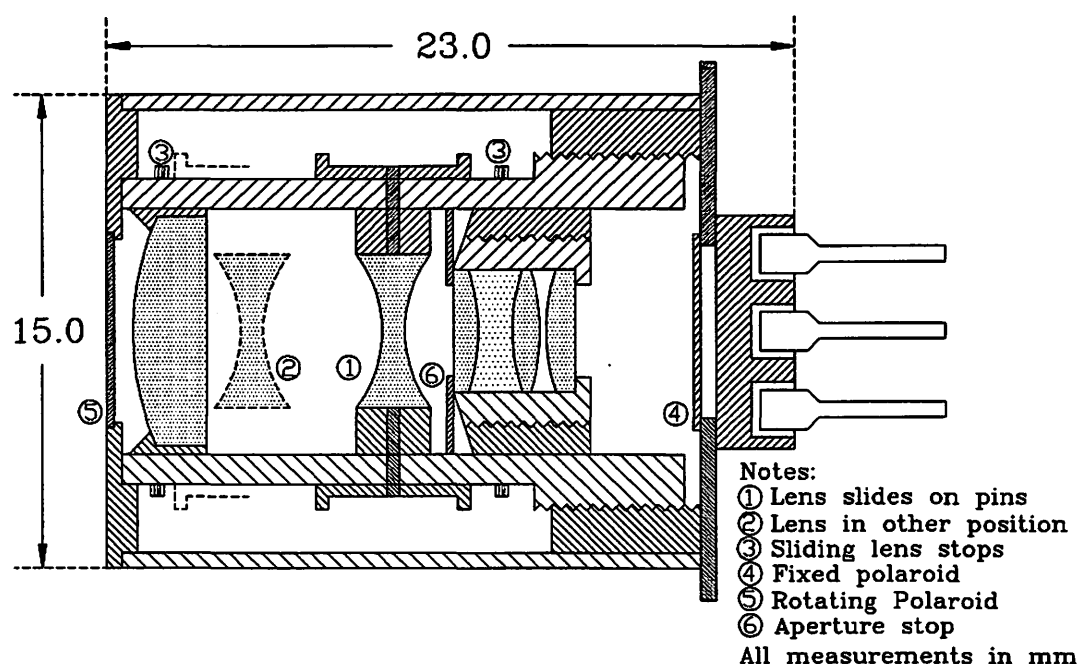


Figure 2.7: Mechanical drawing of the zoom camera.

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,

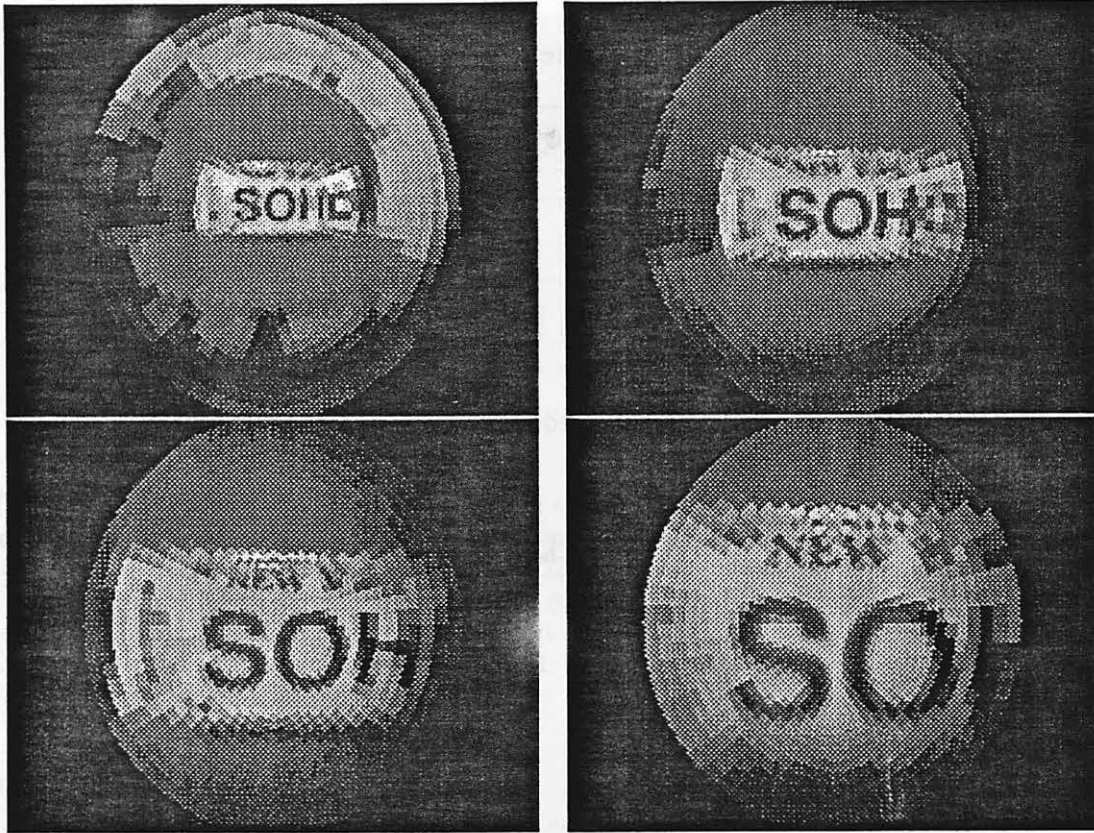


Figure 2.8: The same scene imaged with the zoom lens in four different positions (manually actuated).

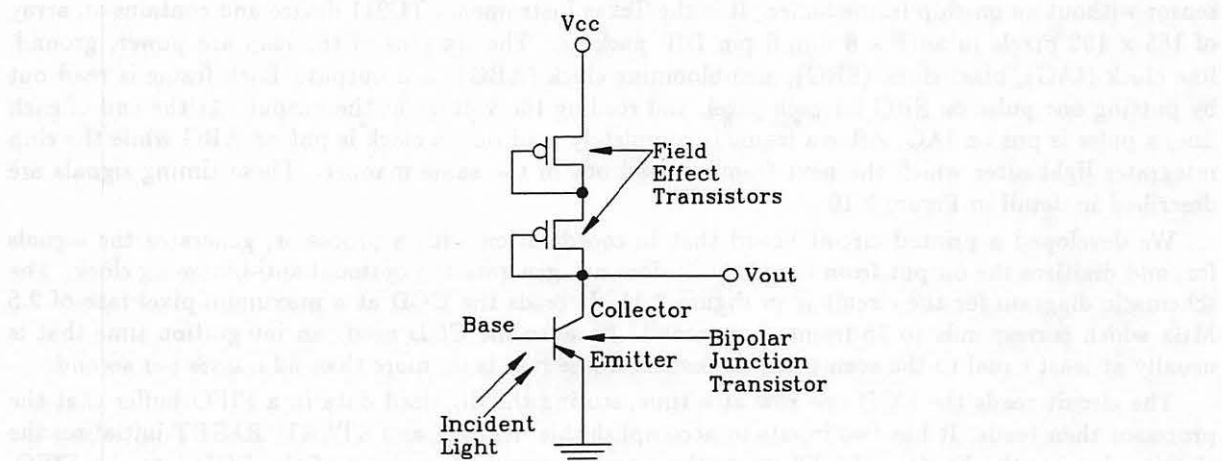


Figure 2.9: A vertical bipolar junction transistor hooked up with two field effect transistors to produce a photo cell with a logarithmic output.

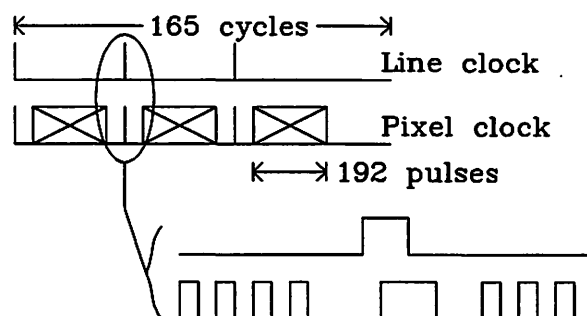


Figure 2.10: Timing signals used to control the CCD image sensor.

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

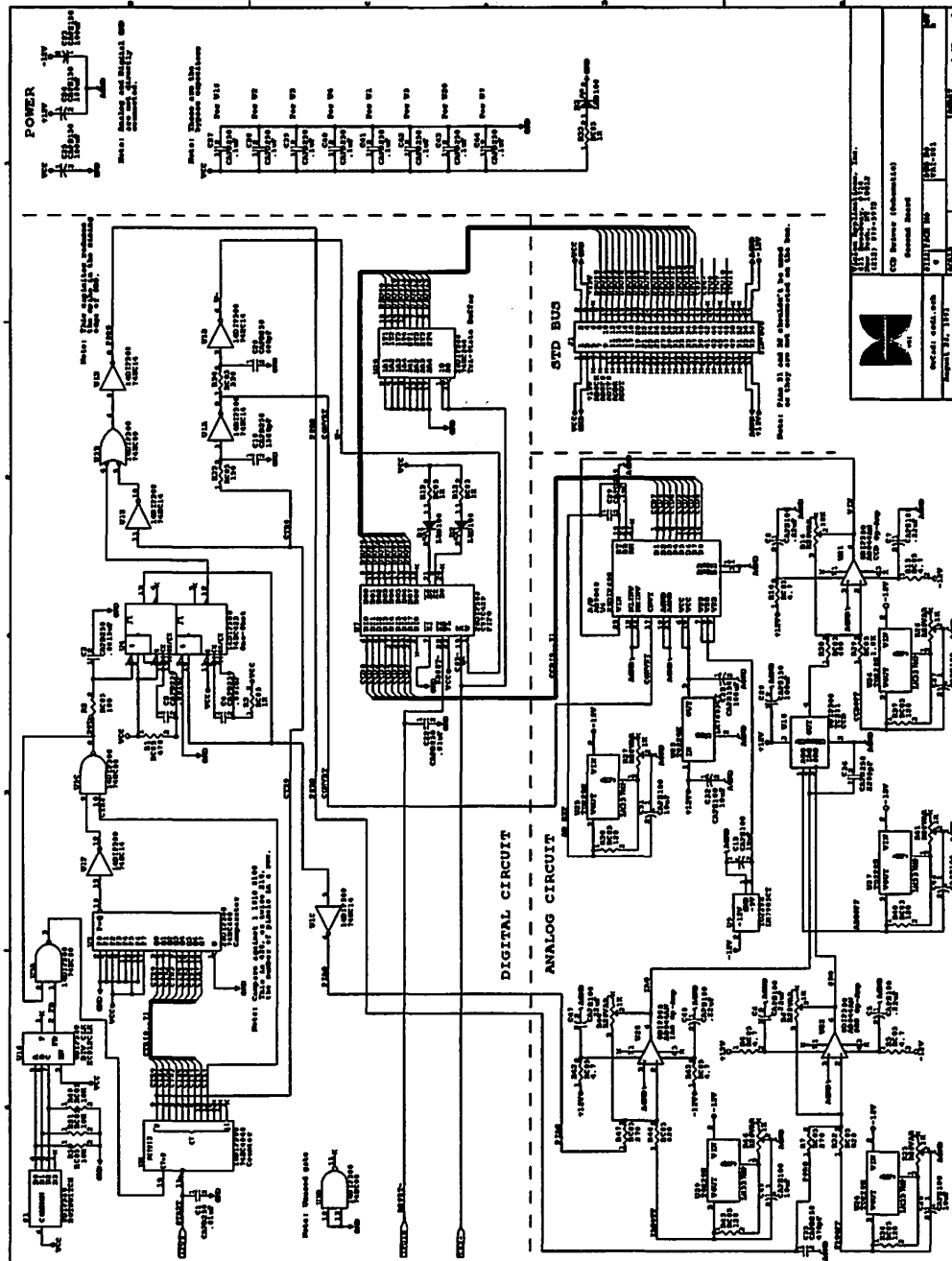


Figure 2.11: Schematic diagram for CCD driver board.

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

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

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

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

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

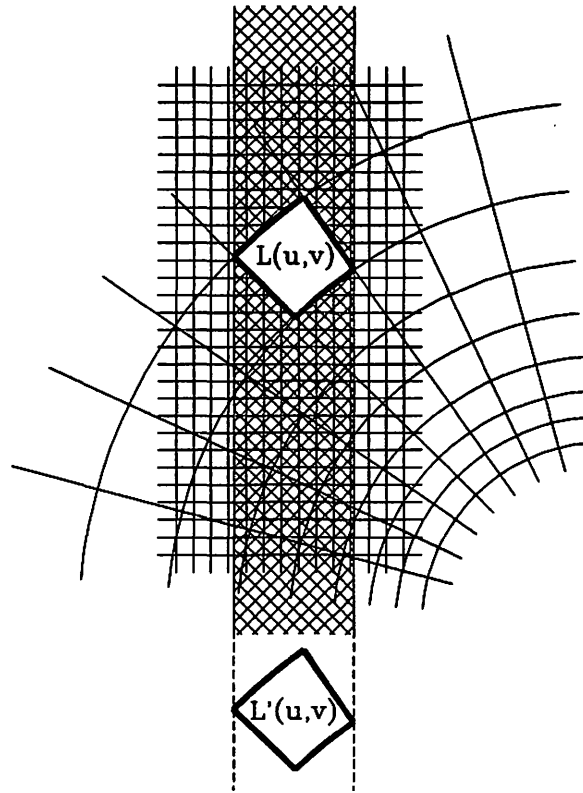


Figure 2.12: Elimination of vertical smearing. Actual logmap pixel $L(u, v)$ is calculated from read out logmap pixel $L'(u, v)$ and a weighted sum of all other logmap pixels sharing TV pixels in the same columns.

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

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.

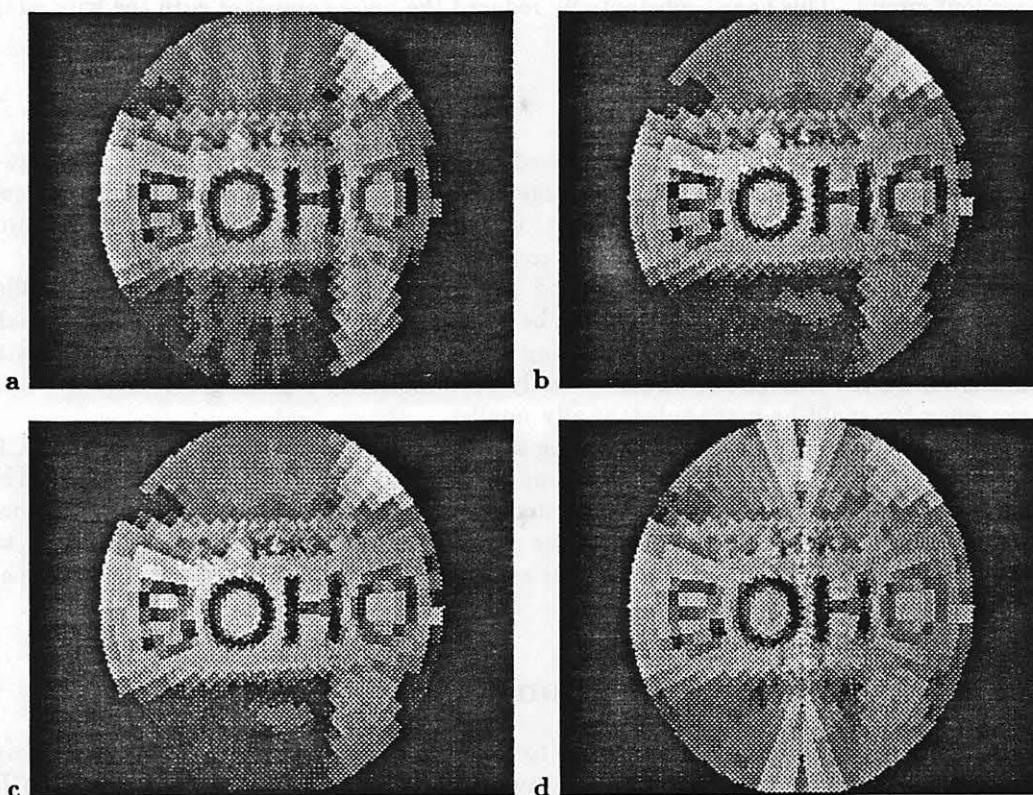


Figure 2.13: a) Image from miniature camera with vertical smearing; b) desmeared image; c) desmeared image using only $|A^{-1}| > 0.001$; d) using only $|A^{-1}| > 0.01$.

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

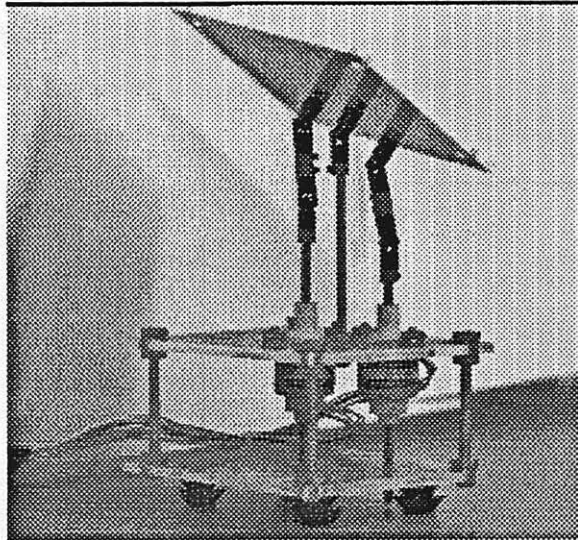


Figure 3.1: The Platform Pantilt. This stepper motor based actuator is capable of positioning a load of 80 grams, for example a small video camera. Dimensions are $7.5 \times 7.5 \times 12$ cm.

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

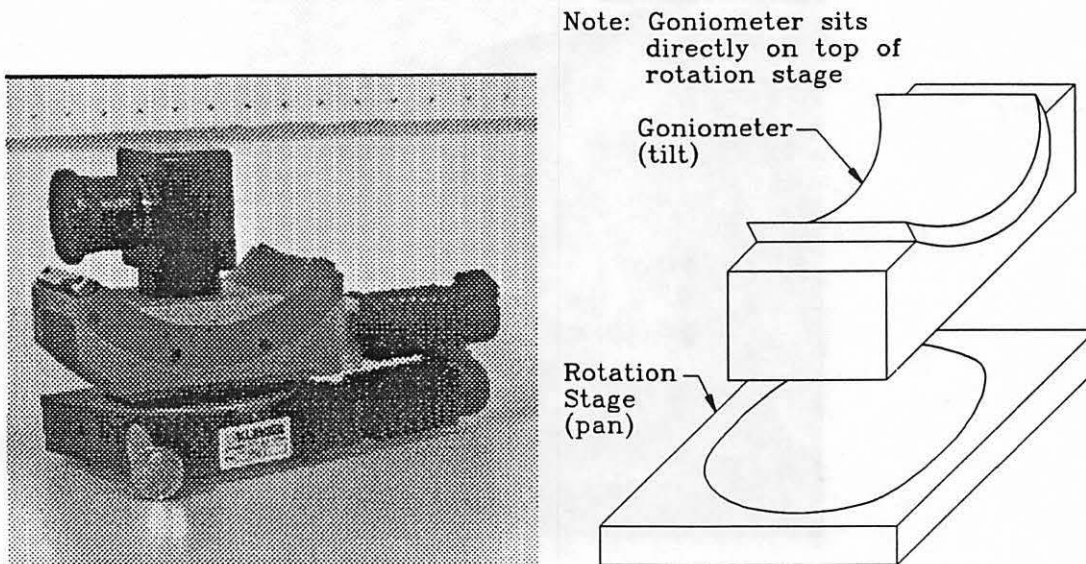


Figure 3.2: Photo and illustration of a traditional motor-on-motor design. Actuator from Klinger Scientific. Dimensions are $10 \times 11 \times 22$ cm (not including the camera or the rack-mounted controller/driver).

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

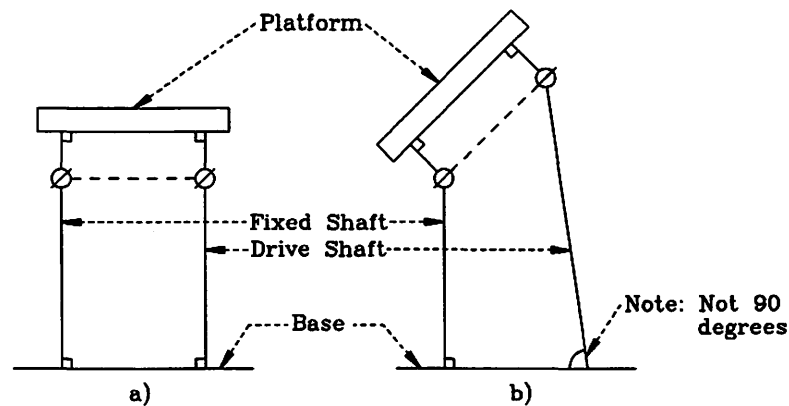


Figure 3.3: A non-working two-dimensional mechanism in two different positions with one pin joint on each shaft. Note that in b), the lower right angle is not fixed at 90° as it should be.

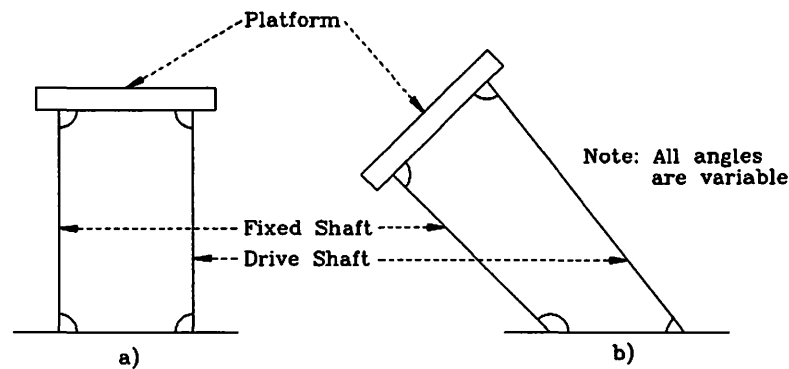


Figure 3.4: A two dimensional mechanism based on the Stewart Platform. Here, the drive shafts must be mounted on spherical joints.

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

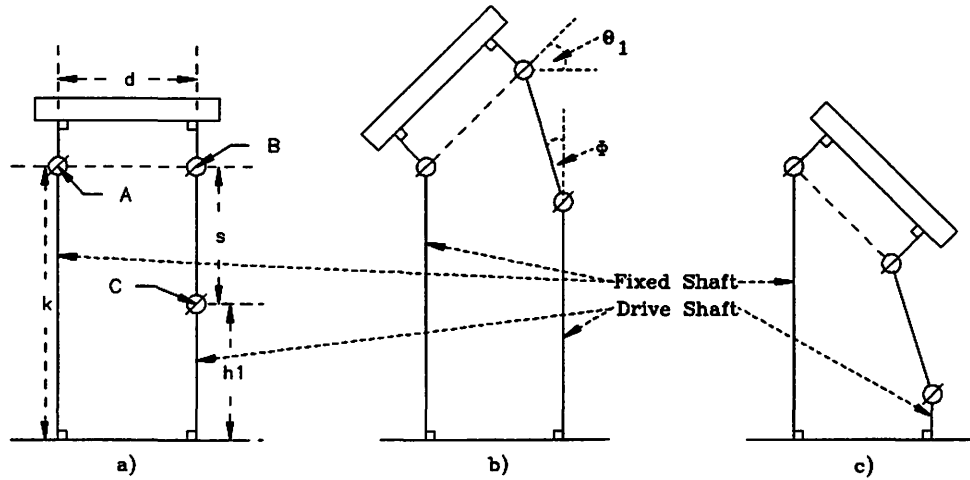


Figure 3.5: A two dimensional version of the mechanism we use shown in three positions. All the connections between the shafts and the platform and base are rigid. The dashed line between points A and B represents a fixed distance and does not represent a physical connection.

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

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

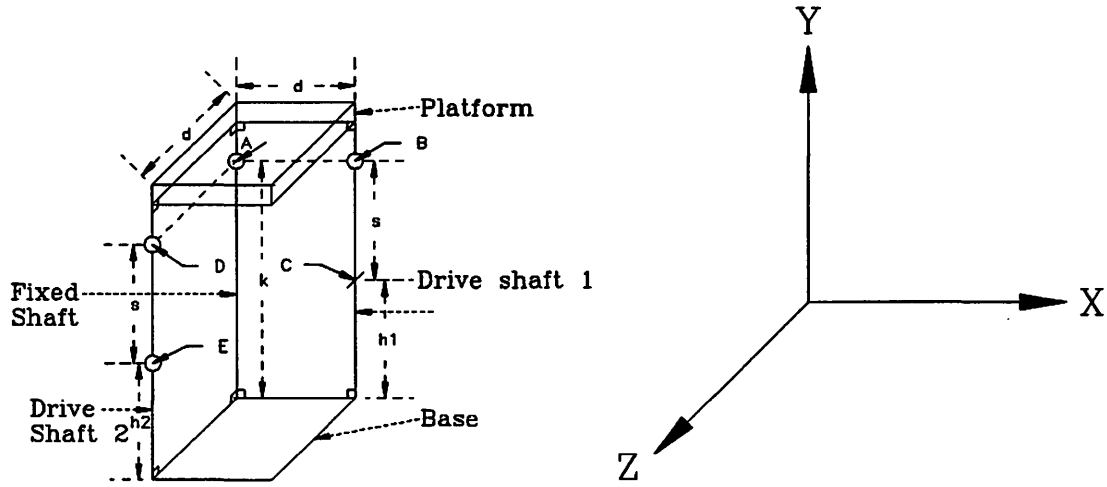


Figure 3.6: left) The three dimensional mechanism we use. Note that circles at points A, B, D and E denote universal joints with two degrees of freedom while the line at point C denotes a pin joint with only one degree of freedom. right) The coordinate system used in this section.

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

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

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

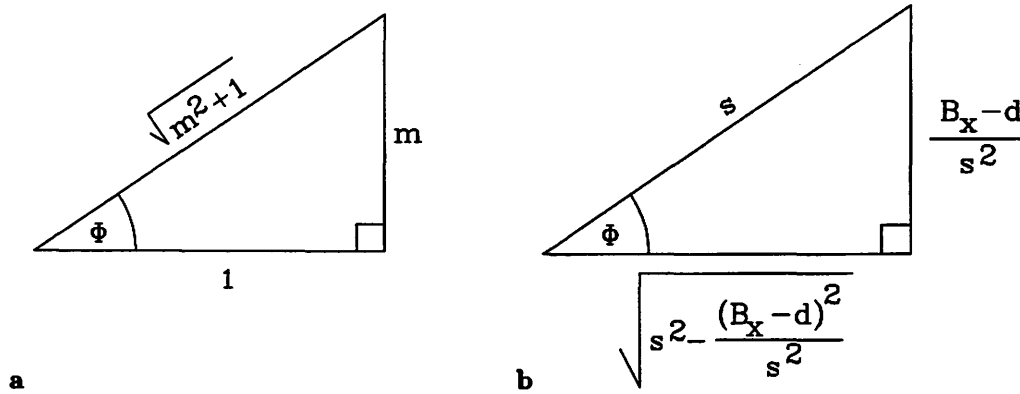


Figure 3.7: a) Triangle used to solve Equation 25. b) Triangle used to solve Equation 26.

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.

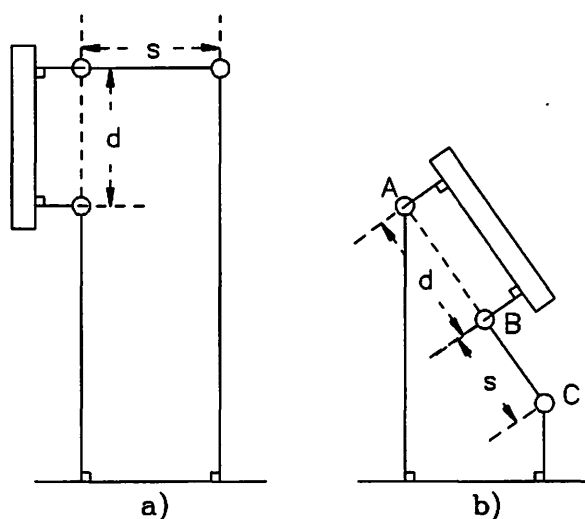


Figure 3.8: The limits of motion in two dimensions.

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.

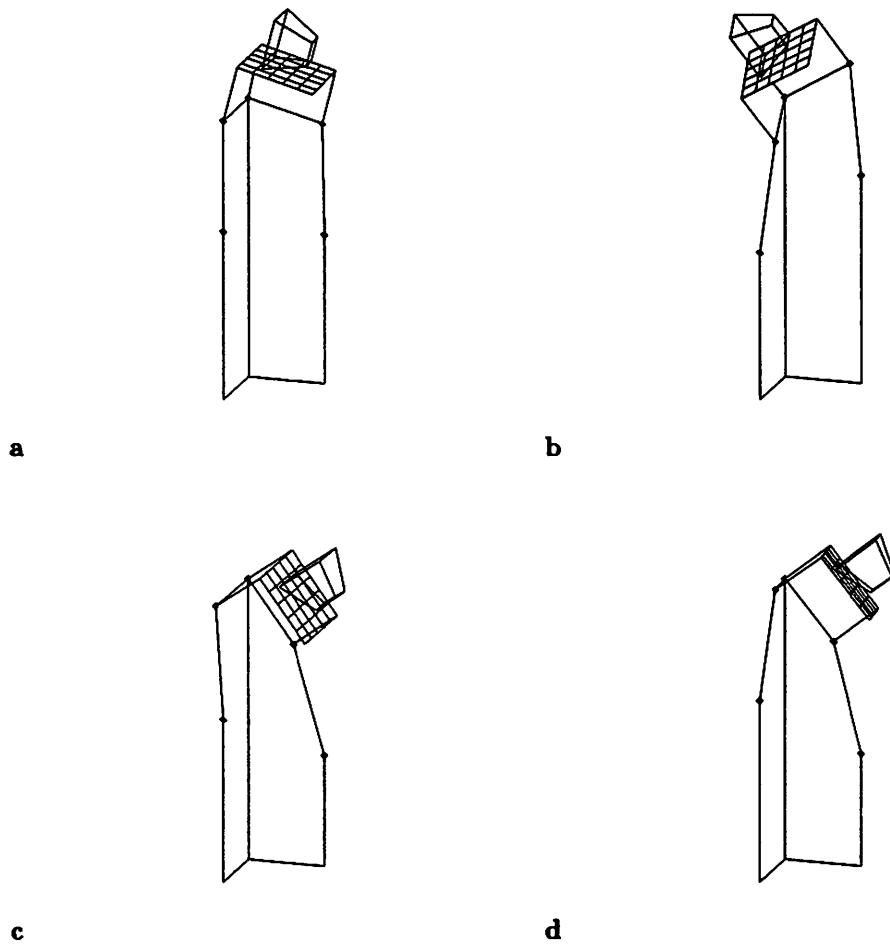


Figure 3.9: Simulation of the forward kinematics equations of this section. The Platform Pantilt is shown in four different positions.

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.

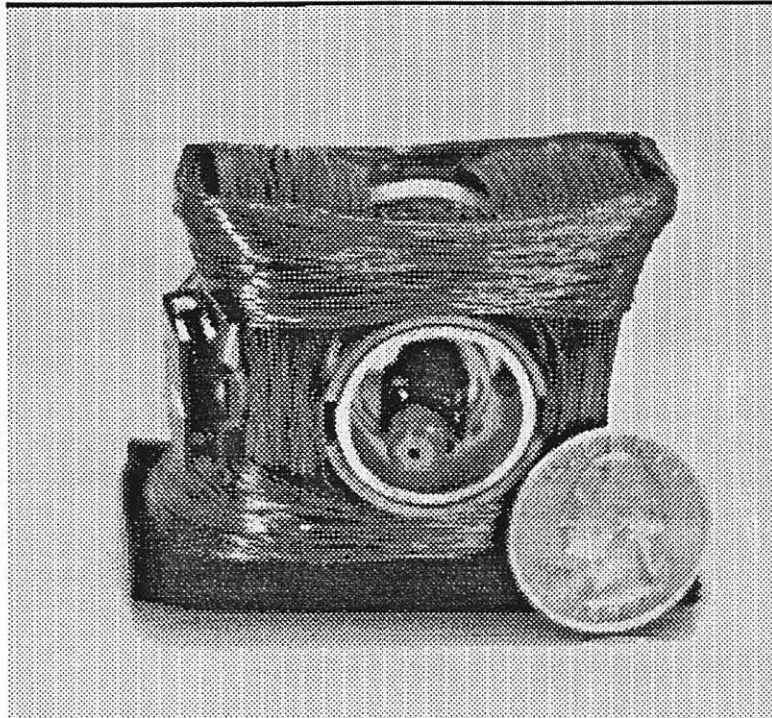


Figure 4.1: The Spherical Pointing Motor. At the center is a miniature camera consisting of a single CCD sensor chip and a lens assembly that fits on the rotor of this motor.

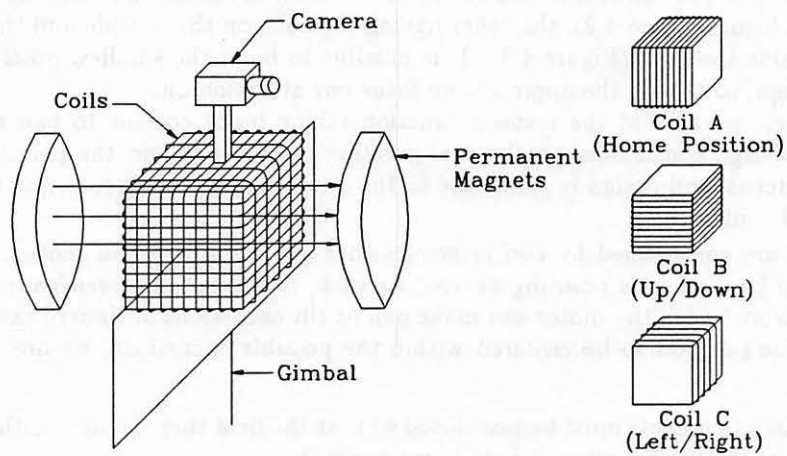


Figure 4.2: **left)** Illustration of the internal spherical pointing motor shown in its home position. **right)** Labels for the three coils.

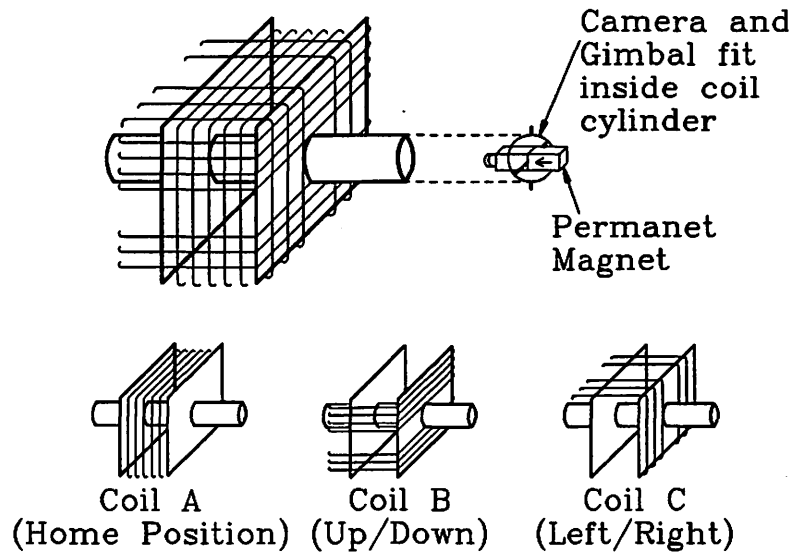


Figure 4.3: top) Illustration of the external spherical pointing motor shown in its home position. bottom) Labels for the three coils.

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

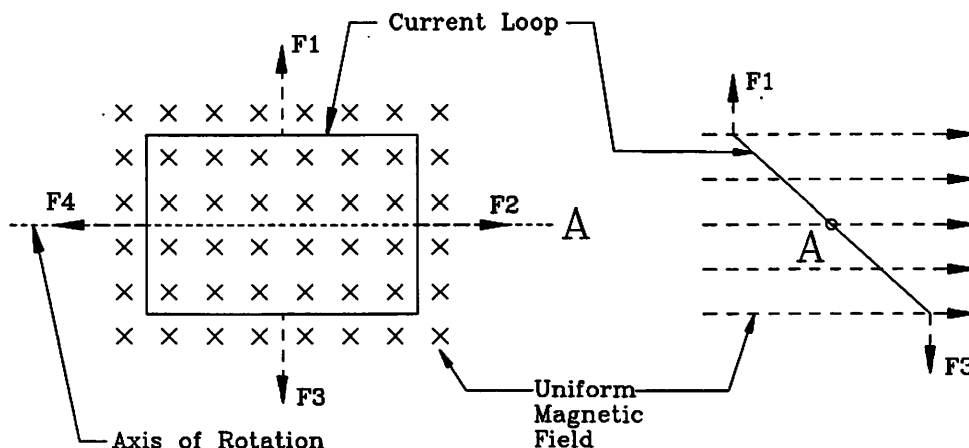


Figure 4.4: Torque on a current loop. **left)** Current loop rotates about horizontal axis with clockwise current. Uniform magnetic field normal to the page. Forces F_2 and F_4 are aligned with the axis and produce no net force. Forces F_1 and F_3 rotate the coil so that the coil encloses as many magnetic 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.

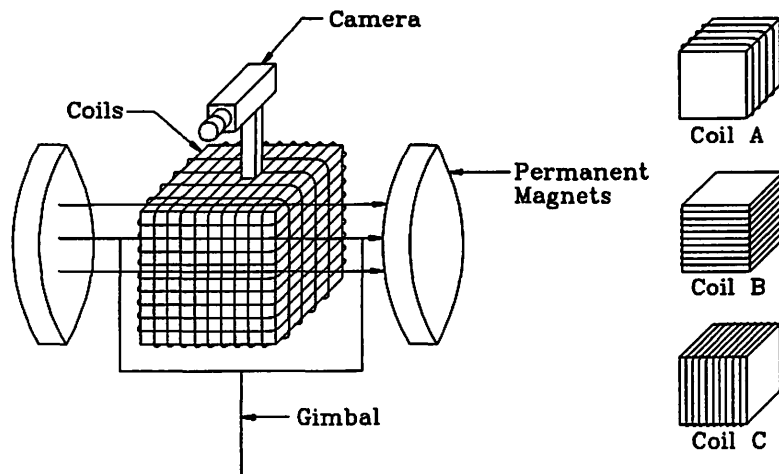


Figure 4.5: left) The internal coil motor panned 90° so that tilting is not possible. right) The coil labels for the motor in this position. Notice how coils *A* and *C* have swapped orientations.

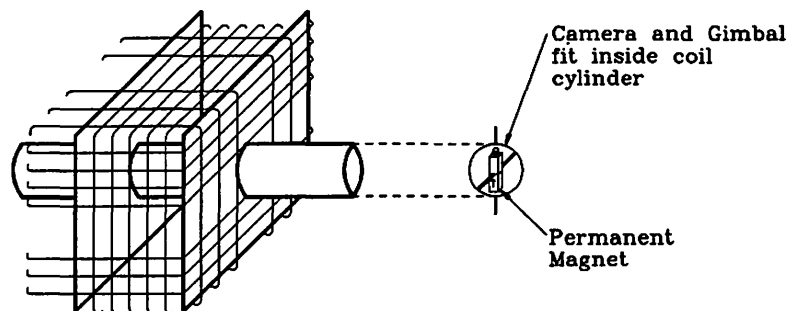


Figure 4.6: The external coil motor tilted 90° so that panning is not possible.

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.

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

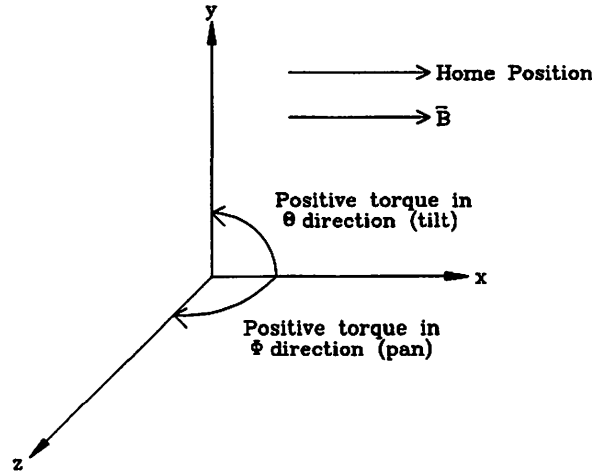


Figure 4.7: The coordinate system used for calculations. The permanent magnetic field and the home position both lie on the direction of the positive x-axis.

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

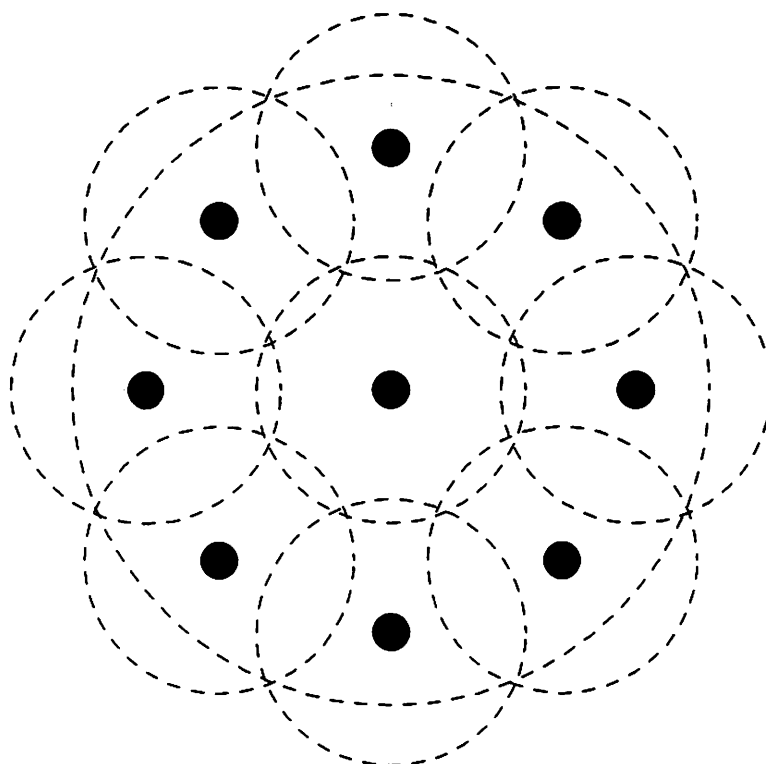


Figure 4.8: The solid dots comprise the scene used to calibrate the SPM. The nine smaller dashed circles are not part of the calibration pattern, but represent the field of view of the camera when centered on each dot. The single large dashed circle is also not in the scene, but represents the workspace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

Do

Move 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 $p_2 \in L$

$\text{visited}[p_2] \leftarrow \text{FALSE}$

End For

$\text{visited}[p_1] \leftarrow \text{TRUE}$

$\text{front} \leftarrow 0$

$\text{back} \leftarrow 0$

While $((L(p_1) \geq \text{threshold}) \text{ AND } (\text{front} < \text{NPIXELS}) \text{ AND } (a(p_1) > 0))$ **Do**

For each neighbor $p_2 \in \mathcal{N}(p_1)$

If $(\text{visited}[p_2] = \text{FALSE})$ **Then**

$\text{visited}[p_2] \leftarrow \text{TRUE}$

$\text{recurse}[\text{back}] \leftarrow p_2$

$\text{back} \leftarrow \text{back} + 1$

End If

End For

$p_1 \leftarrow \text{recurse}[\text{front}]$

$\text{front} \leftarrow \text{front} + 1$

End While

$\text{return}(\text{centroid}(p_1))$

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( $\mathcal{P}_b$ )

```

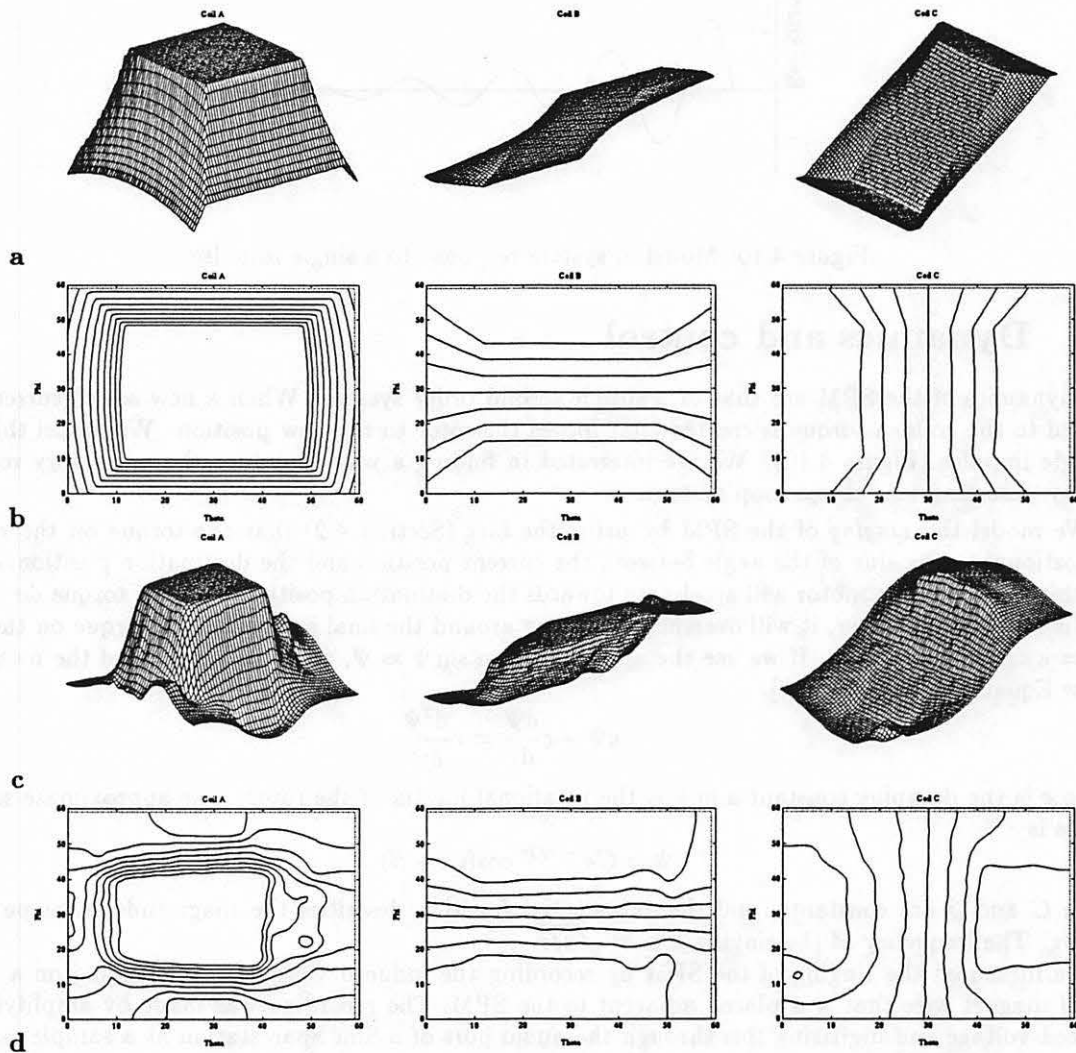


Figure 4.9: a) Theoretical calculated currents vs. position for each of the three coils shown as a mesh plot. b) The same as a) shown as a contour plot. c) Calibrated currents vs. position for each of the three coils shown as a mesh plot. d) The same as c) shown as a contour plot. The data shown here is for the $\pm 30^\circ$ workspace in (Θ, Φ) .

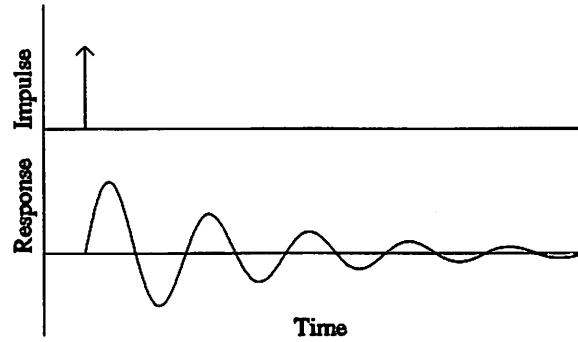


Figure 4.10: Model of system response to a single impulse.

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

$$\Psi = Ce^{-ct/2r} \cos(\omega t + \beta) \quad (46)$$

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).

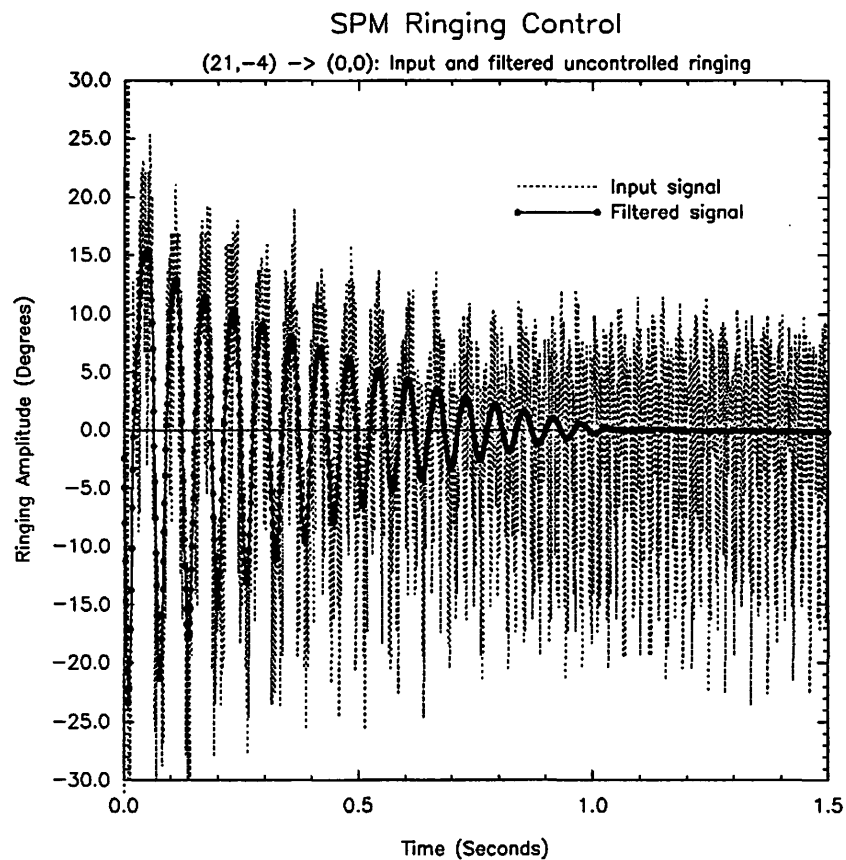


Figure 4.11: Uncontrolled ringing measurement taken by induced voltage on a separate pickup coil. The input signal (combination of ringing and PWM) is shown along with the filtered signal that isolates the ringing.

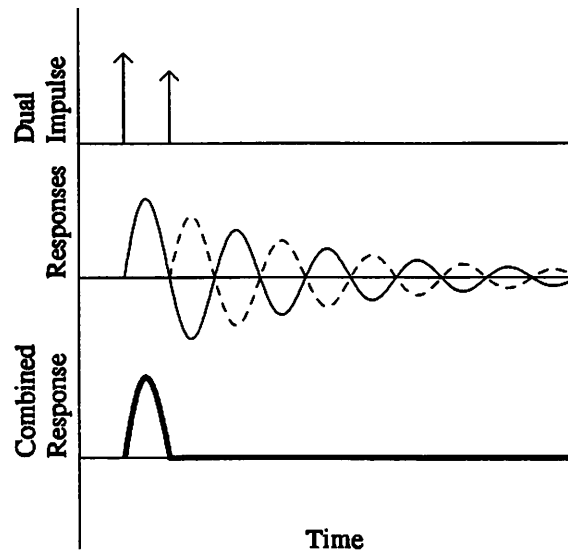


Figure 4.12: Ideal model of system response to a dual impulse.

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.

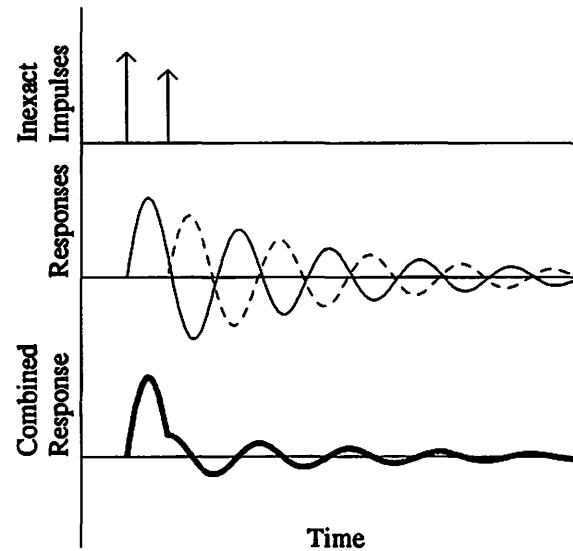


Figure 4.13: Model of system response with inexact timing to a dual impulse.

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.

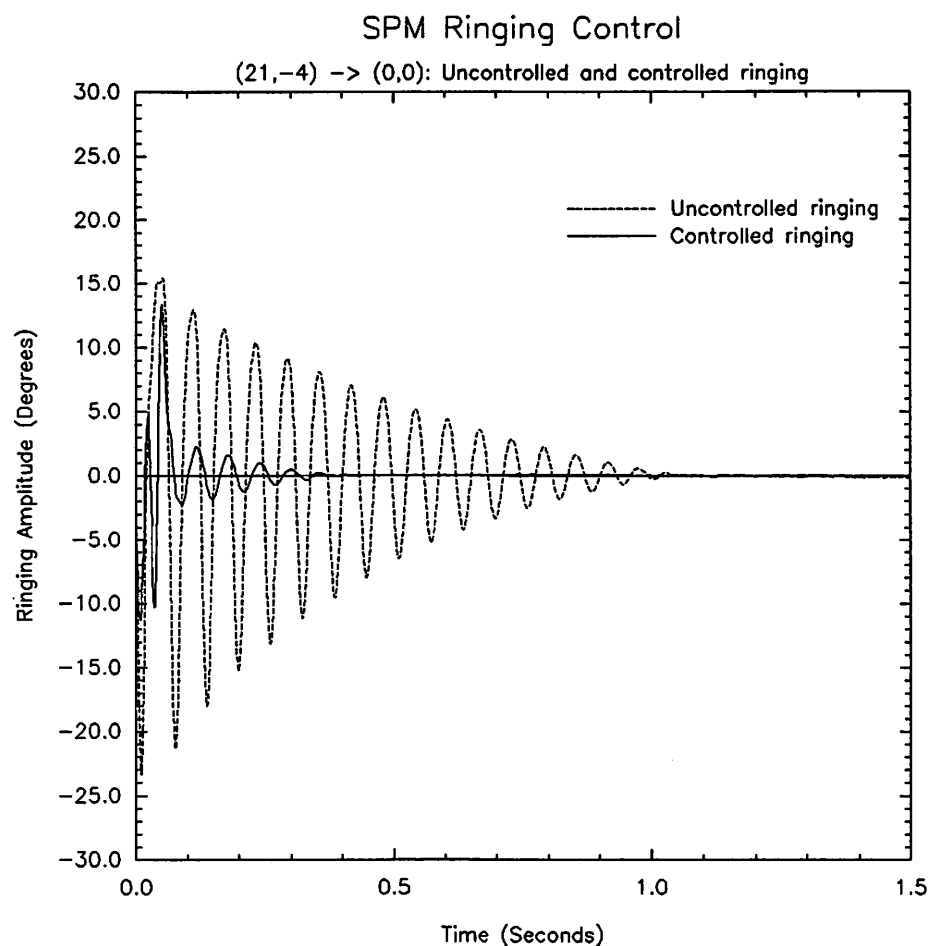


Figure 4.14: Measurement of two-step open-loop control strategy to reduce SPM ringing. Results are shown for the same movement of 21.4° with and without control. The motor currents are changed to hold the motor at the final position when the motor is expected to have zero velocity. Because the midpoint 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.

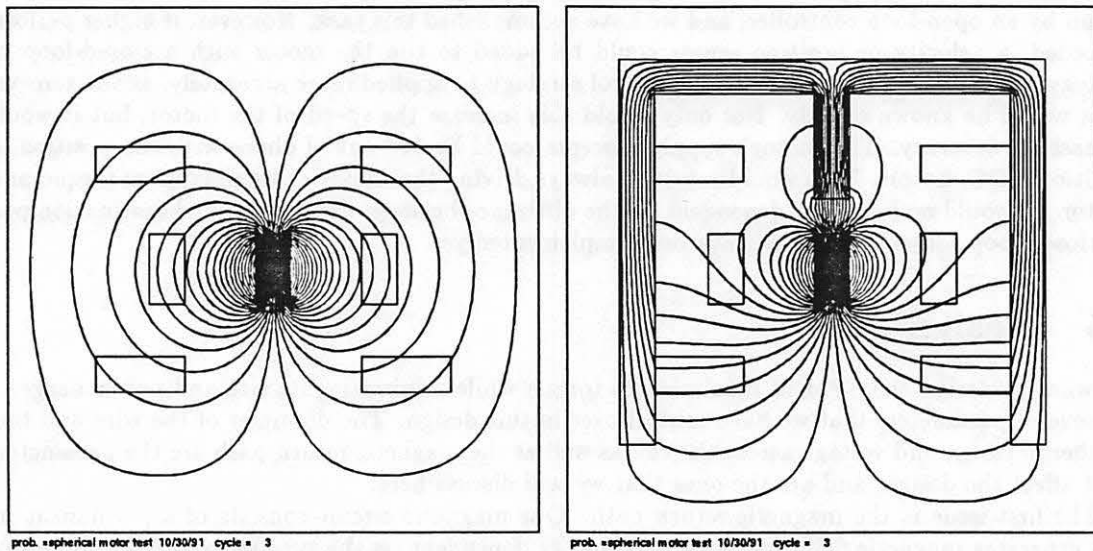


Figure 4.15: Simulation of magnetic circuit for the SPM (left) without and (right) with an iron magnetic return path. The permanent magnet is in the center. A camera mounted on the magnet would be looking towards the bottom of the page. The rectangles represent the coils of the motor. It can be seen 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.

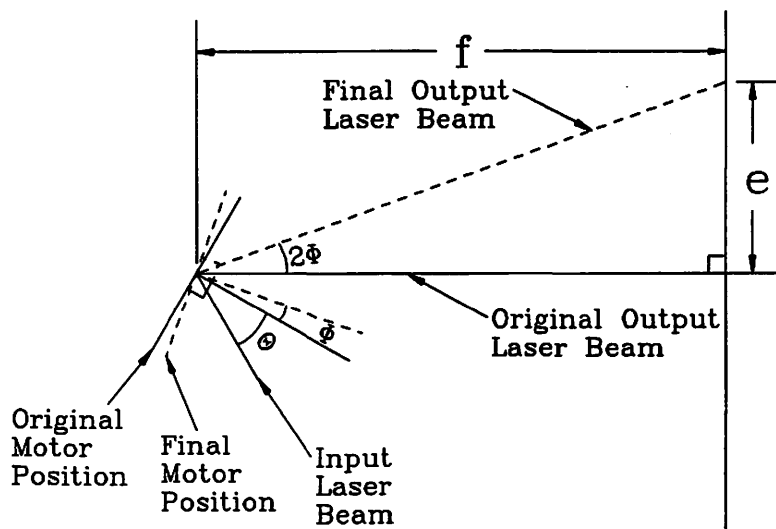


Figure 4.16: Illustration of our experimental setup for measuring motor accuracy. The laser beam is reflected off the rotor onto a flat surface. The motor is then moved to a new position and back to the original position. The new laser beam position is compared with the original position to see how accurately 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.

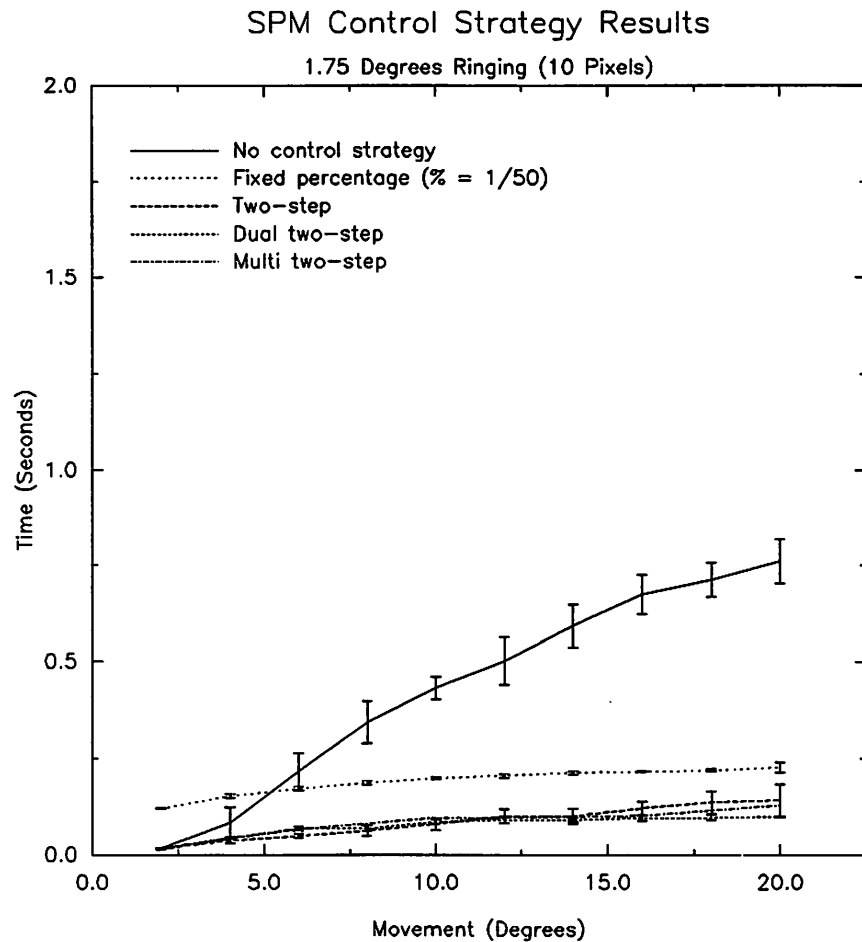


Figure 4.17: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 1.75° (10 pixels in our camera). Control strategies are discussed in the text.

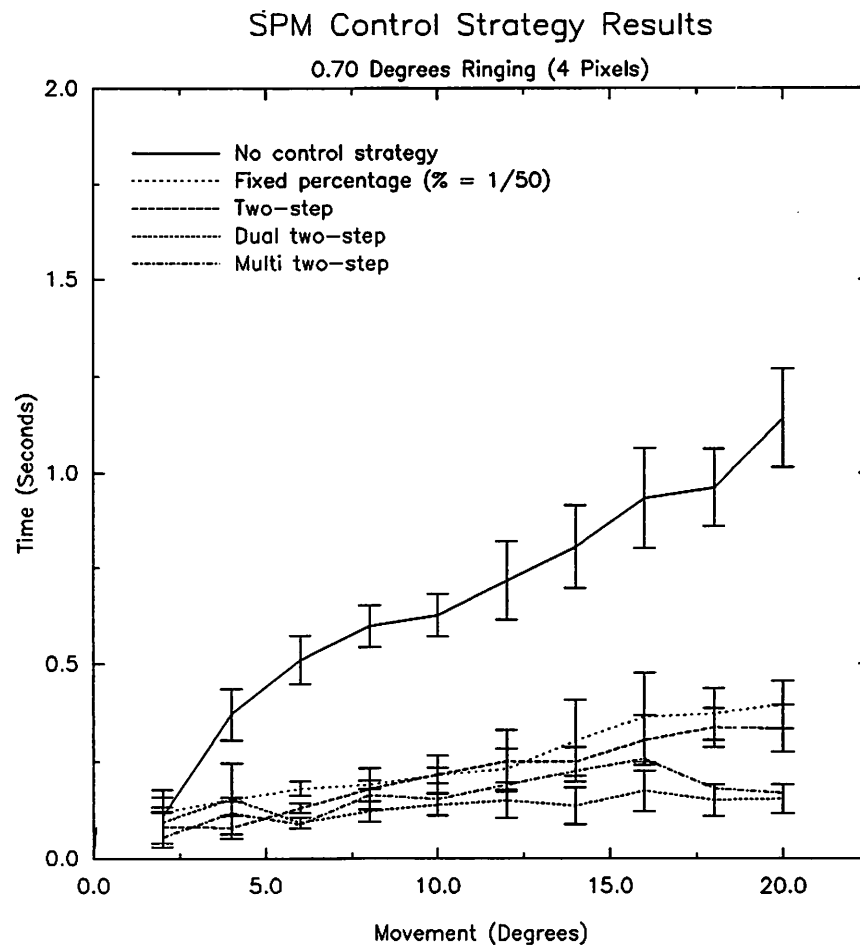


Figure 4.18: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 0.70° (4 pixels in our camera).

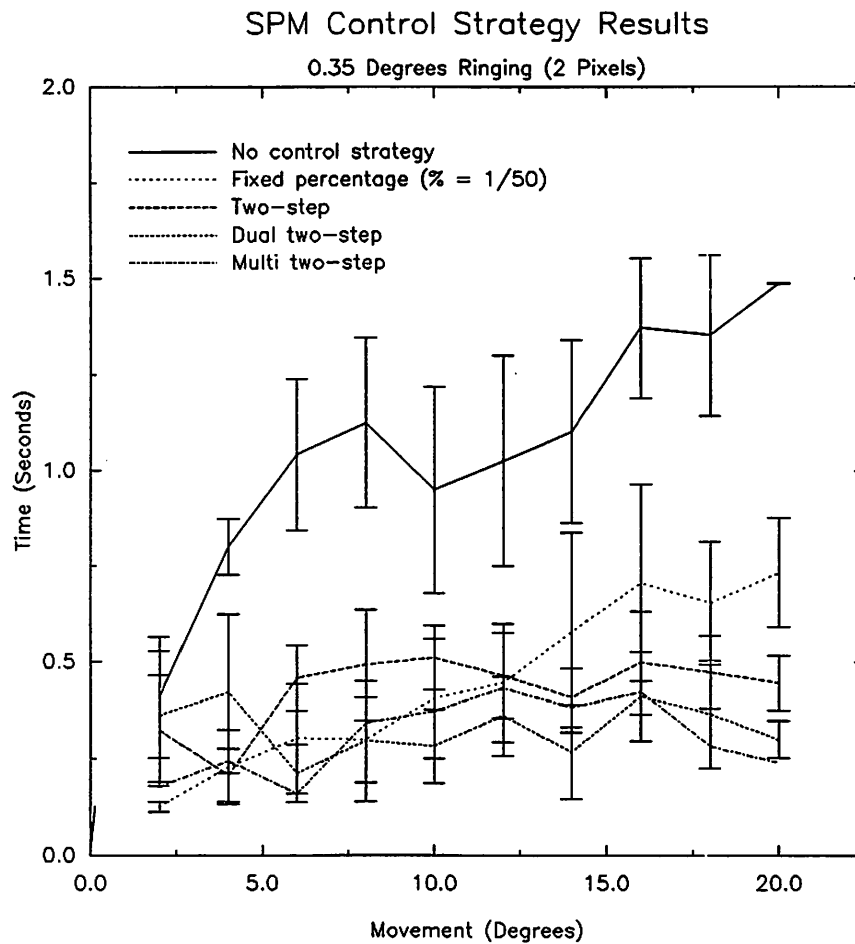


Figure 4.19: Results of different open-loop control strategies to reduce the ringing of the SPM. Times reported are those after which amplitude of the ringing has been reduced to 0.35° (2 pixels in our camera).

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.

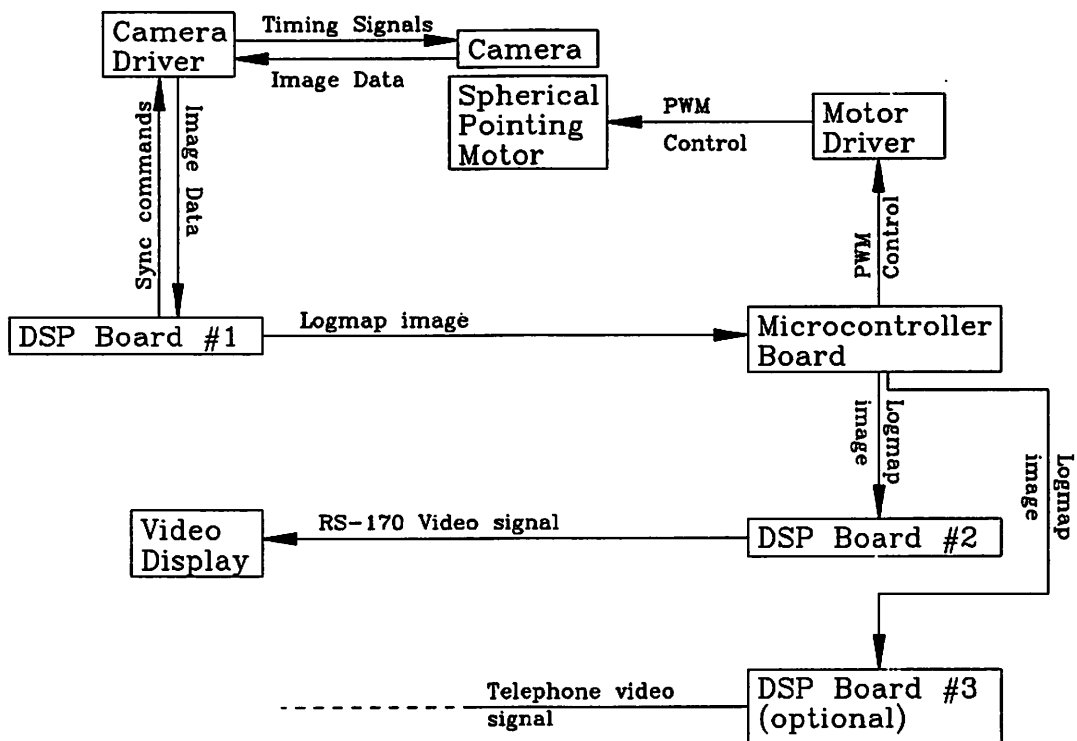


Figure 5.1: System architecture overview.

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.

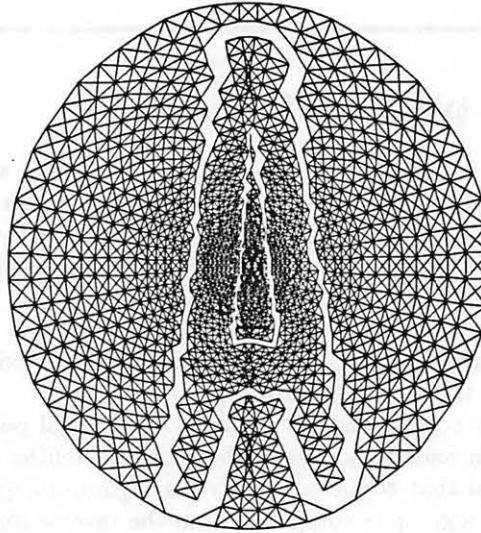


Figure 5.2: The connectivity graph reduces the image connected components problem to its graph theory counterpart. Here we show the foreground-background segmentation of an image containing the letter "A".

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

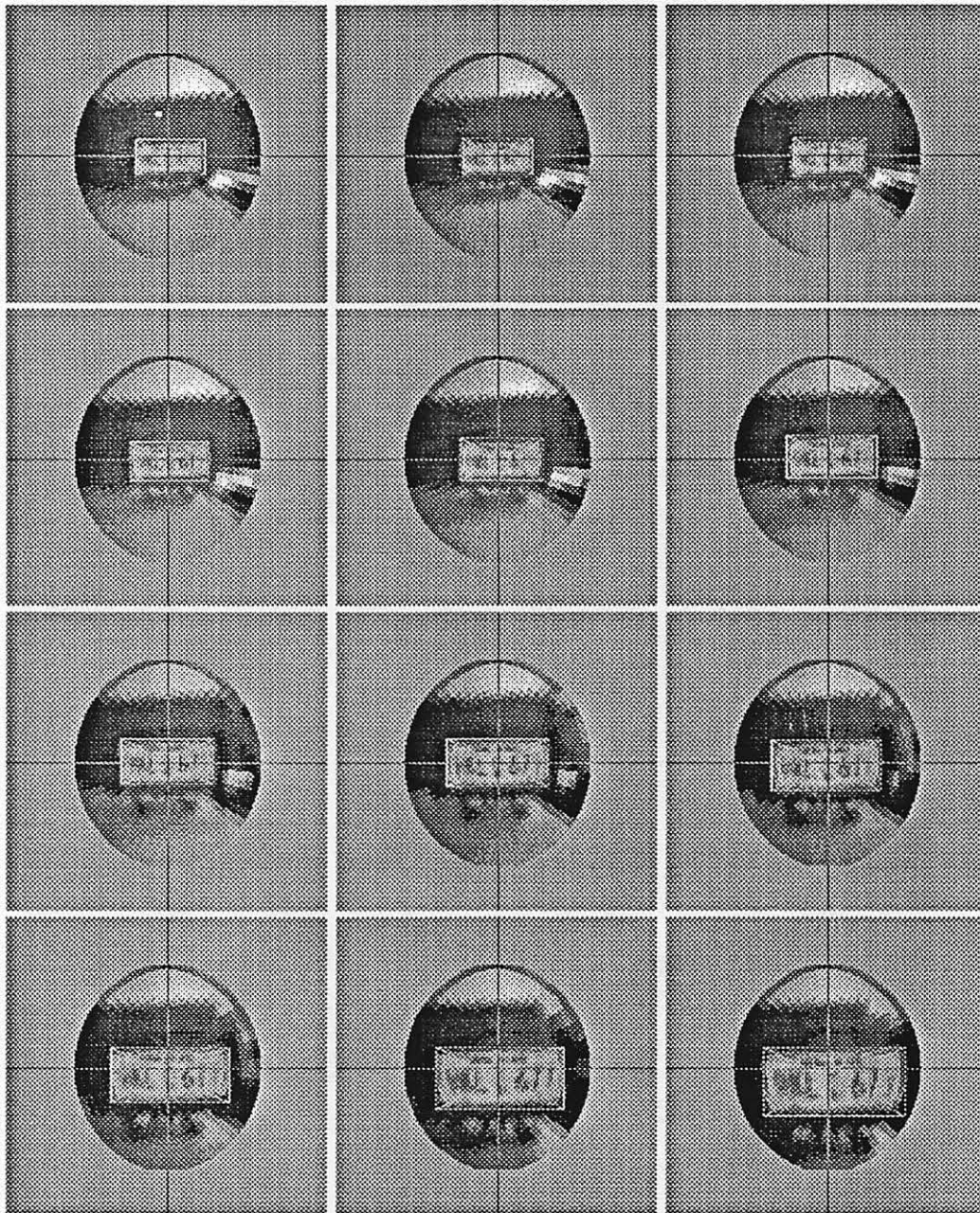


Figure 5.3: Sequence of images showing the results of tracking a moving license plate using a Kalman filter estimation procedure. A rectangle is drawn around the license plate showing that the license plate has been found correctly.

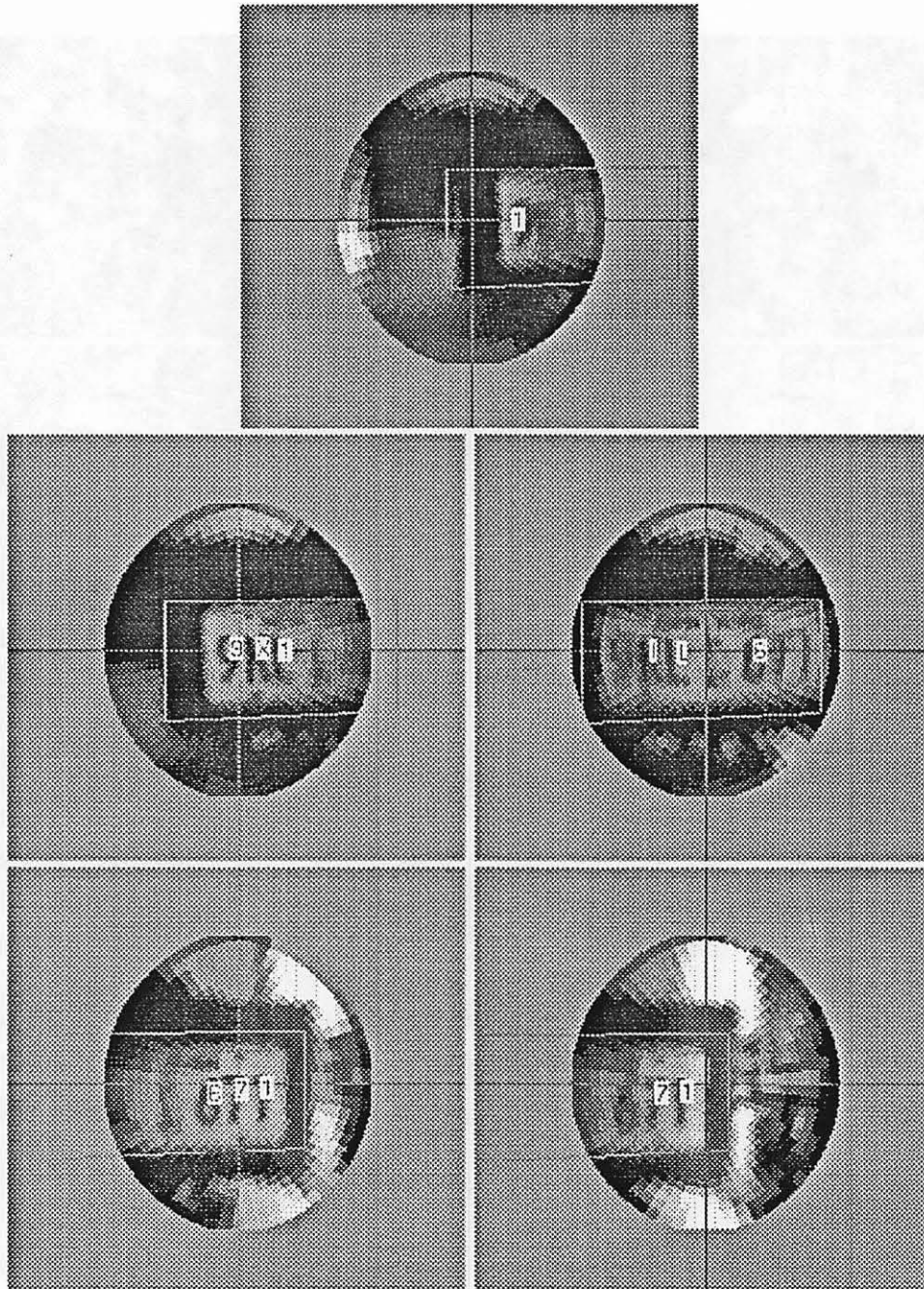


Figure 5.4: The OCR results obtained with several fixation points while the license plate was still moving. The OCR results are superimposed on the images. The actual text on the license plate is "9KL 671". The program output was "9X1L 6711".

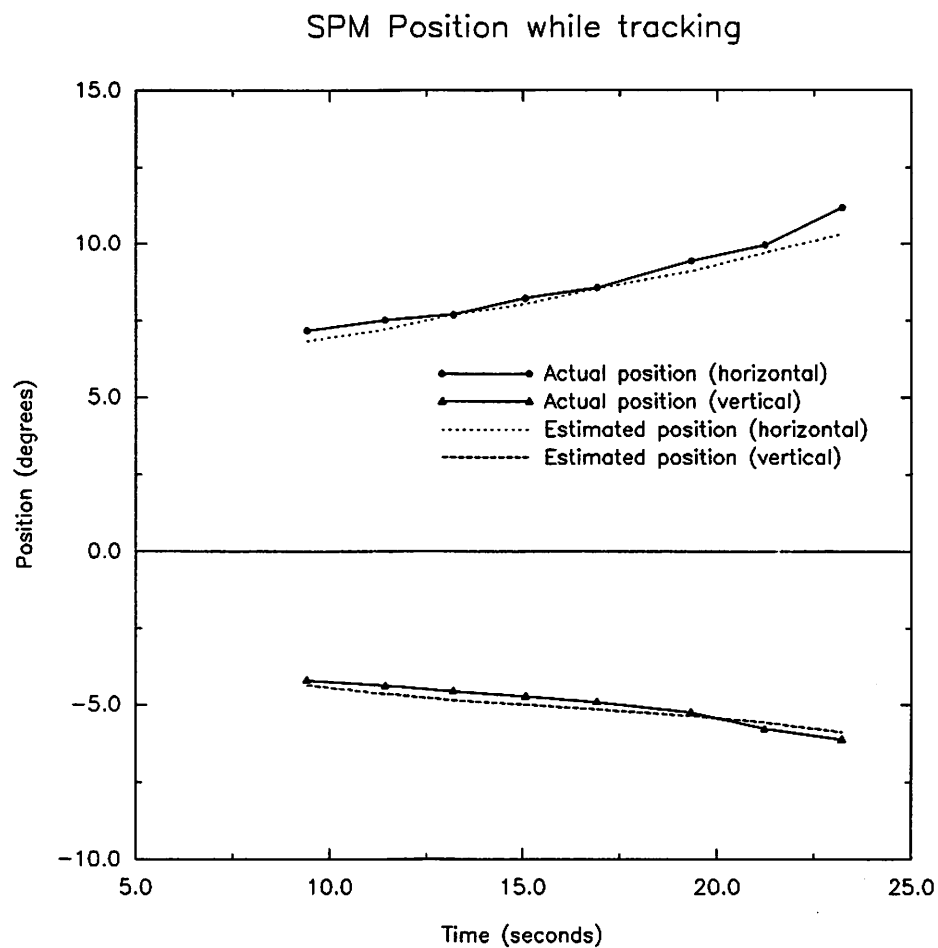


Figure 5.5: Horizontal and vertical actual and estimated position of the motor while tracking a license plate on the moving toy car.

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.

APPENDIX 2

Pocket PhotoMesa: A Zoomable Image Browser for PDAs

Amir Khella

Human-Computer Interaction Lab
Institute for Advanced Computer Studies,
Computer Science Department
University of Maryland,
College Park, MD 20742
+1 301 405-2764
akhella@cs.umd.edu

Benjamin B. Bederson

Human-Computer Interaction Lab
Institute for Advanced Computer Studies,
Computer Science Department
University of Maryland,
College Park, MD 20742
+1 301 405-2764
bederson@cs.umd.edu

1. ABSTRACT¹

Small devices such as Palm and Pocket PC have gained wide popularity with the advance and affordability of mobile technologies. Image browsers are among popular software applications on small devices. The limitations introduced by these devices such as screen resolution, processing power and storage impose a challenge for multimedia applications designed for larger displays to adapt to small screens. For an image browser, layout of images and navigation between them are critical factors of the users' experience.

Motivated by these challenges, we developed Pocket PhotoMesa: an image browser for the pocket pc that employs quantum strip Treemaps for laying out images and Zoomable User Interfaces for navigation. In this paper, we discuss the development of Pocket PhotoMesa and we describe a usability study comparing the performance and users' experience using Pocket PhotoMesa and ACDSee image browser (a current commercial offering) for the Pocket PC.

1.1 Keywords

Image browsers, Information visualization, mobile devices, mobile multimedia, Treemaps, Zoomable User Interfaces (ZUIs), Animation, Graphics, Pocket PC.

¹ 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.

MUM 2004, October 27-29, 2004 College Park, Maryland, USA. Copyright 2004 ACM 1-58113-981-0 /04/10... \$5.00

2. INTRODUCTION

The past decade witnessed a major advance in the development of mobile technologies that provided ubiquity and affordability of small devices, fitting every day's needs and everyone's pocket. Starting in the mid nineties, several companies introduced monochrome portable displays for scheduling and address books. Few years later, Pocket PCs were introduced with color screens, more processing power, and larger storage. However, many limitations still exist for application development on mobile devices: screen resolution and size, limited processing power and stylus interaction are among the toughest challenges for mobile application developers.

Popular applications on PDAs include Personal Information Managers (PIMs), file explorers, board games and image browsers. Due to limited screen size and resolution, image browsing applications use scroll bars to cycle through image thumbnails and locate images of interest. Since scroll bars require finely tuned pointing skills on small devices, we designed and implemented an image browser that eliminates the need of scroll bars. We use Treemaps [3] to layout image thumbnails on a single screen and Zoomable User Interfaces (ZUI) [1] to navigate through images. In this paper, we show our design and the results of a usability study that investigates users' performance and satisfaction with our interface compared to a traditional image browsing interface.

We believe that laying out images on a single screen in groups corresponding to physical folders will help users to visually identify themes of each group of images and be able to locate images faster than using traditional interfaces that display only folder names and We also believe that animated zooming will enable users to maintain the navigation context and improve the users' experience.

3. RELATED WORK

3.1 Zoomable User Interfaces (ZUIs)

Zoomable User Interfaces present users with a single view of large information space populated with graphical objects (images and image groups in our case). The interface allows users to navigate the object hierarchy using smooth animated zooming through different levels of the details. Initially, a ZUI renders the information space in a single screen allowing users

to get an overview of the information domain allowing them to identify themes and patterns in the big picture.

ZUIs were introduced more than thirty years ago in Sketchpad interface [4] which implemented an interactive object oriented 2D graphics system that enabled zooming and rotation of rendered objects. Almost a decade later, several systems started implementing interactive zooming: Spatial Data Management System (SDMS) [5] implemented two levels of semantic zooming, Pad and Pad++ [1] were developed as toolkits for building Zoomable User Interfaces. Zooming has also been a component of several other interfaces and toolkits developed later. Two of the major zooming toolkits available are Jazz [7], and its successor Piccolo [8], a toolkit for interactive structured graphics available in Java and C# on the desktop and on Pocket PC. These toolkits have been used in several domains such as slide show presentations [2], navigating ontology information [9], image browsing [10] as well as several other applications.

ZUIs have shown a statistically significant interaction improvement over several image browsing interfaces, but have not yet been shown to outperform traditional thumbnail grid interfaces [11]. The same study concluded that the number of images displayed within the browser is an important factor for users' performance and error rate. However, we believe that the potential for ZUIs is more promising on small devices when screen space is at an even greater premium.

A basic characteristic of ZUIs is that objects can be rendered quite small when you zoom out. Since image thumbnails can be difficult to understand when they are small, it is crucial that the thumbnails focus on the relevant part of the image. One approach to this is via automatic image cropping to generate thumbnails focusing on the salient parts of the image [6]. We think this is an important approach that should be considered in photo browsers, but is not one that we have pursued in Pocket PhotoMesa.

3.2 Treemaps

Treemaps are space-filling visualizations for large hierarchical datasets where the display area is divided into several rectangles whose areas correspond to some attribute of the dataset. Among the algorithms used for the Treemap layout are slice and dice [3], clustered [12], squarified [13] and strip Treemaps [16]. Treemaps are used in several visualizations such as SmartMoney's market map [14], image browsing [15] and many other domains. In image browsing, Treemaps suffered from the problem of aspect ratio: thumbnail sizes varied from one group to another. Quantum treemaps and quantum strip treemaps [16] solved the aspect ratio problem by using fixed size elements (quantum) across different areas.

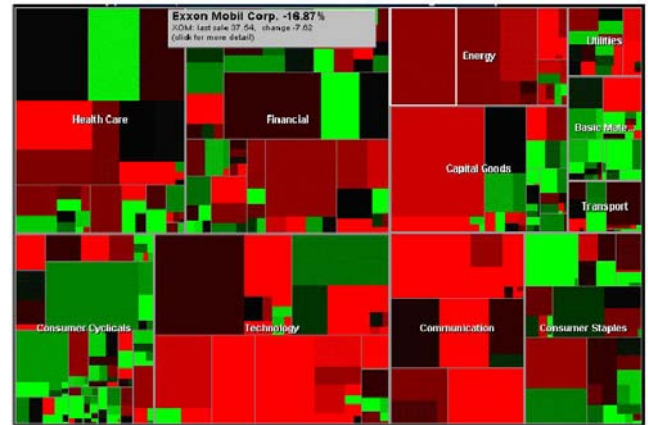


Figure 1: SmartMoney (www.smartmoney.com): An example of the use of treemaps in practice.

3.3 Desktop PhotoMesa

Desktop PhotoMesa is a zoomable image browser for the desktop PC (www.photomesa.com). The browser enables users to view hierarchies of directories in a Treemap layout and browse through them in an animated navigation mechanism. PhotoMesa uses quantum strip Treemaps to arrange images over a single screen. It also uses semantic zooming to display different thumbnail resolution at each level of detail. PhotoMesa is efficient for browsing large set of images, which would otherwise span several screen pages and requires the use of scrollbars in traditional image browsers.

Figure 2 shows 132 images loaded into PhotoMesa from 8 groups into a single screen. When users click inside a group, the view is smoothly zoomed into this directory. When the user is zoomed in to a group, a viewfinder, which can grow and shrink in size using the mouse wheel, allows users to specify a subset of the group images to zoom into or double click on a single image to zoom into a full resolution version of the image. At any time, users can press the right mouse button to zoom out to the previous zoom level. Users can also use arrow keys to navigate between images and groups. When the cursor is left to dwell over a thumbnail for a short period, a 200 pixel wide preview of this image is displayed in a tool tip overlay fashion (shown in the figure). The preview is removed as soon as the mouse moves.

Desktop PhotoMesa is implemented in JAVA using the Jazz toolkit [7]. Jazz is a polythetic structured graphics toolkit developed at the Human Computer Interaction Lab to support building interfaces in Java. The implementation of Jazz empowers programmers to create interfaces in a scene graph object oriented fashion. The scene graph is a hierarchy of nodes that represent relationships between objects. Functionality is built up by composing a number of simple objects. Complexity is therefore tackled by small reusable nodes that build up the scene graph.



Figure 2: Desktop PhotoMesa (www.photomesa.com)

4. POCKET PHOTOMESA

We faced several challenges during the design and the implementation phase to port PhotoMesa from the desktop to the Pocket PC. Limited interaction on the Pocket PC and small screen space imposed many restrictions on the interface design. In the following subsections, we present these challenges and the proposed solutions.

While the progress in mobile technologies has been promising, mobile processors are still not powerful enough to cope with the increasing demands of multimedia applications. Current PDAs are powered by 60-600 MHz processor with 16 to 64 Megabytes of memory. The processing power and memory limitations were critical factors in developing efficient algorithms and interaction techniques tailored to fit these particular devices.

Another limitation is the screen size and resolution. Pocket PCs have a screen resolution of 240x320 with 8 to 16 bit color depth. It is obvious that with this limited screen size, there is a maximum number of images that can be displayed on one screen. Moreover, with the current color depth, it is even harder to identify smaller thumbnails. A final limitation was imposed by the stylus input mechanism. A stylus has three modes of operation: up, down, and tap-and-hold, as opposed to the mouse having six modes: mouse over, left button up, left button down, right button up, right button down, and wheel scroll. It is obvious that the lack of these extra modes using stylus interaction introduce major design limitations, especially that all these interactions modes are used in the desktop version of PhotoMesa.



Figure 3: Three views of Pocket PhotoMesa at different zoom levels (www.photomesa.com).

4.1 Pocket PhotoMesa Interface Overview

Pocket PhotoMesa tackles the problem of limited screen space by using quantum strip Treemaps to efficiently layout images on the screen, minimize the amount of white space and render fixed size thumbnails for all the images.

Using the stylus, users can perform the following functionalities:

- Zoom into a specific group by tapping into a white space inside the group or the group's name.
- Zoom into an image by tapping on the image's thumbnail.
- When zoomed into an image, tapping on the image renders the full resolution version and enables users to pan around by dragging the stylus or go back to the fit-screen image mode by tapping on the image again.
- Users can also select an area of the image to zoom into by tapping and dragging to draw the zoom viewport.
- Tapping on a white space within a current level brings the user up one level.
- Hardware keys and toolbar icons can be configured to provide more functionality, such as reordering the group to fill the screen and predefined zooming levels.

4.2 Pocket PhotoMesa Implementation Overview

While the original PhotoMesa was developed using Jazz toolkit, Pocket PhotoMesa was implemented from scratch without the use of any existing API. Several factors influenced this decision: First, there is no structured graphics toolkit available for building applications on the Pocket PC. In addition, Jazz was not suitable since the Java runtime environment is neither stable nor fast enough on the Pocket PC. Now, Piccolo.NET is available for Pocket PC (www.cs.umd.edu/hcil/piccolo), and so at this point, we probably would have chosen to develop Pocket PhotoMesa using Piccolo.NET.

In our case, it was essential to develop a tailored implementation to optimize the application for maximum performance. The bottleneck was rendering smoothly animated zooming transitions. At each step of the zooming, many arithmetic operations were involved to interpolate the position of visible images from the initial to the final position. These calculations were taking more time than rendering the visible portion of the canvas, which caused jagged animations. A successful solution for this problem was to pre-compute all the intermediate positions of each image and store them in a temporary structure. This way, at each zooming step, only the rendering overhead is considered which was fast enough to give a smooth transition. Other optimization techniques were used at several parts of the implementation to improve the performance of the interface.

The application was implemented in Microsoft Embedded Visual Studio 3.0 using MFC and Pocket PC SDK and consists of approximately 10,000 lines of code.

5. USABILITY STUDY

An experiment was conducted to compare Pocket PhotoMesa to Pocket ACDSsee (www.acdsee.com). The split interface of

ACDSee shown below is a traditional image browsing interface based on choosing a group of images (folder) to browse, and displays the images in fixed thumbnail size, scrollable interface. It does not use any animated transitions.



Figure 4: Pocket ACDSee (www.acdsee.com)

5.1 The Experiment

We designed an experiment to answer the following questions:

- Is it better to fit all images in one screen or in multiple screens with scrollbars?
- What is the best number of images displayed on a single screen so that thumbnails are still identifiable?
- Do Treemaps provide a better layout than a traditional layout providing equal size rectangles to every group?

Our hypothesis was that the use of an image browser that lays out all the images efficiently in one screen would enable users to quickly locate images of interest by visually identifying the themes in each image group and remembering the location of a previously visited image. Moreover, we thought that the use of animation in zooming will improve users' satisfaction while not having a significant effect on the time to locate images.

Our independent variable is the application used. For this variable, we have three treatments:

- Pocket PhotoMesa with animated zooming
- Pocket PhotoMesa with single step zooming
- Pocket ACDSee

Our dependent variables are:

- Objective: The time required to locate a specific image.
- Subjective: User satisfaction.

The experiment was run within subjects, where each used both Pocket PhotoMesa and Pocket ACDSee for the tasks (order of use was randomized between participants to insure balance). Each interface was used with a different set of images to cancel learning effects. To insure that users use visual identification of images in Pocket PhotoMesa, group labels were disabled. The entire experiment took approximately 30 minutes per

participant. The experiment had 15 participants who are all computer science students, of which two are females. All participants were already familiar with pen based interaction and most of them own a Palm or Pocket PC.

After a training period with each interface, users were given a couple of minutes to navigate the interface and remember the location of the images. Users were then asked to locate 5 different images. These five images were given to the users via a written textual description, printed color version, and shown on the Pocket PC screen for 2 seconds. Target images were carefully chosen for each task to ensure a balance between tasks. Tasks ranged in difficulty from locating one of many existing images from description to locating a visually ambiguous image displayed for 2 seconds. By visually ambiguous we mean that the thumbnail of the image is visually similar to some other thumbnails.

We chose sets of 75 images categorized into 6 groups from the Corbis image library (www.corbis.com). The number of images was carefully chosen after running two pilot experiments with 100 images and the users' feedback was clear that identifying thumbnails at this small size was too difficult. A third pilot experiment was run with 75 images and users were able to visually identify most thumbnails on the screen.

5.2 Results

5.2.1 Quantitative results

The compilation of background surveys showed that the participants had mixed backgrounds. All the users had some background using pen based devices (most of them use Palms), mostly for appointments and contacts. None of the participants used it for image browsing.

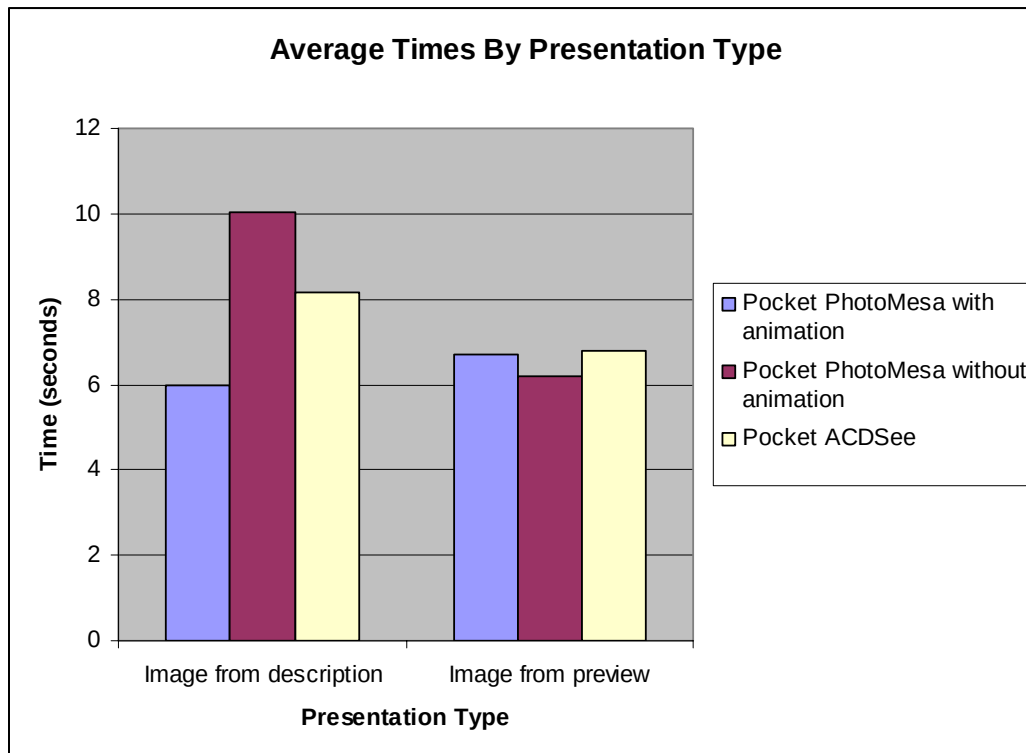
An analysis of the results shows that we got the best results using Pocket PhotoMesa without animated zooming when the image has been previously seen. Pocket PhotoMesa with animated zooming gives the best average time for locating images from a written textual description.

The total average time for locating an image on Pocket PhotoMesa with animation was 6.2 seconds, the time for the same interface without animation was 7.4 seconds and finally the average time using Pocket ACDSee was 6.4 seconds.

The previous analysis shows that there was no significant difference in time between Pocket PhotoMesa with animation and Pocket ACDSee, but interestingly, users' performance slowed down by over 1 second when animation was removed from Pocket PhotoMesa.

5.2.2 Subjective satisfaction results

On a scale of 1-9, Pocket PhotoMesa scored an average ease of use of 7.5 and interface enjoyment of 6.6 while ACDSee scored 6.2 and 4.8 respectively.



All users found that non-animated zooming was helpful for them to perform the given task (non-animated zooming satisfaction of 4.5/5) and screen layout was useful in arranging the images into equal sized thumbs wasting the least amount of screen real estate possible (layout satisfaction of 4.25/5). When it came to the use of animated zooming, users had mixed opinions: While most of them agreed that the animation increased slightly the time it takes to find the image, they mentioned that it helped them to maintain the context of navigation.

Participants also agreed that the biggest advantage of having all the image thumbnails laid out in one screen is to visually identify themes from colors and contrasts of thumbnails. During the experiment, we hid the group labels in Pocket PhotoMesa to ensure that the navigation would be based on visual grouping of themes and fast identification of thumbnails by targeting a candidate group theme that contains the image of interest. Users easily remembered the constant location of thumbnails provided by the Treemap layout and tend to zoom out to the original view and navigate to another image.

Users also agreed that when the image of interest does not have distinct color theme or contrast features, the task of identifying the image thumbnails visually became difficult at the level of detail showing all images. They also stated that the folder names in ACDSee helped them identify the themes but it was tricky during one of the tasks (e.g. to identify the image of a kangaroo, users using ACDSee went to inspect the folder named "animals" while the original image was in folder named "Australia". To perform the same task on Pocket PhotoMesa, users were relying on their mental model of a kangaroo (light brown vertical shape, long tail and small head, usually exists in the desert) to visually identify thumbnails having mostly brown colors.

Three users said that the thumbnail sizes were large enough to identify average colors and patterns but were too small to

identify shapes if the objects don't occupy a good percentage of the image.

5.2.3 Observations

Our major observation for the Pocket PhotoMesa interface is that almost all the users tried to take advantage of the screen layout to reduce interaction steps whenever possible: When they were given a task, they usually start by scanning the interface looking for a specific color or intensity. They were also limiting their visual search to one or more groups that have promising color themes for the target image. If users did not find the target image in the first 5-10 seconds, they start navigating the groups using the stylus. Most users were able to identify most of the images without the need to navigate the interface.

We also observed that in Pocket PhotoMesa, it was easier to locate images from memory than to locate it by description. For example, browsing for a picture of a zebra, users were not thinking about locating an image of an animal, instead they were looking for an image having a pattern of white and black vertical stripes. When they were given a task to locate a chessboard, it took them more time since they board was slightly oriented and had an unusual color theme.

While the interaction (zooming in and out using stylus on different areas of the screen) was not intuitive, users learned to use it quickly and had no problems performing the given tasks. The rate of errors due to wrong interaction was negligible.

6. CONCLUSION

While the quantitative results did not show a significant time improvement in locating images using a zoomable user interface with Treemap layout, user satisfaction showed that the interface was easy and fun to use. We believe that the main use of the interface is exploring and navigating rather than locating images. A search option that finds images by name or by average dominant color will improve the time locating specific images. Since all the images fit in a single screen, thumbnail

size depends on the number of images. We believe that having more than 75 images on the pocket pc screen will increase the time to locate a specific image since some users already had difficulties working with the current thumbnail size. We have found that Pocket PCs are not usually used to hold a large number of images because of their storage limitation. Typical Pocket PC users hold pictures of their families, some memorable moments, or some portfolio work, hence 75 images is a good upper bound for such typical usage.

7. FUTURE WORK

Integration of more navigation controls and search features are on the top of our list for the next version of Pocket PhotoMesa.

8. REFERENCES

- [1] Bederson, B., Meyer, J., (1998), Implementing a Zooming User Interface: Experience Building Pad++, Software: Practice and Experience, 28 (10), pp. 1101-1135.
- [2] Good, L., and Bederson, B.B. (2002) Zoomable User Interfaces as a Medium for Slide Show Presentations, Information Visualization, pp. 35-49.
- [3] Johnson, B. and Shneiderman, B., Treemaps: A Space Filling approach to the visualization of hierarchical information structures, in Proc. IEEE Visualization '91, pp. 284-291, San Diego, IEEE Computer Society Press.
- [4] Bederson, B.B. et al., Pad++: A zoomable graphical sketchpad for exploring alternate Interface physics. J. Vis. Lang. Comput., 7:3, Mar. 1996.
- [5] Herot, C.F., R. Carling, M. Friedell, and D. Kramlich. (1980). A Prototype Spatial Data Management System. Computer Graphics, 14(3), 63-70.
- [6] Suh, B., Ling, H., Bederson, B. B., & Jacobs, D. W. (2003). Automatic Thumbnail Cropping and Its Effectiveness. UIST 2003, ACM Symposium on User Interface Software and Technology, CHI Letters, 5(2), pp. 95-104.
- [7] 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), pp. 171-180.
- [8] Bederson, B. B., Grosjean, J., & Meyer, J. (2004). Toolkit Design for Interactive Structured Graphics. IEEE Transactions on Software Engineering, 30(8).
- [9] Suh, B., & Bederson, B. B. (2002). OZONE: A Zoomable Interface for Navigating Ontology Information. In Proceedings of Advanced Visual Interfaces (AVI 2002) ACM Press, pp. 139-143.
- [10] 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), pp. 71-80.
- [11] Combs, T. T. A., and Bederson, B. B. (1999). Does Zooming Improve Image Browsing? In Proceedings of Digital Library (DL 99) New York: ACM, pp. 130-137.
- [12] Wattenberg, M. (1999). Visualizing the Stock Market. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 99) ACM Press, pp. 188-189.
- [13] Bruls, M., Huizing, K., and 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.
- [14] Wattenberg, M. Map of the Market (1998). <http://www.smartmoney.com/marketmap>.
- [15] Bederson, B. B. (2001). Quantum Treemaps and Bubblemaps for a Zoomable Image Browser. In Proceedings of User Interface and Software Technology (UIST 2001) ACM Press, (in press).
- [16] 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), pp. 833-854.

APPENDIX 3

SlideBar: Analysis of a linear input device

LESLIE E. CHIPMAN, BENJAMIN B. BEDERSON and JENNIFER A. GOLBECK

Department of Computer Science, Human-Computer Interaction Laboratory, University of Maryland, College Park, Maryland, USA; e-mail: {gchipman, bederson, golbeck}@cs.umd.edu

Abstract. The SlideBar is a physical linear input device for absolute position control of 1° of freedom, consisting of a physical slider with a graspable knob positioned near or attached to the keyboard. Its range of motion is directly mapped to a one dimensional input widget such as a scrollbar. The SlideBar provides absolute position control in one dimension, is usable in the non-dominant hand in conjunction with a pointing device, and offers constrained passive haptic feedback. These characteristics make the device appropriate for the common class of tasks characterized by one-dimensional input and constrained range of operation. An empirical study of three devices (SlideBar, mouse controlled scrollbar, and mousewheel) shows that for common scrolling tasks, the SlideBar has a significant advantage over a standard mouse controlled scrollbar in user preference. In addition, users tended to prefer it over the mousewheel (without statistical significance).

1. Background

Personal computers have come to be so popular largely because they are multi-function devices. They consist of general-purpose processors, general-purpose input devices, and general-purpose output devices. This is a fundamental design feature of today's personal computers, and it works remarkably well. However, this design represents one side of a trade-off that results in a computer that can do many tasks well, but does not necessarily do each individual task as well as a specialized device could. Certain very common tasks, such as scrolling, are a case in point. It is no surprise then, that design, implementation and evaluation of scrolling mechanisms and pointing devices are well-studied areas.

In traditional graphical user interfaces (GUIs), a document such as a spreadsheet, text file, or file list in a folder is often larger than the viewing window. Thus, the user views only part of the document at a time. To view portions of the document outside the viewing window,

one must move the window relative to the document. Scrolling behaviour is an extremely frequent task in GUIs, and it becomes even more frequent in browsing World Wide Web pages.

Scrolling is just one example of a common class of computer input tasks characterized by having one dimension of freedom and a constrained range of operation. Some other examples are selecting from menus, zooming in and out of documents and operating onscreen widgets, e.g., slider widgets. Even if there is not an explicit mouse controlled widget, the task can still exist and in current applications is often controlled via keyboard or with specific selections from menu items. For instance, photo browsing applications commonly have a menu for controlling the magnification with common values such as 50, 100 or 200%, or in specific modes, the user may increment or decrement at predefined values with a mouse click, giving limited control of a task that actually has a smooth, broad range of input values.

Since this common class of tasks is broad and well defined, it seems logical to consider an input device tailored specifically for this purpose. In this paper, we introduce the *SlideBar*, a linear input device for absolute position control, which we built with the goal of providing users with an input device better suited for linear tasks. In this paper, we delve into the reasons we expect this device to be well suited for linear tasks and give the results of a study comparing it to common input devices for the specific linear task of scrolling.

As computer users, we have become accustomed to using the standard mouse and keyboard as input devices and so it may be difficult to recognize some of their inherent drawbacks. There is more willingness to incorporate new devices into the interface, especially with the increasing popularity of laptop and even smaller computers where the limitations of a mouse become more pronounced. We believe the SlideBar

offers a large enough advantage to justify being built in to some computers.

1.1. *Scrolling*

We have observed that users scroll primarily in two modalities. The first modality is to read a document, where the document is slowly scrolled a few lines at a time so that the user can easily track the line being read. We will refer to this as *read mode*. The second modality is to scan a document, typically when a user is searching the document to find a specific part of it. In this mode, the user typically scrolls the document back and forth through large areas. We will refer to this as *scan mode*. After the specific area sought for is found, the user typically switches to read mode, and reads the document line by line.

1.2. *Scrolling mechanisms*

The most basic scrolling mechanism is the standard scrollbar controlled by a mouse. Other common methods include the mousewheel, keyboard, touchpad or joystick.

1.2.1. *The scrollbar:* A scrollbar is an on-screen widget that the user controls with a general pointing device, such as a mouse, joystick, or touchpad. It supports absolute motion by acquiring and dragging the movable ‘thumb’, which allows the user to scroll through the document to any given position in a simple motion, independent of the document length. This is most appropriate for scan mode and the thumb provides lower resolution control for longer documents. Scrollbars also support relative motion by clicking on the arrows at the ends of the scrollbar to scroll one line at a time, or on the remaining space between the arrows and the handle, which scrolls several lines at a time. This is appropriate for read mode.

The scrollbar appears to be an excellent control mechanism for scrolling since it provides support for relative and absolute motion and for both modalities of use. Of course, it does work well, and we all use it regularly. However, scrollbars do have several shortcomings. The primary one results from the fact that users must use a general-purpose pointing device to control it. Each movement of the user’s hand between the keyboard and the mouse takes time. A study by Douglas and Mithal found it takes 0.67 s for users to acquire a mouse and 0.44 s for a keyboard joystick (Douglan *et al.* 1994). We would expect a time

somewhere in this range for acquiring a keyboard mounted SlideBar.

Even if the user’s hand is already on the mouse, the user must then move the pointer on the screen to the appropriate part of the scrollbar. Since common scrollbars are only about 15 pixels wide, acquiring the scrollbar takes up to two seconds (Zhai *et al.* 1997), and impacts performance, in accordance with Fitts’ Law (Fitts and Peterson 1954). Another shortcoming of scrollbars is that the user must take his or her eyes off the document and focus on the scrollbar to know where they are in the document.

Finally, for ongoing scrolling tasks that last a long time, the user must hold down the mouse button for the full length of the scrolling task (if using the thumb), or regularly click (if using the trough or arrows). For users with Repetitive Stress Injury (RSI), this continuous force required of the fingers can be destructive.

These actions are in conflict several common goals of good user interfaces: unobtrusiveness, transparency and ease of use. A good interface diverts a minimal amount of the user’s attention and effort away from the primary task of viewing a document in order to explicitly manipulate GUI widgets.

1.2.2. *The mousewheel:* A mousewheel is commonly built into mice. It is a wheel between the two primary input buttons, which can be mapped to control the scrollbar with relative motion. It can be used to scroll a document a small amount with a corresponding wheel movement. For this reason, it supports read mode well. Because the range of motion is limited, the user must repetitively spin the wheel to scroll more than a few lines, making it ineffective for scan mode in all but the smallest documents.

Some applications support rate scrolling with a mousewheel. With rate scrolling, the user presses and holds the mousewheel down, then the direction of movement of the mousewheel provides the direction of scrolling and the amount of displacement controls the rate of scrolling. In a comparison of various scrolling devices and mechanisms, Zhai *et al.* (1997) showed the performance of a mousewheel to be similar to that of the standard mouse. They also show rate scrolling to be more effective than standard scrolling when using an isometric joystick and indicate that the isometric nature of the device to be an important factor in rate scrolling.

1.2.3. *Arrow and page up/page down keys:* Most keyboards have specialized keys for scrolling. The up and down arrow keys move the cursor up and down one line with a document. When the edge of the screen is reached, the document is scrolled. The Page Up and Page Down keys scroll several lines at a time. The actual

amount the Page Up and Page Down keys scroll is specified by the application, but typically the document is scrolled some significant portion of the screen height. Though arrows keys are useful in read mode, they provide only relative control and so are very limited for use in scan mode. With the exception of specialized keys such as the 'Home' key, they provide no method for moving quickly to an absolute position in the document.

1.2.4. Touchpad: Some laptops with touchpads can be used for controlling the scrollbar. Sliding the finger along the right edge of the touchpad provides relative control of the scrollbar. This supports read mode well. It is limited in scan mode by its relative nature, much as a mousewheel is, though constantly repositioning one's finger on a touchpad may be less fatiguing than working a mousewheel over and over. It is possible to support rate scrolling with this input device, but as mentioned above, lack of isometric feedback is an issue.

1.3. The scrollbar

The SlideBar is a physical linear input device that can be used to control absolute position of a 1° of freedom parameter (figure 1). It is intended primarily for scrolling, although it can also be used to control any other one-degree of freedom application. It is designed for non-dominant hand use, to be used in tandem with a mouse or other general-purpose pointing device in the dominant hand.

Because the SlideBar is an absolute positioning device, the user can immediately move through the entire document, from top to bottom and through all

the intermediate positions, although with limited resolution. Because the document being scrolled has a top, bottom and a length, and the SlideBar has a top, bottom and a length, there is always a direct linear mapping between the SlideBar and the scrolled document.

One issue with absolute positioning is that if a document is scrolled by a mechanism other than the SlideBar, the position of the SlideBar may no longer correspond to the viewing window of the document. The control software has been designed so that as soon as the SlideBar is moved at all, the document viewing windows jumps to the position that corresponds to the SlideBar. This is important since we hope that the prototype SlideBar can take advantage of the fact that it has a physical position in space, which allows users to develop awareness of what portion of the document they are viewing.

1.4. Previous work

Proprioception is the ability of people to sense the position and movement of their bodies, and has been studied for several decades. It has been shown that humans exhibit a remarkable ability to remember the position of their limbs (Boff *et al.* 1986). The SlideBar's absolute linear mapping of document position should benefit from these effects.

The use of haptic feedback in computer input devices is also a well-studied area. *Passive haptic feedback* refers to a mechanism where the user can control and feel the position of the input device but where there is no response or control from the computer, in contrast to force or active feedback. Note that this terminology

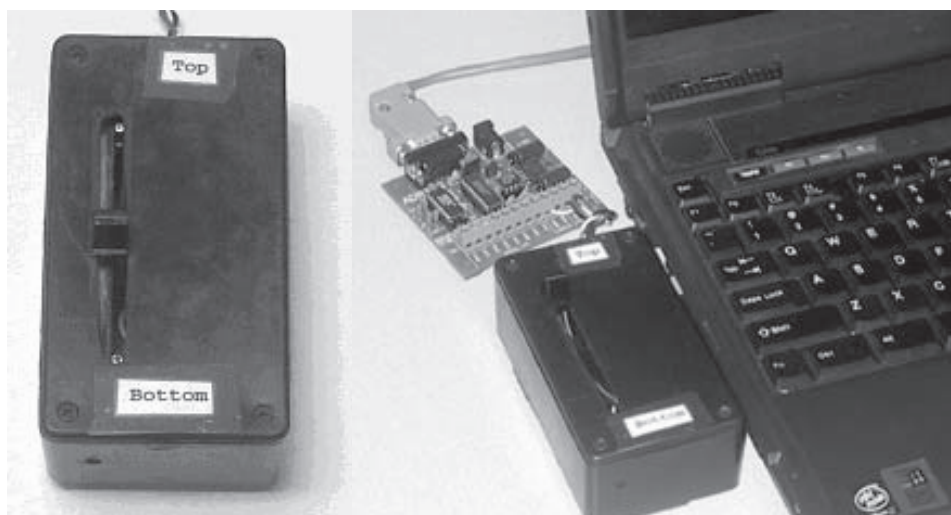


Figure 1. The prototype scrollbar.

differs from that used in psychology where passive positioning means that one's body was moved externally by an apparatus and active refers to positioning done by voluntary muscle control. Fitzmaurice *et al.* (1995) and Fitzmaurice and Buxton (1997) showed the advantages of having more specialized, graspable input devices that operate via passive haptic feedback. Studies (MacKenzie 1992, Lindeman *et al.* 1999, Wang and MacKenzie 2000) have also shown the importance of placing constraints on object movement for passive haptic feedback. The SlideBar should benefit from this as well since it is a constrained input device, physically moving only within the range of the task.

User studies dealing with comparative analysis of input devices (Kabbash *et al.* 1993, 1994, Zhai *et al.* 1997) have shown that given the chance to use both hands, people will naturally use the non-dominant hand for rough positioning and the dominant hand for fine positioning. To study two-handed input, Buxton and Myers (1986) used linear input devices in participants' non-dominant hand. For a scaling and positioning task, a physical linear slider with relative control for the scaling portion of the task was used. In a scrolling task, two touchpads were used for linear control. One gave absolute position control and the other gave relative control of the scrolling task. This gave users a different linear device corresponding to the different scrolling modes of scanning and reading. The SlideBar differs from these input devices in that only the SlideBar provides both absolute positioning and passive haptic feedback with physical constraints. Our purpose is to establish the benefit of the SlideBar, which is different from their purpose of studying the effect of two-handed input. We do expect two-handed input to be one of the SlideBar's advantages and their study establishes this effect.

The combination of two handed input, absolute positioning and constrained passive haptic feedback makes the SlideBar a unique computer input device for 1° of freedom tasks, which we believe it will excel at.

2. Implementation of the Slidebar

The current implementation of the SlideBar is a proof of concept prototype that consists of a linear potentiometer that is wired to generate an analog voltage as the potentiometer is moved. A graspable knob is attached to the potentiometer and it is placed to the side of the keyboard. The SlideBar has a range of motion of 4.5 cm, which is designed so that moving just the wrist and fingers without moving the forearm can access the full range. The analog voltage generated by the

potentiometer is then digitized through a 12-bit A-D converter.

We wrote driver software to continuously read the SlideBar position via the serial port and generate events that are applied to scrolling. Position data for small changes is averaged, allowing for smoother control at slow scrolling rates, yet immediately applied for large position changes, maintaining the SlideBar's ability to move anywhere within a document without delay. The amount of averaging applied varies and is inversely proportional to the position change (up to a limit), so that the averaging used does not suddenly change at an arbitrary boundary. We also wrote custom applications that listen to SlideBar events and control scrolling within those applications.

The A-D converter limits the resolution of this version of the SlideBar to 4096 positions. As a comparison, scrollbars controlled by a mouse are limited in resolution to one pixel, determined by the resolution of the screen. Every time the scrollbar moves one pixel, the document scrolls a corresponding amount. For a high-resolution display of 1024 vertical pixels, a mouse controlled scrollbar provides one-fourth the resolution provided by the SlideBar prototype. With the same pixel resolution, a 40-page document and a 10-point font, the SlideBar prototype resolution is about one line of text.

However, the resolution of human movement may easily be less than that of the SlideBar. Numerous studies in joint proprioception show error to usually only a few degrees (Boff *et al.* 1986) and that the finger usually has higher error than other joints (Balakrishnan and MacKenzie 1997). But studies of finger position are finger alone, not finger and thumb grasping an object and references hint at this potential cause of inferior finger performance (Fitts 1954, Balakrishnan and MacKenzie 1997). None of these studies can be easily applied to calculate the error resolution of the SlideBar due to the various methods employed. These studies do show the error to be based on angle, not position, and here the mouse controlled scrollbar may regain some lost advantage. Because the distance of movement for a mouse is much greater than that of the prototype SlideBar, the angles traversed during scrolling are greater. Rosenbaum *et al.* (1991) found that the optimal movement amplitude for the finger is about 45° (Rosenbaum *et al.* 1991). A study with positioning of sliders did show that errors in positioning are only a few percent for slider distances greater than about 5 to 10 cm, which is comparable to the full range of the prototype SlideBar (Boff *et al.* 1986). But again, methods are not directly applicable, in this case largely because most participant movements were very large and involved primarily elbow and shoulder movements. One interesting result was that participants tended to

overestimate short ranges and underestimate longer ones. These findings will have some bearing on future SlideBar implementations.

While the prototype implementation of the SlideBar is a separate device that is placed next to the keyboard, we envision a production version that would be built into the keyboard. In addition to reducing the complexity and cost of the SlideBar, this would also improve its usability by providing stability to the device as well as putting it in a consistent and accessible position. We expect it would require only a small amount of additional area on a keyboard, making it especially desirable for laptop computers where space is a premium and a mouse is often not available.

Some software design issues exist as well. We expect the SlideBar driver would automatically switch to control the vertical scrollbar of the focused window, however, other modes may be appropriate. The SlideBar will also likely be out of sync with a document when it moves into focus, as that document will not necessarily be scrolled to the depth corresponding to the SlideBar's current position. Our intuition is to have the document snap to the SlideBar's position as soon as that position starts to change, but there may be other ways to deal with this concern.

3. Methods

3.1. Conditions and materials

We conducted an experiment to evaluate the prototype SlideBar device on a laptop computer (700 MHz Pentium III, 128 MB ram) with a screen resolution of 1024×768 pixels. There were two independent variables: scrolling device; and document length. Three scrolling devices were tested: the prototype SlideBar, a standard mouse controlling a scrollbar, and a mousewheel. No acceleration algorithms were used with the mouse or mousewheel. The three documents lengths were three pages, 20 pages and 70 pages corresponding roughly to SlideBar resolutions of; less than one pixel, half a line and two lines, respectively. Dependent variables were the time to complete a scrolling task, the index of performance for a scrolling task and subjective satisfaction. The study was set up as a within-subjects design, so each subject performed both tasks with all three devices.

Participants were first presented with a description of each device and how it worked. The study administrator demonstrated the devices and tasks on a practice document. Participants were instructed on how each exercise was to be completed, and then given unlimited time to practice device use and familiarize themselves with the tasks until they felt comfortable enough to

proceed. On both the mousewheel and SlideBar, the two least familiar devices, subjects on average spent over a minute practicing, with several subjects spending several minutes. On the scrollbar, a device familiar to all of the subjects, average practice time was still nearly 45 seconds, ensuring that the use of the device in context of the tasks was familiar.

Subjects were administered a pre-test to gather demographic information and data about their previous experience with computers and various input devices. For the actual experiment, users completed all the tasks for a given device before proceeding to the next device. The order of the devices was randomized for each subject.

After completing the tasks for the last device, users completed a post-test questionnaire that gathered their preferences and comments about the devices. They were also interviewed to gather any additional opinions about the experiment. The entire process took approximately 1 h.

3.2. Tasks

Participants performed two types of tasks. For each task, a document of random words formatted in paragraphs was presented to the participant. Approximately one-inch square, brightly coloured icons were placed randomly in the document as targets. Possible icons were circles, squares, rectangles, ovals and triangles. Colours and icons were chosen at random for each task. Prior to each task, a full screen page presented instructions for the upcoming task, including what graphical icon to look for. A button to begin the task was placed near the centre of the screen. The task document was presented when the participant clicked the button, ensuring the starting position of the mouse was near the centre of the screen. When the document was presented, it was scrolled all the way to the top and targets were never placed on the first page. The devices were presented in random order for each participant. During a trial for a specific device the other devices were physically present, but disabled.

The first type of task was the common scrolling task of scanning a document for a target in an unknown location. A single icon target was placed randomly in the document. We refer to this as the target location task. We wished to study the effects of device resolution, especially for the SlideBar, so we used a second independent variable, document length, with three values, leading to nine total trials for this task. Participants were automatically timed from the moment the start button was clicked until they found and clicked on the target.

The second task type used a 20-page document of random words, which contained two graphical icons placed within it. The location of the first icon was randomly placed somewhere in the first half of the document. The second icon was always a fixed 10 pages from the first. Participants were asked to find and select the first target then the second target, then the first again, repeatedly, for six clicks on each target. This is similar to Fitts' (1954) reciprocal tapping task. Participants were automatically timed from the beginning of the task (start button clicked) to the selection of the first target and from each target selection to the next.

In both types of tasks, each type of scrolling described above was used. To locate the coloured icon, users progressed in 'scan mode', moving through the document quickly until they saw the icon for the first time. After roughly locating it in the document, users switched to 'read mode', moving slowly through the isolated section to bring the icon onto the screen and click it.

3.3. Subjects

There were a total of 24 participants in these experiments, 11 male and 13 female. All but three were right handed, and everyone used the mouse in their dominant hand and SlideBar in the non-dominant hand. They were drawn from a general university population, ranging in age from 18–30. Most did not have a computer science background, but used computers every day. All of the subjects were very familiar with the scrollbar, and all but two were mousewheel users.

3.4. Hypothesis

It was our hypothesis for the target location task that the SlideBar would outperform the standard mouse controlled scrollbar due to its advantages of two-handed input that eliminates the time necessary to acquire the scrollbar with the mouse. It was also expected that the SlideBar would outperform the mousewheel due to its nature as an absolute positioning device vs. the relative nature of the mousewheel. This effect may be minimized for the shorter document length.

It was our hypothesis for the reciprocal tapping task that the SlideBar would be faster than the other two devices. Only the SlideBar takes advantage of the haptic memory function of the human body and the task of repeatedly scrolling between pre-

viously located targets should take advantage of this effect.

3.5. Measures

In the target location task, time for completion was measured. The index of performance (IP) (Card *et al.* 1983) was then calculated:

$$IP = \log_2(D/W + 1)/MT$$

where D is depth of target, W is width of target (fixed for all our trials) and MT is movement time. Because the target depths were varied at random for this task, it was not possible to use movement time directly as the performance measure. We chose index of performance because it normalizes the movement time based on target depth (Hinckley *et al.* 2002). Several cases were discarded when participants attempted to use the incorrect device for the trial.

For reciprocal tapping tasks, 12 times were recorded. The first half of these was discarded to allow the participant to learn the location of the targets. Since the target distances and widths were fixed for every trial, movement time could be used directly as a measure of performance.

After the completion of the study tasks, participants completed a post-questionnaire where they were asked to rate each device for speed of use, clarity of use and satisfaction of use. They were also asked to choose a preferred device and given the opportunity to comment in detail.

4. Results

4.1. Locating a target

The mousewheel was significantly better than the other two devices for the target location task (figure 2). The SlideBar had a higher mean performance for this task than the scrollbar, but the effect was not significant. For the target location task, univariate analysis of variance (ANOVA) showed device to be a significant factor, $F(2,195) = 10.747$, $p < 0.001$. A *post hoc* Bonferroni comparison of the factors showed the mousewheel to perform significantly better than the scrollbar, $p < 0.001$, with mean IP difference of 0.41. The mousewheel performed significantly better than the SlideBar, $p = 0.047$, with a mean IP difference of 0.22. The SlideBar was not significantly better than a scrollbar, $p = 0.17$, with a mean IP difference of 0.19.

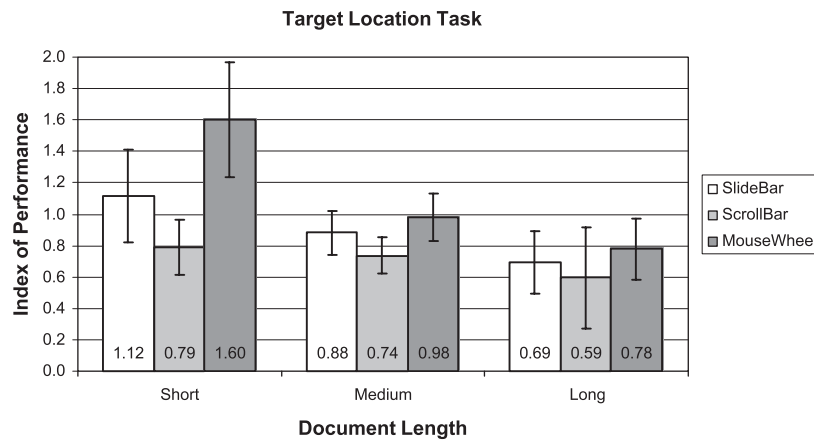


Figure 2. Target location task results. Error bars represent 95% confidence interval.

The analysis also showed document length to be a significant factor for the target location task, $F(2,195) = 14.165$, $p < 0.001$. The interaction between device and document length was significant at $p = 0.035$.

4.2. Reciprocal tapping

For the reciprocal tapping task, the SlideBar had significantly better performance than the scrollbar (figure 3). For the reciprocal tapping task, univariate analysis of variance (ANOVA) also showed device to be a significant factor, $F(2,499) = 7.79$, $p < 0.001$. A *post hoc* Bonferroni comparison of the factors again showed the SlideBar to be significantly faster than a scrollbar, $p < 0.001$, with mean times of 5.8 s for the SlideBar vs. 8.7 s for the scrollbar. Again, there was no significant difference in speed between the SlideBar and mousewheel.

4.3. User preference

In subjective preference, the SlideBar offered significant benefits over the scrollbar (figure 4). It also was preferred over the mouse wheel, but there was no statistically significant difference. Participants were asked to choose their preferred device and 14 of the 24 chose the SlideBar. Users also gave subjective rankings on a scale of 1 (lowest) to 9 (highest) for speed, clarity of use, and satisfaction. For speed, the SlideBar significantly outranked the other two devices, $F(2,69) = 15.5$, $p < 0.001$; on average the SlideBar ranked 8.08, the mousewheel ranked 5.46, and the mouse with scrollbar ranked 5.83. Users found no significant difference in clarity of use. Users were significantly more satisfied with the SlideBar than the mousewheel, $F(2,69) = 3.28$,

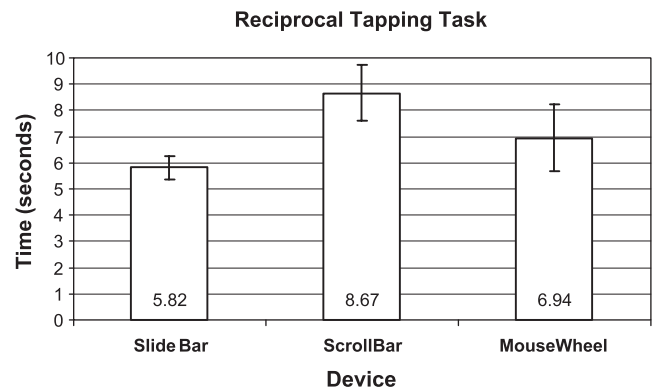


Figure 3. Reciprocal tapping task results. Error bars represent 95% confidence interval.

$p < 0.044$; on average the SlideBar ranked 7.04, the mousewheel ranked 5.5, and the mouse with scrollbar ranked 6.16.

5. Analysis

The results support our hypothesis that the SlideBar is faster than a standard mouse controlled scrollbar for the reciprocal tapping task, but it does not support our hypothesis that the SlideBar is faster than a mousewheel. Though the data does not offer a clear reason for why the SlideBar was not faster than the mousewheel, our observations of the subjects offer one explanation. When using the SlideBar, users appeared to rely on the passive haptic feedback for document positioning. Although this was not available with the mousewheel, nearly every user started off by counting the number of 'spins' of the mousewheel that separated the two icons. Users were then able to flick the mousewheel a fixed

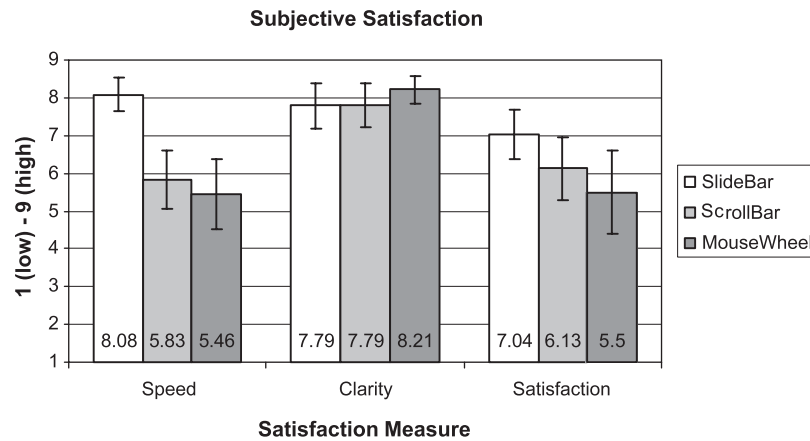


Figure 4. Subjective satisfaction results. Error bars represent 95% confidence intervals.

number of times to quickly switch between the targets, and this may have counterbalanced any benefit that the SlideBar offered.

The results do not support our hypothesis that the SlideBar would outperform the other two devices at the target location task. The motivation for this hypothesis was that the two-handed nature of the SlideBar would give it an advantage over the other devices. Though previous research has established the advantages of two-handed input, under the parameters in this experiment, that effect did not make a significant difference. To the contrary, the results here showed that the mousewheel was significantly faster than both the SlideBar ($p = 0.047$) and scrollbar ($p < 0.001$). For this task, the scrollbar clearly underperformed. This is very likely due to the overhead of acquiring the scrollbar with the mouse at each target switch. Neither the mousewheel nor SlideBar suffers from this drawback.

The interaction between device and document length was significant. For shorter documents the mousewheel shows a gain in performance over the other devices (figure 3). The poorer performance for the scrollbar can be attributed to the fact that there is constant time required to acquire the scrollbar with the mouse, and this time becomes a larger percentage of the overall movement time for short documents. The increase in performance for the mousewheel is likely because, for short documents, it is often possible to scroll to the target in a single motion of the mousewheel, eliminating time spent re-positioning the finger on the mousewheel.

6. Conclusions

We designed the SlideBar to take advantage of the potential benefits of two-handed use and

constrained passive haptic feedback to give it an advantage over traditional scrolling devices. We tested these devices with two common tasks: scrolling a document while scanning for a target at an unknown location, and switching between two targets at known locations.

All devices showed roughly the same index of performance, except in the case of the short document. In this case, the mousewheel was shown to be significantly faster than both the SlideBar and the scrollbar. For generic scrolling, the SlideBar is a device with performance equivalent to the most popular mechanisms. This, combined with the fact that user preference was strongly in favor of the SlideBar, lends support to the need for further investigation of its applications.

It was interesting to note that users *subjectively* rated the SlideBar as the fastest device, even though the data indicates otherwise. For long documents, repetitively spinning the mousewheel makes the tasks very tedious. Whereas the SlideBar allows the user to move through the document extremely fast. But time may be lost if the target is missed and the user must scan again. In addition, once in the general vicinity of the target, the mousewheel is a very good device for operating in 'read' mode to home in on the target. In the authors' own informal experimentation with these devices, the combination of SlideBar and mousewheel seemed quite effective.

Clearly, more work will be necessary to improve the SlideBar and to better understand what tasks it is best suited for. With respect to the device itself, it will be useful to determine what parameters are necessary to improve its usability. Experimenting with factors such as range of motion and force of motion, or adding features for control of relative position, such as touch sliders or

some isometric rate device, may enhance its ability as an input device. Studies would be needed to define and optimize these benefits.

While we expect that scrolling will continue to be such a common task performed by computer users, the expense of adding a linear input device such as the SlideBar will need justification before it is included in production systems. Aside from scrolling, tasks such as controlling zoomable user interfaces and controlling on screen widgets, especially for potentially sensitive tasks like dynamic queries, may combine well with the SlideBar. The impetus behind the SlideBar is that, unlike the mousewheel or scrollbar, it allows users to take advantage of passive haptic feedback. In the reciprocal tapping task, where users knew the position of icons in the document, the SlideBar did offer significantly better performance. In other applications like zooming, where there is an immediate visual feedback on the screen, and users have a sense of what percentage of a zoom they have achieved, the haptic feedback should again offer an advantage. Further studies will be help to isolate the specific tasks that may bring out the best applications of the SlideBar.

References

- BALAKRISHNAN, R. and MacKENZIE, I. S. 1997, *Performance differences in the fingers, wrist, and forearm in computer input control*. Proceedings of CHI'97, pp. 303–310.
- BOFF, K. R., KAUFMAN, L. and THOMAS, J. P. 1986, *The Handbook of Perception and Human Performance*, Volume I. (Wiley, New York), chapter 13.
- BUXTON, W. and MYERS, B. A. 1986, *A Study in Two-Handed Input*, CHI'86 Proceedings, pp. 321–326.
- CARD, S. K., MORAN, T. P. and NEWELL, A. 1983, *The Psychology of Human-Computer Interaction*. (Lawrence Erlbaum Associates, Hillsdale, New Jersey), p. 51.
- DOUGLAS, S. A. and MITHAL, A. K. 1994, *The Effect of Reducing Homing Time on the Speed of a Finger-Controlled Isometric Pointing Device*, Proceedings of CHI'94 Human Factors in Computing Systems, pp. 411–416.
- FITTS, P. M. 1954, The information capacity of human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, **47**, 381–391.
- FITTS, P. and PETERSON, J. 1964, Information capacity of discrete motor responses. *Journal of Experimental Psychology*, **67**, 103–112.
- FITZMAURICE, G. W., ISHII, H. and BUXTON, W. 1995, *Bricks: Laying the Foundations for Graspable User Interfaces*. Proceedings of CHI'95, pp. 432–449.
- FITZMAURICE, G. W. and BUXTON, W. 1997, *An Empirical Evaluation of Graspable User Interfaces: towards specialized, space multiplexed input*. Proceedings of CHI'97, pp. 43–50.
- HINCKLEY, K., CUTRELL, E., BATHICHE, S. and MUSS, T. 2002, *Quantitative Analysis of Scrolling Techniques*. Proceedings of CHI'02 Human Factors in Computing Systems, pp. 65–72.
- KABBASH, P., MacKENZIE, I. S. and BUXTON, W. 1993, *Human Performance Using Computer Input Devices in the Preferred and Non-Preferred Hands*. Proceedings of CHI'93 Human Factors in Computing Systems, pp. 474–481.
- KABBASH, P., BUXTON, W. and SELLEN, A. 1994, *Two-handed Input in a Compound Task*. Proceedings of CHI'94 Human Factors in Computing Systems, pp. 417–423.
- LINDEMAN, R. W., SIBERT, J. L. and HAHN, J. K. 1999, *Towards usable VR: An empirical study of user interfaces for immersive virtual environments*. Proceedings of CHI'99, pp. 64–71.
- MacKENZIE, C. L. 1992, Making contact: Target surfaces and pointing implements for 3D kinematics of humans performing a Fitts' task. *Society for Neuroscience Abstracts*, **18**, 515.
- ROSENBAUM, D. A., SLOTTA, J. D., VAUGHAN, J. and PLAMONDON, R. 1991, Optimal movement selection. *Psychological Science*, **2**, 86–91.
- WANG, Y. and MacKENZIE, C. L. 2000, *The Role of Contextual Haptic and Visualization in Virtual Environments*. Proceedings of CHI'00, pp. 532–539.
- ZHAI, S., SMITH, B. A. and SELKER, T. 1997, *Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks*. Proceedings of Interact'97, pp. 286–292.

APPENDIX 4

Multi-Modal Text Entry and Selection on a Mobile Device

David Dearman
University of Toronto

Amy Karlson
Microsoft Research

Brian Meyers
Microsoft Research

Ben Bederson
University of Maryland

ABSTRACT

Rich text tasks are increasingly common on mobile devices, requiring the user to interleave typing and selection to produce the text and formatting she desires. However, mobile devices are a rich input space where input does not need to be limited to a keyboard and touch. In this paper, we present two complementary studies evaluating four different input modalities to perform selection in support of text entry on a mobile device. The modalities are: screen touch (*Touch*), device tilt (*Tilt*), voice recognition (*Speech*), and foot tap (*Foot*). The results show that Tilt is the fastest method for making a selection, but that Touch allows for the highest overall text throughput. The Tilt and Foot methods—although fast—resulted in users performing and subsequently correcting a high number of text entry errors, whereas the number of errors for Touch is significantly lower. Users experienced significant difficulty when using Tilt and Foot in coordinating the format selections in parallel with the text entry. This difficulty resulted in more errors and therefore lower text throughput. Touching the screen to perform a selection is slower than tilting the device or tapping the foot, but the action of moving the fingers off the keyboard to make a selection ensured high precision when interleaving selection and text entry. Additionally, mobile devices offer a breadth of promising rich input methods that need to be carefully studied in situ when deciding if each is appropriate to support a given task; it is not sufficient to study the modalities independent of a natural task.

KEYWORDS: Mobile device, multi-modal, text entry, text formatting, target selection, foot, tilt, touch, speech

INDEX TERMS: H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

1 INTRODUCTION

Text entry is a fundamental and common activity that users perform on their mobile devices. Although the mobile phone is most commonly used to send and receive simple unadorned text messages [7], users are looking for new ways to improve the expressivity their devices can afford. Rich text entry tasks such as writing a detailed and structured email, posting an update to a Blog, and editing a Word document are becoming increasingly more common. These rich text tasks are typically supported by selectable interface features that allow for faster (e.g., word completion), accurate (e.g., word correction), descriptive (e.g., font, format, colour) and structured (e.g., bullets, indentation) text



Figure 1. A user entering text on a mobile device and selecting the appropriate character level formatting.

entry. Selecting these features is currently accomplished by touching an on-screen widget, or navigating among a list of options using the directional pad.

Touching an on-screen widget or manipulating the directional pad are natural methods of selection, but they require users to interleave selection and typing, slowing the user's rate of text input. Given that text entry is already considered a bottleneck for expression on mobile devices, we wondered how well alternative input channels (e.g., accelerometers and speech recognition) might be used to efficiently complement text entry. These alternatives have the potential to be used in parallel with typing, allowing the user to dedicate her fingers to the primary typing task and reduce the impact that editing functions have on throughput.

The focus of this research is to better understand the efficiency potential that alternate input channels hold for increasing the expressivity and throughput of mobile text entry. Specifically, we are interested in comparing screen touching to alternate input channels as a means to support selection during text entry. The input channels we have chosen to evaluate are device tilt (*Tilt*), voice recognition (*Speech*), and foot tap (*Foot*). While there are a large number of alternate input channels that we could explore, we compared Touch with Tilt, Speech, and Foot because they cover a range of input technologies supported by devices today (noting that foot sensing is becoming more common place with the Nike + iPod Sports Kit [9]).

To better understand the relative tradeoffs that screen touch, device tilt, voice input and foot tapping offer users for performing multimodal edit-based selections during text entry, we conducted a controlled laboratory evaluation. Our goal was not to evaluate the technologies themselves, but rather their unique impacts on the flow of text input and formatting. Overall, we found:

- Touch resulted in the highest text throughput. Thus, our core hypothesis that parallel input channels would be faster was false. Coordinating selection and typing was difficult using tilt, foot and voice.
- A significant trade-off exists between selection speed and accuracy. Selection was quickest with device tilt and foot tapping, but screen touch and voice resulted in fewer errors.

dearman@dgp.toronto.edu
{karlson, brianme}@microsoft.com
bederson@cs.umd.edu

Graphics Interface Conference 2010
31 May - 2 June, Ottawa, Ontario, Canada
Copyright held by authors. Permission granted to
CHCCS/SCDHM to publish in print form, and ACM
to publish electronically.



- There are interesting human performance issues with respect to the orientation of a target within an input type. For example, tapping the ball of the foot is more accurate than using the heel.
- The time to select a target is slower than the time to resume typing the text.

2 RELATED WORK

The dominant modes for interacting with a mobile device currently require a user to touch the screen or use the directional pad. However, a mobile phone can support alternate input modalities such as accelerometers [3, 4, 10, 11, 15, 19, 20], speech recognition [14, 16, 17, 22], cameras [23] and chording [24] have been explored, some of which are already common in today's smart mobile devices.

Touching the screen with a finger or stylus, manipulating the directional pad and typing on a QWERTY or 12-button keypad are typical modes for interacting with mobile devices. The majority of phones provide one or more of these modes to interact with the information space. ChordTap by Wigdor *et al.*, based on the principles of a chorded keyboard, extends the default keypad by adding three buttons on the back of a mobile phone [24]. The ChordTap keys allow users to differentiate between the multiple characters for each button on a 12 button keypad.

Speech has been used as a means to provide direct input [22] and facilitate text correction [1, 5]. Jiang *et al.* fused word candidates generated by typing Chinese characters while speaking in parallel to generate a single reduced word set [5]. Similarly, Ao *et al.* corrected recognition errors in Chinese handwriting by having people verbally repeat the sentence [1].

Tilt and orientation of a mobile device have been used extensively to navigate through lists and menus [3, 4, 10, 11] and disambiguate between characters when typing [11, 15, 19, 20, 24]. Oakley *et al.* used tilt to navigate through one dimensional lists and menus [10, 11], invoking commands by rotating the device into one of three target regions along a 90 degree rotational space (vertical to horizontal). Unigesture, a tilt-to-write system by Sazawal *et al.*, enabled single handed text entry [20]. Rather than typing on a keypad, three to four characters are organized into seven regions that are selected using the orientation of the device. Partridge *et al.* refined the unigesture technique in TiltType [15], allowing users to disambiguate between the characters within a region by pressing one of four buttons. TiltText, a technique designed specifically for a mobile phone, utilized 30 and 60 degree rotations along the vertical and horizontal plane to disambiguate between the available characters on a standard 12-button mobile phone keypad [24]. Vision TiltText [23] mimicked the functionality of Wigdor *et al.*'s TiltText, but used the mobile phones built-in camera to differentiate between characters by detecting the user's hand movement, rather than the devices tilt.

Foot based input for a mobile device is generally a discounted mode of interaction. However, many professional examples confirm that feet can be elegantly engaged in tasks (*e.g.*, musicians, audio scribes). Pearson and Weiser explored alternate topologies of foot movement and the design space to support interacting with desktop computing [16]. They later implemented one such design called the planar slide mole, assessing its performance against a mouse for target selection [17]. Although the mouse was generally faster and less prone to errors, the foot was extremely quick in gesturing. Pakkanen *et al.* conducted a similar study utilizing a trackball to perform target selection, comparing the foot to the hand [14]. Their results highlight that although the foot is not as dexterous as the hand, it is appropriate for tasks that do not require precision or quick homing and

selection. The design of our study is in keeping with these recommendations. The foot is not required to perform homing, only selection.

Many alternate input modalities exist for mobile devices to support direct text input. These methods have never been directly compared in order to assess their effectiveness as a parallel input channel to enrich text entry. Understanding the performance of these alternate input modes will provide a better understanding of how multimodal selection and text entry can be integrated [12] and how to best support user's integration patterns [13].

3 STUDY

We performed two studies in order to explore human capabilities while performing selections during text entry. Four input modes were compared: standard screen touching (*Touch*), device tilting (*Tilt*), voice recognition (*Speech*), and foot tapping (*Foot*). Tilt, Speech and Foot all allow the user to keep her fingers on the keyboard, and so have the potential to be used independently from the act of letter selection. However, these channels also have characteristics that may impair the text entry task. For example, Speech is considered a "natural" form of input that does not require much in the way of physical effort, but verbalizing commands while typing words may impose additional cognitive demands. Tilt may be familiar to many modern mobile phone or game users, but coordinating Tilt with text entry may be difficult or uncomfortable in practice. Finally, while many professional examples confirm that feet can be elegantly engaged in tasks, Foot selection for the uninitiated user may simply be too awkward and too "distant" from the mental/manual focus of the task to be used effectively.

To evaluate user performance with our chosen input methods, we devised two experiments. Experiment one involved a stimulus-response *Target Selection* task to establish the speed and accuracy with which users can make target selections using each of our four chosen input modes. In experiment two, we used a *Text Formatting* task to evaluate how quickly and accurately users can apply text formatting using each input method while completing a text entry task. The purpose of studying the two tasks independently was to isolate systematic differences in users' abilities to perform selections using the different input methods (*Target Selection*) from other influences affecting the flow and throughput of text entry (*Text Formatting*). Initially we had envisioned using a more common word correction or prediction feature for this task. However, those tasks would not have allowed us to control for *when* the user applied the correction or prediction feature. The usefulness of correction and prediction features is dependent on the input text and the user's perception that selecting a word is faster than correcting or typing the word. Text Formatting was a realistic text entry task that allowed us to maintain control over when and where a selection is made.

The Touch, Tilt, Speech and Foot input methods vary greatly in the granularity of expression they support. For example, voice supports a large unconstrained input space, limited only by the human capacity to label and verbalize a selection. In contrast, researchers have formally characterized the limits of hand tilt to a much smaller input space [18]. Because our focus was on understanding the relative tradeoffs between the input methods during text entry and not comparing the expressive limits of each method, we chose a selection space of four options to achieve parity across the input methods. Limiting the selection space also allowed for straightforward visual mappings between the input gestures and on-screen selection targets.



3.1 Target Selection and Text Formatting Tasks

The *Target Selection* experiment involved a stimulus-response task designed to evaluate the speed and accuracy with which participants could identify and select on-screen targets in four different positions using the four input methods (Figure 2: top row). The goal of this task was to understand users' motor abilities across input methods. The target placement and alignment differed for each input method, but for each method the placement corresponded to the physical action necessary to perform a selection (Figure 2). Each trial began with a blank screen, requiring the participant to press the 'F' and 'J' keys simultaneously to display the target object, shown in red. The start posture was designed to ensure that the device was held in a consistent manner across trials and participants. Selection time was calculated from the time the 'F' and 'J' keys were pressed until a selection was made.

The *Text Formatting* experiment involved a modified text entry task that required participants to reproduce short text phrases that included visual formatting characteristics (Figure 2: bottom row). The goal of this task was to evaluate the speed and accuracy with which users could interleave the selection and de-selection of formatting states while concurrently entering text, and how the four input methods impact the primary text entry task. This is in contrast to the *Target Selection* task, which simply evaluated the user's ability to execute four distinct commands using each of the four input types.

Participants were required to enter the characters of a text phrase and apply formatting to the text at various positions in the phrase. The tasks allowed for partially formatted words, meaning that format mode activations were required both at the beginning and middle of a word, and format mode deactivations were required at both the middle and end of a word. In practice, the format commands were modal—only one format could be active at a time. For example, selecting 'Orange' would activate the orange text mode. All subsequent characters entered would be shown in orange until the user selected 'Orange' again; returning the text mode to "regular" entry mode.

The placement and alignment of the selectable format buttons was identical to that of the *Target Selection* task. The interface supported formatting sequences of three or more characters. Words of three to five characters could be formatted in whole, while words



Figure 3. Experimental setup for the Foot input condition.

To conduct this study we developed an application test-bed that provides input to a HTC Touch Pro 2 (shown in Figure 1) using screen Touch, Tilt of the device, Speech, and Foot tapping. The test-bed consisted of two components: a desktop computer running Windows Vista and the HTC Touch Pro 2 running Windows Mobile 6.1. The desktop and mobile device was connected wirelessly by a dedicated Linksys Wireless router using 802.11g.

3.2 Apparatus

Foot and Speech input was accomplished through the desktop computer by wirelessly communicating commands to the mobile device. Input for Foot was performed using two X-keys 3 switch foot pedals connected to the desktop computer via USB. One foot pedal was placed sideways under each foot such that the ball and heel of the foot depressed the respective left and right switch (Figure 3). In the default state, the switches were depressed by the pressure of the user's foot resting on them. A selection was registered when a switch was released, not pressed. This implementation allows for four possible inputs by lifting the ball or heel of the left and right foot. For example, lifting the heel of

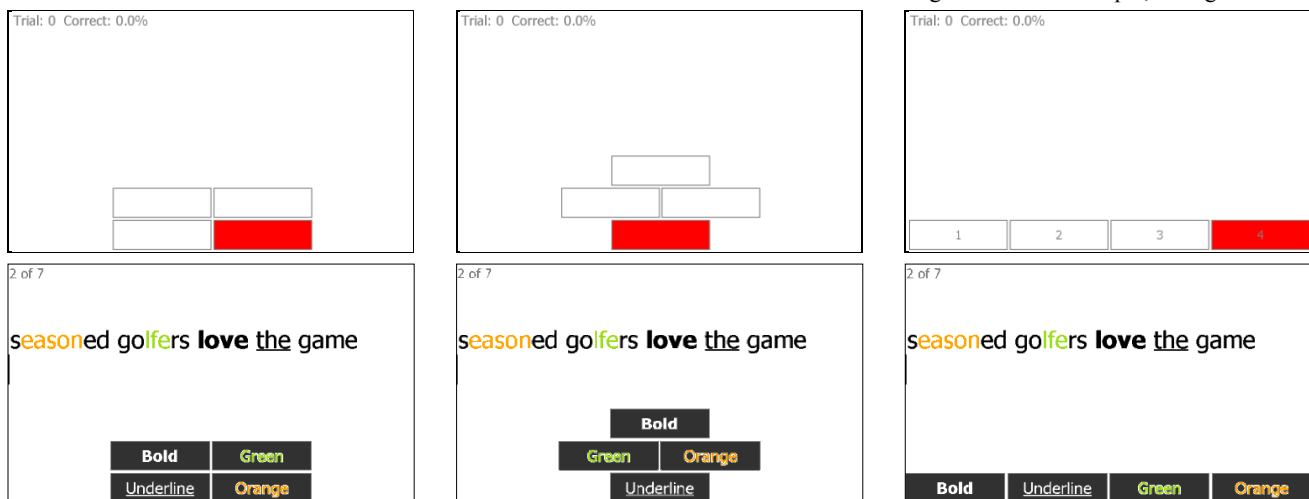


Figure 2. The software interface for the Target Selection (top) and Text Formatting (bottom) tasks. Presented are the Foot (left), Tilt (middle) and Touch/Voice (right) interfaces. The target to select for the Target Selection task is red. The phrase used in the Text Formatting task includes all four formats: 'eason' is orange; 'lfe' is green; 'love' is bold; and 'the' is underlined.



the right foot would select the red target (Figure 2: top-left) or 'Bold' format mode (Figure 2: bottom-left).

Input with the Speech condition was performed by saying the target's label. The speech recognition component of our test bed was implemented using a Wizard of Oz simulation. We chose not use computer-based speech recognition because the systems we tested for the desktop and mobile device incurred a noticeable lag between the time when a label was spoken and interpreted. We wanted the latency between saying a label and it being selected to be as small as possible to allow for a fair comparison. To accomplish this, we relied on a human wizard to listen to the participants' verbal selection and press one of four corresponding keys on a keyboard connected to the desktop computer. The selection was then wirelessly communicated to the mobile device. For example, saying "four" would result in the selection of the red target (Figure 2: top-right). Similarly, saying "Bold" would select the 'Bold' target and enter 'Bold' input mode (Figure 2: bottom-right).

Tilt and Touch inputs were implemented directly on the mobile device and did not require the desktop computer. Input using the Tilt of the mobile device was implemented by sampling the integrated six degree of freedom accelerometer at 25 Hz. We interpreted four Tilt gestures: tilting the device forward, backward, left and right. Gestures exceeding 30 degree changes from a continually updated "neutral" position were recognized as a command along the direction of movement. We chose to implement a relative rather than absolute origin because it allowed users to choose a comfortable position and angle at which to hold the device. For example, tilting the device backward would select the red target (Figure 2: top-middle), or the 'Underline' formatting state (Figure 2: bottom-middle).

Input using Touch was performed by physically pressing the appropriate target displayed on the device's resistive screen. For example, pressing the red target would select it (Figure 2: top-right). Similarly, pressing the 'Bold' target would activate the 'Bold' formatting state (Figure 2: bottom-right).

3.3 Participants

Twenty-four participants recruited from the general population took part in the study – 11 females and 13 males. The age of participants varied between 18 and 38, with a median age of 26. We recruited participants that currently owned a mobile device (e.g., Blackberry, HTC Touch, iPhone or iPod Touch) that they currently use to enter text on a daily basis and have done so consistently for at least the past four months. All participants owned a device with either a physical or touch screen based QWERTY keyboard. All but one participant were right handed. Participants were compensated (removed for review) for their time.

3.4 Procedure

The *Target Selection* experiment was administered first. The participant was first introduced to all four input methods and trained how to use each. The number of training trials varied across participants, but always continued until both the participant and experimenter felt comfortable with the participant's performance. The participant then completed the selection task for each of the four input methods, completing all trials for a given input method before continuing to the next method. Participants were instructed to make the selections as quickly and accurately as possible.

For the *Text Formatting* experiment, the participant was first asked to read a document describing the formatting task and then enter 10 to 12 training phrases for each input method. If the participant felt uncomfortable after the training tasks for any input

mode, she was allowed to continue training until both the participant and experimenter felt comfortable with her performance. After training on all input methods, the testing phase began, during which the participant completed four sets of test phrases grouped by input type. Participants were instructed to enter and format the text as quickly and accurately as possible, and to correct all mistakes. However, perfect input was not enforced. After completing trials for each input type, a modified NASA TLX survey was administered. After the final input method was completed, a concluding survey was administered asking participants to rank the inputs in order of preference and to provide subjective feedback with respect to the least and most preferred input.

3.5 Design

The order Tilt, Touch, Speech and Foot were presented was fully counterbalanced across the twenty-four participants for the *Target Selection* and *Text Formatting* experiments. The *Target Selection* experiment was a 4×4 design. It comprised the following factors and levels:

- *Input Type* {Touch, Tilt, Speech, Foot}
- *Target Position* {1, 2, 3, 4}

Each *Input Type* was evaluated over six blocks of trials (1 training; 5 testing) with 20 test trials per block - five trials for each of the four *Target Positions*. Each participant performed $4 \times 5 \times 4 \times 5 = 400$ test trials or 9,600 among all 24 participants. The presentation order of the target positions was randomly assigned within each block, but consistent across participants. The first block for each *Input Type* was training and excluded from the analysis.

The *Text Formatting* experiment was a 4×3×4 design. It comprised the following factors and levels:

- *Input Type* {Touch, Tilt, Speech, Foot}
- *Format Position* {Start, Middle, End}
- *Target Position* {1, 2, 3, 4}

Each *Input Type* was evaluated over five blocks of trials (one training and four testing) with between 8 and 12 phrases per block. Each block required 48 format selections – four trials for each *Format Position* × *Target Position*. Participants performed $4 \times 4 \times 3 \times 4 \times 4 = 768$ format selection and entered 3,111 characters or 18,432 format selections and 74,664 characters among all 24 participants. The length of the words and the number of words per phrase dictated the overall number of phrases required to meet the 48 format selections per block. The presentation order of the format position and the type of format to be applied was randomly assigned within blocks, but presented consistently across participants. The first block for each *Input Type* was training and excluded from the analysis.

4 RESULTS

The *Target Selection* and *Text Formatting* experiments were conducted independently, and thus analyzed separately. Selection time trials that exceeded three standard deviations from the mean were removed as outliers. To account for the variability in human selection, the median selection time for each participant was used in the analysis. Timing data was analyzed with repeated measures ANOVAs using Wilk's Lambda. Event-count measures such as error were analyzed with nonparametric Friedman tests and post-hoc pairwise comparisons were conducted with the Wilcoxon test. All post-hoc comparisons were conducted using Holm's sequential Bonferroni correction.



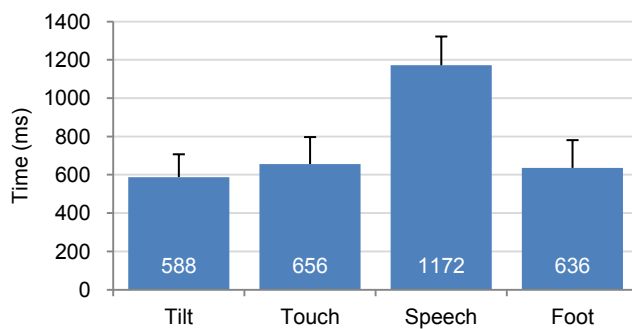


Figure 4. The average selection time for the Target Selection experiment grouped by the Input Types. The error bars indicate the standard deviation.

In addition to evaluating the participants' performance for each *Input Type*, we evaluated how each input impacted the task of text entry by analyzing the participant's character stream with Wobbrock and Myers TextTest StreamAnalyzer [25].

We do not directly compare the *Target Positions* across the four *Input Types* because the positions vary and are therefore not equivalent. Rather, we focus our analysis on the selection time and error between *Target Positions* within *Input Types*.

4.1 Target Selection Results

Seventy-six outliers (0.8%)—three standard deviations from the mean—were excluded from the analysis. Analysis of the blocks yielded no evidence of a learning effect; therefore all test trials are included in the time and error analyses.

4.1.1 Target Selection Time

Repeated measures ANOVA on the median selection times yielded a significant effect of *Input Type*, $F_{3, 21}=879.98$, $p<0.001$. Post-hoc pairwise comparisons show that the overall selection time for Speech (1172 ms) was slower than the other *Input Types* (Figure 4), all at $p<0.001$. Although Touch (656 ms) and Foot (635 ms) have similar mean selection times, Tilt (588 ms) was found to be faster than the other *Input Types*, all at $p<0.001$.

Repeated measures ANOVAs for *Target Positions* within *Input Types* found a significant effect on selection time for Tilt ($F_{3, 21}=8.64$, $p<0.001$), Touch ($F_{3, 21}=13.80$, $p<0.001$) and Speech ($F_{3, 21}=11.10$, $p<0.001$), but not for Foot.

Tilt. Tilting the device forward (561 ms) was faster than tilting in the other directions: forward was 8.2% faster than left (594 ms; $p<0.05$), 5.9% faster than right (591 ms; $p<0.001$), and 5.3% faster than backward (607 ms; $p<0.01$).

Touch. The second target (680 ms; ordered left to right) was 4.0% slower than the first (654 ms; $p<0.001$), and 7.1% slower than the fourth (635ms; $p<0.001$). While somewhat surprising, the slower selection time for the second target is likely attributed to the majority of our participants being right-hand dominant. This agrees with our observation that many participants opted to use their right thumb to reach across the screen to the second target rather than using their proximally closer left thumb.

Speech. The fourth target (1199ms; ordered left to right) was 3.5% slower than the first (1158ms; $p<0.001$), and 3.9% slower than the second (1154ms; $p<0.001$). While the slower selection time of the fourth target might be attributed to a tendency for participants to scan left to right, this explanation seems somewhat suspect since we would have expected target recognition to occur preattentively and be immune to position bias. Furthermore, it is useful to remember that the Speech condition involved two independent human response components (participant and

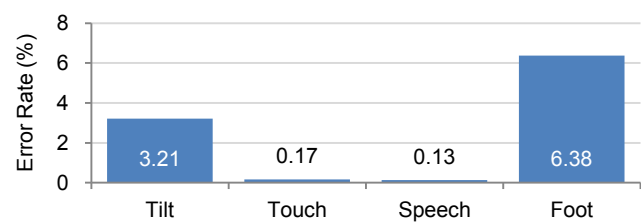


Figure 5. The selection error rate for the Target Selection experiment grouped by the Input Types.

wizard). Together with the fact that overall differences in selection times across positions was very small (≤ 41 ms) it is unlikely that the position effect has practical meaning with respect to user performance for Speech.

4.1.2 Target Selection Errors

The overall selection error rate was 2.47%. A Friedman test showed a significant main effect of *Input Type*, $\chi^2_{(3, N=24)} = 55.29$, $p<0.001$. The error rates for Touch (0.17%) and Speech (0.13%) are negligible ($<1\%$) and significantly lower than Tilt (3.21%) and Foot (6.38%), all with $p<0.001$. In addition, the error rate for Foot is greater than Tilt, $p<0.005$.

Pairwise analysis of the *Target Positions* within the *Input Types* yielded significant differences in the error rate for Tilt ($\chi^2_{(3, N=24)} = 7.88$, $p<0.05$) and Foot ($\chi^2_{(3, N=24)} = 8.52$, $p<0.05$). Post-hoc pairwise comparisons were conducted with the Wilcoxon test.

Tilt. Forward tilt ($n=29$; 1.21%) resulted in a higher error rate than backwards ($n=10$; 0.04%), $p<0.005$. No differences were found for left ($n=18$; 0.75%) and right ($n=20$; 0.83%).

Foot. The error rate for the left-heel ($n=52$; 2.17%) was greater than the right-ball ($n=27$; 1.13%), $p<0.005$. No differences were found for the left-ball ($n=32$; 1.33%) or right-heel ($n=42$; 1.75%). We also compared the combined error rate of the ball and heel of the left and right foot. Overall, the heel ($n=94$; 3.92%) has an error rate greater than the ball of the foot ($n=59$; 2.46%), $p<0.05$. Since most of our participants were right-footed, it makes sense that participants would be most agile with the ball of their dominant foot (right) and least agile with the heel of their non-dominant foot (left).

4.2 Text Formatting Results

Two-hundred and ten (1.1%) selections were identified as outliers (greater than three standard deviations from the mean) and removed from the analysis. The timing data for P1 using Touch (192 entries) was not included in the analysis because of a software logging error. In the analysis we differentiate:

- *Selection Time* – the time difference between typing a character and selecting a subsequent formatting.
- *Resumption Time* – the time difference between selecting a format and typing a subsequent character.

Pairwise comparison shows that selection time is slower, $p<0.001$, than resumption time (Figures 6 and 7).

4.2.1 Format Selection Time

Repeated measures ANOVA of the median selection time yielded a significant effect of *Input Type*, $F_{3, 20} = 95.23$, $p<0.001$, and *Format Position*, $F_{2, 21} = 15.0$, $p<0.001$. Similar to the *Target Selection* results, post-hoc pairwise comparisons show that the selection time for Speech (1146 ms) was slower than the other *Input Types* (Figure 6), all at $p<0.001$. Although Touch (855 ms), Tilt (797 ms) and Foot (834 ms) have similar mean selection times, Tilt was found to be faster than Touch, $p<0.001$. Analysis



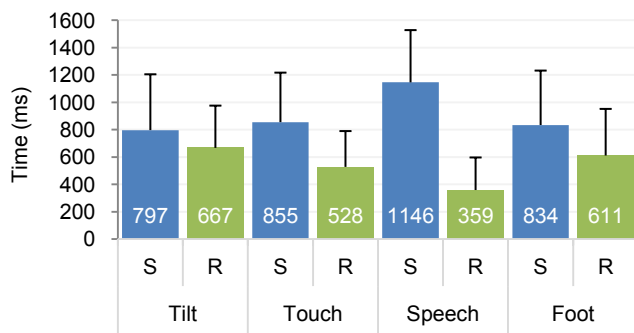


Figure 6. The average selection time (S) and resumption time (R) for the Text Formatting experiment grouped by the Input Types. The error bars indicate the standard deviation.

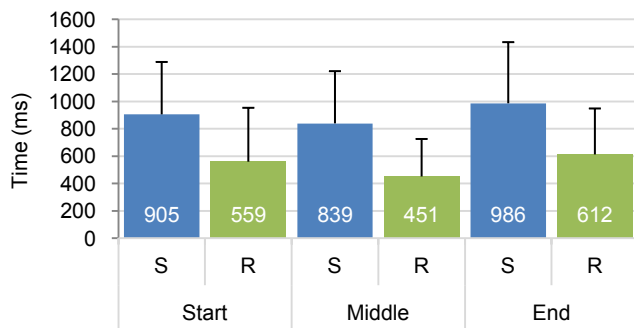


Figure 7. The average selection time (S) and resumption time (R) for the Text Formatting experiment grouped by the Format Positions. The error bars indicate the standard deviation.

of the *Format Position* revealed that toggling a format selection at the End of a word (839 ms) is faster than the Start (905ms; $p<0.01$) and Middle of a word (985 ms; $p<0.001$).

Repeated measures ANOVAs for *Target Position* within *Input Types* found a significant main effect on selection time for Touch ($F_{3, 19}=11.04$, $p<0.001$), Speech ($F_{3, 19}=11.62$, $p<0.001$), and Foot ($F_{3, 18}=7.30$, $p<0.005$), but not Tilt.

Touch. The second target (920 ms) is 13.6% slower than the third (810ms) and 12.1% slower than the fourth (821 ms), all at $p<0.001$, which matches our findings from the Target Selection study.

Speech. The second target (1192 ms) is slower than all the other target positions: 5.7% slower than the first (1128 ms; $p<0.005$); 7.6% slower than the third (1108 ms; $p<0.001$); and 3.3% slower than the fourth (1154 ms; $p<0.05$).

Foot. The left-heel (903 ms) is 14.0% slower than the right-ball (792 ms; $p<0.005$) and 11.3% slower than the right-heel (811 ms; $p<0.001$), but not the left-ball (834 ms).

4.2.2 Format Resumption Time

Repeated measures ANOVAs on the median resumption times yielded a significant effect of *Input Type*, $F_{3, 20}=22.27$, $p<0.001$, and *Format Position*, $F_{2, 21}=80.10$, $p<0.001$. In contrast to the selection time results where Speech was the slowest *Input Type*, Speech (359 ms) was the fastest *Input Type* for resumption, all at $p<0.001$. Touch (528 ms), Tilt (667 ms) and Foot (611 ms) have comparable mean resumption times, but Touch was faster than Tilt, $p<0.001$. Analysis of the *Format Position* revealed that toggling a format at the End of a word (451 ms) is faster than the Start (559 ms) and Middle of a word (611 ms), all at $p<0.001$. In addition, toggling a format at the Start of a word is faster than the Middle, $p<0.001$.

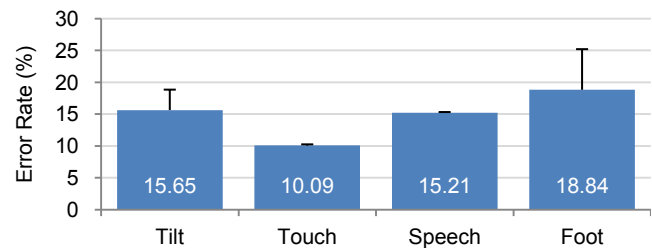


Figure 8. The selection error rate for the Text Formatting experiment grouped by the Input Types.

Repeated measures ANOVAs for *Target Position* within *Input Types* found a significant main effect of resumption time for Touch ($F_{3, 19}=7.89$, $p<0.001$), but not Tilt, Speech or Foot.

Touch. The first target (502 ms) was faster than the other target positions: 5.2% faster than the second (529 ms; $p<0.05$); 8.0% faster than the third (543 ms; $p<0.001$); and 7.0% faster than the fourth (538 ms; $p<0.005$).

4.2.3 Format Selection Errors

The overall error rate was 14.95%. A Friedman test showed a significant main effect of *Input Type*, $\chi^2_{(3, N=23)}=27.05$, $p<0.001$, but not the *Format Positions*. Post-hoc pairwise comparisons showed that Touch incurs fewer errors than the other *Input Types*, all at $p<0.001$.

Analyses of *Target Position* within *Input Types* yielded significant differences in error rate for Speech ($\chi^2_{(3, N=23)}=10.08$, $p<0.005$) and Foot ($\chi^2_{(3, N=24)}=8.52$, $p<0.05$), but not Touch and Tilt. Post-hoc pairwise comparisons were conducted with the Wilcoxon test.

Speech. The error rate for the second target ($n=154$; 3.34%) was lower than the third target ($n=206$; 4.47%), $p<0.005$, but not the first ($n=171$; 3.71%) or fourth ($n=170$; 3.69%).

Foot. The error rate for the left-ball ($n=262$; 5.69%) was greater than the right-ball ($n=207$; 4.49%), $p<0.001$, and right-heel ($n=185$; 4.01%), $p<0.001$, but not the left-heel ($n=214$; 4.64%).

4.2.4 Text Throughput (Characters per Second)

Across all participants, the average character per second (CPS) text throughput was 1.36 (see Table 1). The CPS reported here (1.36) is much lower than the mini-QWERTY CPS (2.65) reported in prior research [2] because of the additional formatting requirements. There is a significant main effect of *Input Type* on CPS, $F_{3, 21}=19.06$, $p<0.001$. Post-hoc pairwise comparison of the CPS shows that Touch resulted in a 9.8% greater throughput than Tilt, $p<0.001$ and 10.7% greater throughput than Foot, $p<0.001$.

4.2.5 Format Errors and Text Corrections

Formatting the phrases correctly required participants to select the four *Target Positions* a total of 4,608 times for each *Input Type*.

Table 1. For each Input Type, the difference between the number of formats required and the actual number of formats used (FEC), backspace count (BS), characters per second (CPS), uncorrected (UER) and corrected error rate (CER). The CPS, UER and CER are calculated using Wobbrock and Myers StreamAnalyzer [25].

	FEC (N)	BS (N)	CPS (N/s)	UER (N/s)	CER (N/s)
Tilt	839	1062	1.32	0.0055	0.0522
Touch	219	1048	1.45	0.0033	0.0506
Speech	184	1619	1.37	0.0037	0.0770
Foot	1320	1451	1.31	0.0019	0.0702



Table 1 shows the total number of formatting errors (FEC), the total number of backspaces (BS), and the uncorrected (UER) and corrected (CER) error rates. Note that UER and CER only capture character level errors, and not format-level errors. Each measure gives insight into how *Input Type* impacted the overall text throughput. We observed a significant main effect of *Input Type* for FEC, $\chi^2_{(3, N=24)}=57.85, p<0.001$, backspaces, $\chi^2_{(3, N=24)}=20.53, p<0.001$ and CER, $\chi^2_{(3, N=24)}=25.65, p<0.001$, but not UER.

Post-hoc pairwise comparisons show that the Foot has a greater FEC than Tilt, $p<0.05$, Touch, $p<0.001$, and Speech, $p<0.001$; and that Tilt has a greater FEC than Touch and Speech, both $p<0.001$. In contrast, the number of backspaces used with Tilt and Touch is lower than Speech (both $p<0.001$) and Foot ($p<0.05$ and $p<0.01$ respectively). Similarly, the CER for Tilt and Touch is lower than Foot and Speech, both $p<0.001$. The higher error rate of Foot and Tilt was likely due to in-place formatting mistakes. The higher correction rate of Speech and Foot suggest that users made more errors placing the formats relative to the text. This is what we would expect if users had trouble coordinating the more asynchronous inputs of Speech and Foot entry with text input.

4.3 Qualitative Responses

The results of the modified NASA TLX surveys and the ranked preferences of the *Input Types* are shown in Figure 9. Participants rated each *Input Type* using the NASA TLX measures on a 7-point Likert scale (1=‘Very Low’; 7=‘Very High’). The subjective measures were: mental demand (*Mental*), physical demand (*Physical*), task success (*Success*), speed of use (*Speed*), ease of use (*Ease*) and ease of learnability (*Learning*).

The participants responses revealed a significant difference in their ranked preference for the *Input Types*, $\chi^2_{(3, N=24)} = 14.9, p<0.005$. Post-hoc pairwise comparison showed a preference for Touch over Foot, $p<0.001$. The majority of participants ranked Touch (11) and Speech (9) as their preferred input, but a small number of participants showed a strong preference for Tilt (3) and Foot (1).

Friedman tests of the NASA TLX measures showed a main effect of *Input Type* for: mental demand, $\chi^2_{(3, N=24)}=22.31, p<0.001$; physical demand, $\chi^2_{(3, N=24)}=30.16, p<0.001$; task success, $\chi^2_{(3, N=24)}=17.31, p<0.001$; speed of use, $\chi^2_{(3, N=24)}=13.48, p<0.005$; ease of use, $\chi^2_{(3, N=24)}=20.00, p<0.001$; and ease of learnability, $\chi^2_{(3, N=24)}=19.26, p<0.001$. Touch and Speech were perceived as more successful than Tilt (both $p<0.005$) and Foot ($p<0.005$ and $p<0.05$ respectively), and as easier to use than Tilt ($p<0.005$ and $p<0.01$) and Foot ($p<0.001$ and $p<0.005$). Speech was easier to learn than Foot, $p<0.005$, and Foot was more mentally and physically demanding to use than Tilt ($p<0.005$ and $p<0.05$), Touch (both $p<0.001$), and Speech ($p<0.005$ and $p<0.001$). Although Tilt was no more mentally demanding than the other *Input Types*, it was more physically demanding to use than Touch, $p<0.01$ and Speech, $p<0.001$. Overall, participants correctly felt that they were faster with Touch than Foot, $p<0.005$.

5 DISCUSSION

Examination of selection speed, independent of the text input, revealed that selecting targets was quickest using Tilt, rather than Touch, Speech and Foot. However, with respect to the text formatting task, participants experienced the greatest text throughput (characters per second entry) using Touch and Speech – the two slowest *Input Types* for performing selections. While these results are seemingly at odds with one another, the impacted text throughput for Tilt and Foot is partially due to their higher error rate. Tilt and Foot allowed for the fastest selections, but resulted in the greatest number of errors. In contrast, Touch and Speech

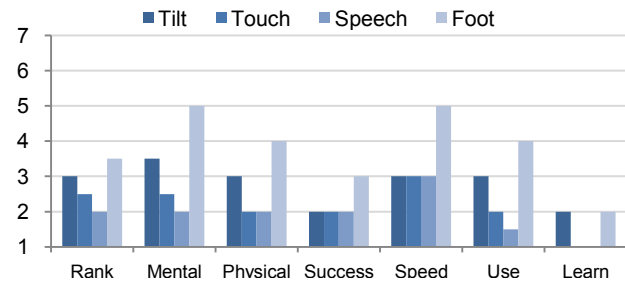


Figure 9. The participants responses (median values) to their ranked preference and the modified NASA TLX survey for each input type. Lower values are better.

resulted in the slowest selection times, but the lowest number of errors. The time required to correct the erroneous selections and any improperly formatted characters accumulated, impacting the overall text throughput for Tilt and Foot. Although we trained participants until both the participant and experimenter felt comfortable with the participant’s performance, some participants commented that Tilt and Foot were “*awkward to coordinate effectively*.” The common sentiment among participants was that Touch and Speech were “*the most natural*”, making them easier to use and less mentally/physically demanding.

It was our core hypothesis for the Text Formatting experiment that Tilt, Speech and Foot would result in the highest test throughput. We believed that the ability to make format selections and enter text in parallel would improve overall text throughput. However, contrary to our initial expectations, Touch resulted in the highest text throughput. The sequential ordering of selection and text entry allowed Touch to strike a balance between speed and a low rate of error. For Speech and Foot, it was difficult to coordinate selection in parallel with typing, resulting in a higher per character error rate than Touch: “*I tend to format a split second before I start typing ... I’d type a letter or two before the formatting would take place ... I would find myself typing the word that was supposed to be green ... before saying green.*” While Tilt and Foot “*seemed easy to use because you could keep your fingers on the keyboard*” and Touch “*felt like it was slowing me down having to move my thumbs up [to select targets]*”, it was in fact more difficult to make an accurate selection with Tilt and Foot, resulting in a higher format error rate than Touch. Moving the thumbs off the keyboard to make a selection, although perceived as slowing down the text entry, ensured participants did not have to coordinate parallel selection and typing. Depending on the granularity required for a text application, the concurrency problem with Tilt, Speech and Foot can be corrected through simple modifications to the interface. Rather than requiring a format selection to be activated at a precise position within the word (e.g., beginning), the format could be activated by the user at any point during word entry and applied to the entire word. If more precision is needed, the format could be applied “back in time” to the characters that were typed when, for example, the utterance was started, rather than after it was completed and recognized.

The majority of participants preferred Touch and Speech, but a small minority preferred Tilt and Foot. We believe that the perceptions of the Foot modality may have been negatively impacted by our use of foot pedals to register the selections. One participant who did not rank Foot as their preferred input commented that, “*I think I performed best with foot, but I’m on a cell phone, why would I use my feet? If it was integrated into my shoe, feet would be number one.*” We explained to participants in the concluding interview that a more realistic implementation would



integrate within actual footwear such as the Nike + iPod Sport Kit [9]. Sensing integrated within a user's own footwear may perform better and be perceived as more useful.

Alternate target placements for each input type highlighted interesting design considerations with respect to the target orientations and input. Handedness and footedness is important when choosing target placement. The left to right ordering for touch resulted in the second button being slower than the third and fourth. It was most frequently the case that participants would use their right thumb to select all buttons except the first one: "*I kept trying to figure out which thumb to use [to select the second button]. It's not like my left thumb was tired, but I just kept crossing all the way over with my right thumb*" Selection using the heel resulted in more errors than the ball of the foot, "*it's easier to trigger with my toes than my heels.*"

6 CONCLUSION

Mobile devices support many rich text entry tasks that require a user to interleave typing and selection to produce the text and formatting she desires. For example, text completion, correction and formatting functions improve the speed of text entry and enrich the presentation of text, but currently require a user perform a selection (via screen touch or the directional pad) while typing. In this paper we performed two complementary studies to explore the performance and limitations of using standard screen touch (*Touch*), device tilt (*Tilt*), voice recognition (*Speech*), and foot tap (*Foot*) to perform selections in support of text entry. The results show that Tilt is fastest for selecting a target, but that selection with Touch while entering text allows for the greatest character per second text throughput. Coordinating selection with Tilt, Speech and Foot in parallel with typing proved to be difficult, resulting in a higher per character error rate than Touch. Although perceived as slow—and slower than Tilt for selecting targets—moving the thumbs off the keyboard to make a selection ensured a natural interleaving of selection and typing; producing a low rate of error, resulting in the need for fewer corrections producing the highest text throughput.

Additionally, we highlighted human performance issues with respect to the orientation of the targets within an input type for handedness and footedness. The left to right ordering for Touch resulted in the second button being slower than the third and fourth because participants would use their right thumb to reach across and select all except the first button. Similarly, tapping the ball of the foot is more accurate than using the heel.

7 REFERENCES

- [1] Ao, X., Wang, X., Tian, F., Dai, G. and Wang, H. Crossmodal error correction of continuous handwriting recognition by speech. In *Proc. IUI 2007*, ACM Press (2007), 243-250.
- [2] Clarkson, E., Clawson, J., Lyons, K. and Starmer, T. An empirical study of typing rates on mini-QWERTY keyboards. *Ext. Abstracts CHI 2005*, ACM Press (2005), 1288-1291.
- [3] Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C. and Want, R. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In *Proc. CHI 1998*, ACM Press (1998), 17-24.
- [4] Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing techniques for mobile interaction. In *Proc. USIT 2000*, ACM Press (2000), 91-100.
- [5] Jiang, Y., Wang, X., Tian, F., Ao, X., Dai, G. and Wang, H. Multimodal Chinese text entry with speech and keypad on mobile devices. In *Proc. IUI 2008*, ACM Press (2008), 341-344.
- [6] Kume, Y. Foot interface: fantastic phantom slipper. *Ext. Abstracts SIGGRAPH 1998*, ACM Press (1998), 114.
- [7] MacKenzie, I.S. and Soukoreff, R.W. Phrase sets for evaluating text entry techniques. *Ext. Abstracts CHI 2003*, ACM Press (2003), 754-755.
- [8] Nielsen. Flying Fingers: Text-messaging overtakes monthly phone calls, http://en-us.nielsen.com/main/insights/consumer_insight/issue_12/flying_fingers
- [9] Nike + iPod Sport Kit, <http://www.apple.com/ipod/nike/>
- [10] Oakley, I. and O'Modhrain, S. Tilt to Scroll: Evaluating a motion based vibrotactile mobile interface. In *Proc. WHC 2005*, IEEE Computer Society (2005), 40-49.
- [11] Oakley, I. and Park, J. A motion-based marking menu system. *Ext. Abstracts CHI 2007*, ACM Press (2007), 2597-2602.
- [12] Oviatt, S. Ten myths of multimodal interaction. *Commun. ACM* 42, 11 (1999), 74-81.
- [13] Oviatt, S., Lunsford, R. and Coulston, R. Individual differences in multimodal integration patterns: what are they and why do they exist? In *Proc. CHI 2005*, ACM Press (2005), 241-249.
- [14] Pakkanen, T. and Raisamo, R. Appropriateness of foot interaction for non-accurate spatial tasks. *Ext. Abstracts CHI 2004*, ACM Press (2004), 1123-1126.
- [15] Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G. and Want, R. TiltType: accelerometer-supported text entry for very small devices. In *Proc. UIST 2002*, ACM Press (2002), 201-204.
- [16] Pearson, G. and Weiser, M. Of moles and men: the design of foot controls for workstations. In *Proc. CHI 1986*, ACM Press (1986), 333-339.
- [17] Pearson, G. and Weiser, M. Exploratory evaluation of a planar foot-operated cursor-positioning device. In *Proc. CHI 1988*, ACM Press (1988), 13-18.
- [18] Rahman, M., Gustafson, S., Irani, P. and Subramanian, S. Tilt techniques: investigating the dexterity of wrist-based input. In *Proc. CHI 2009*, ACM Press (2009), 1943-1952.
- [19] Rekimoto, J. Tilting operations for small screen interfaces. In *Proc. UIST 1996*, ACM Press (1996), 167-168.
- [20] Sazawal, V., Want, R. and Borriello, G. The unigesture approach. In *Proc. Mobile HCI 2002*, Springer-Verlag (2002), 256-270.
- [21] Schoning, J., Daiber, F., Kruger, A. and Rohs, M. Using hands and feet to navigate and manipulate spatial data. *Ext. Abstracts CHI 2009*, ACM Press (2009), 4663-4668.
- [22] Voice command for Windows Mobile, <http://www.microsoft.com/windowsmobile/en-us/downloads/microsoft/about-voice-command.mspx>
- [23] Wang, J., Zhai, S. and Canny, J. Camera phone based motion sensing: interaction techniques, applications and performance study. In *Proc. UIST 2006*, ACM Press (2006), 101-110.
- [24] Wigdor, D. and Balakrishnan, R. A comparison of consecutive and concurrent input text entry techniques for mobile phones. In *Proc. CHI 2004*, ACM Press (2004), 81-88.
- [25] Wobbrock, J.O. and Myers, B.A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (Dec. 2006), 458-489.



APPENDIX 5

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

ACM Transactions on Computer-Human Interaction, Vol. 11, No. 1, March 2004, Pages 90–119.

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.

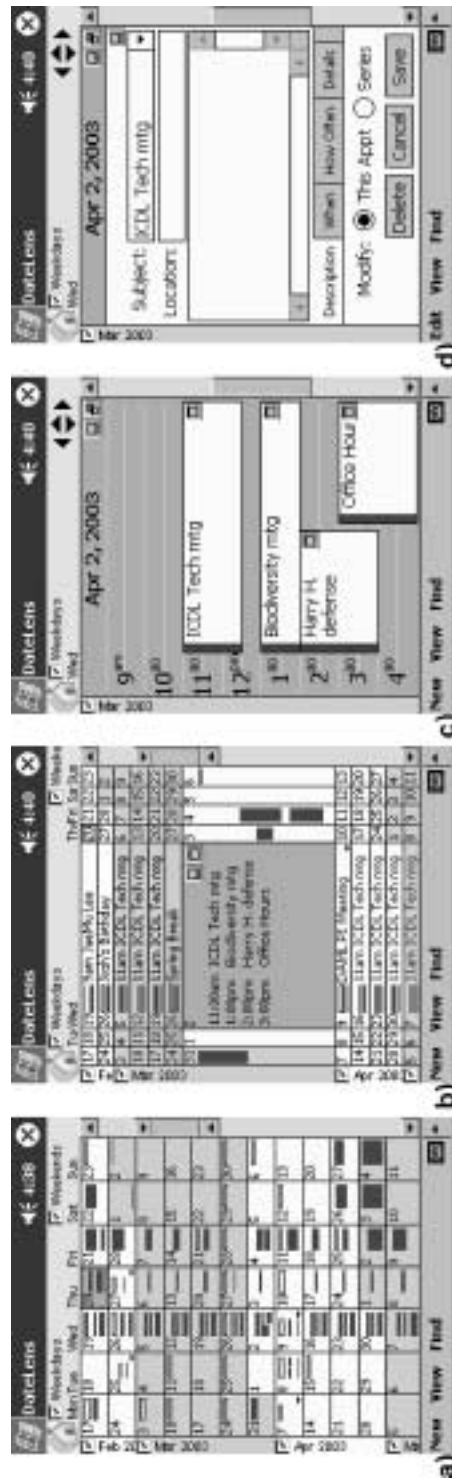


Fig. 1. The DateLens interface with the view configured to show 12 weeks and consecutive levels of detail as the view is: a) shown as an overview; b) zoomed into one day; c) focused on that day; and d) zoomed into an appointment. All transitions between views are animated.

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

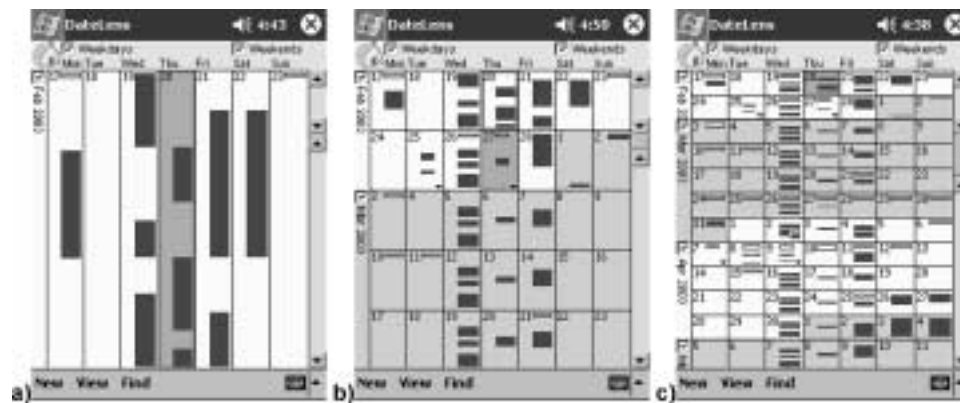


Fig. 2. A series of views as the bottom of the scrollbar thumb is dragged downwards to show: a) 1 week; b) 1 month; c) and 3 months.



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

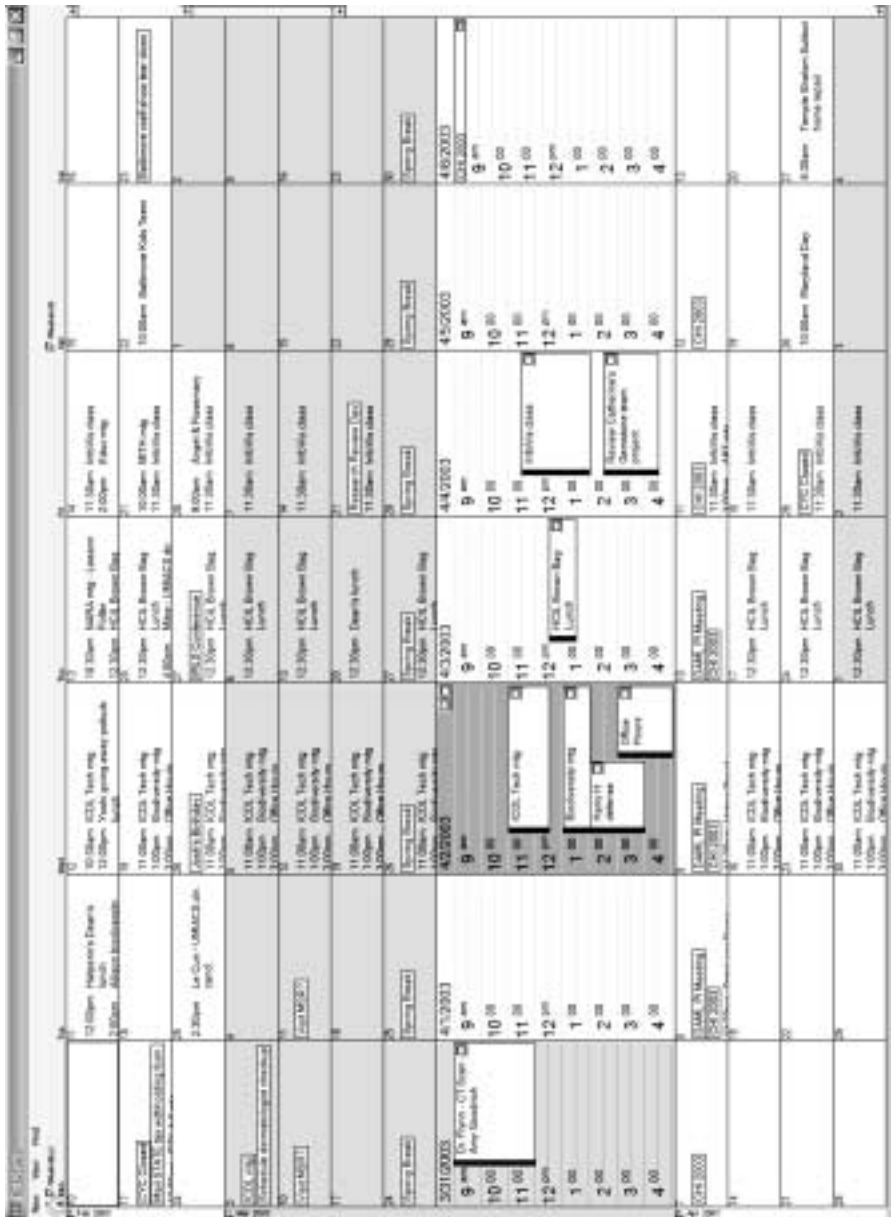


Fig. 4. DateLens scaled up to 1400 × 1050 pixels on a desktop computer. There is room to display an entire week with the full day representation, even while showing three months of the calendar.

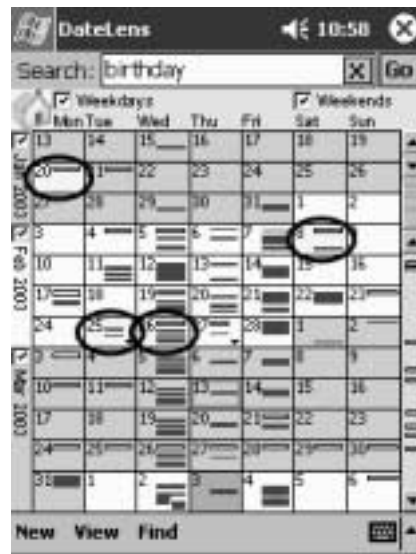


Fig. 5. DateLens showing the results of searching for birthdays (colored highlights are circled for black-and-white printing clarity). A few individual birthdays are highlighted. In addition, the scrollbar shows several days in the future that are highlighted, which also have birthdays.

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.

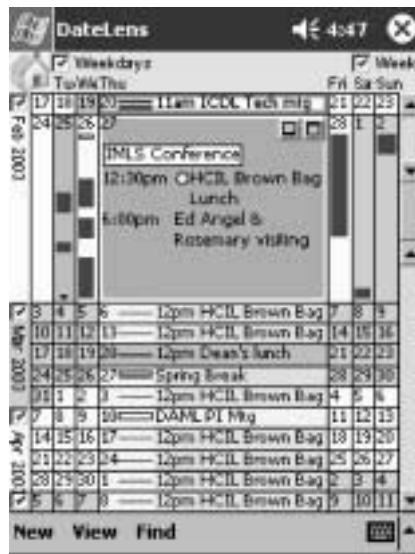


Fig. 6. The results of clicking on the appointment “HCIL Brown Bag Lunch”. Most Thursdays are highlighted in yellow which shows a recurring appointment. In addition, several days are highlighted in orange which match either “HCIL”, “Brown”, “Bag”, or “Lunch”.

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

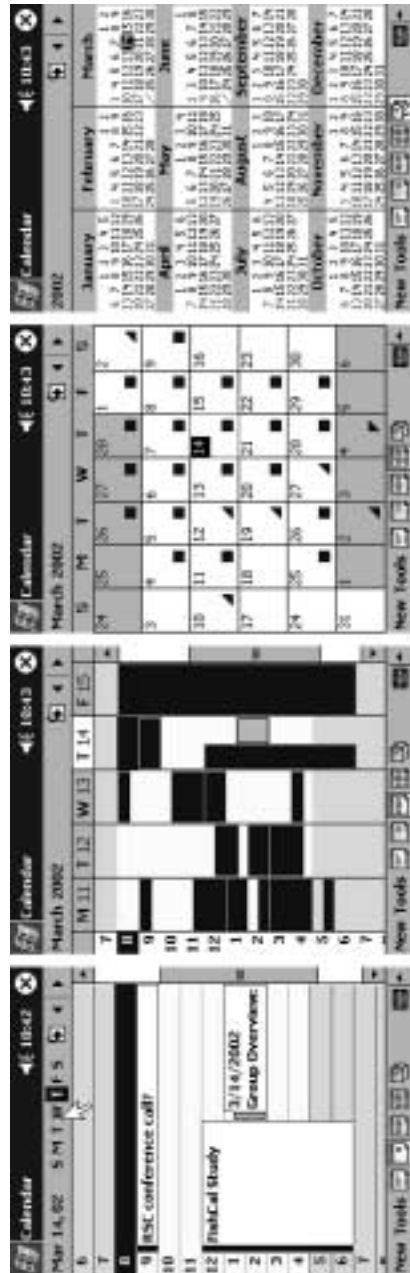


Fig. 7. Screen shots of the Microsoft Pocket PC Calendar program that was used in the study showing day, week, month, and year views.

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

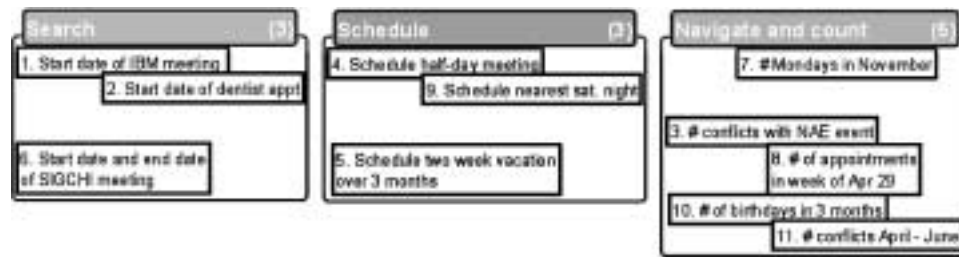


Fig. 8. Summary of tasks grouped by type and complexity. In each group, the top tasks are the “simple” ones, and the bottom ones are the “complex” ones.

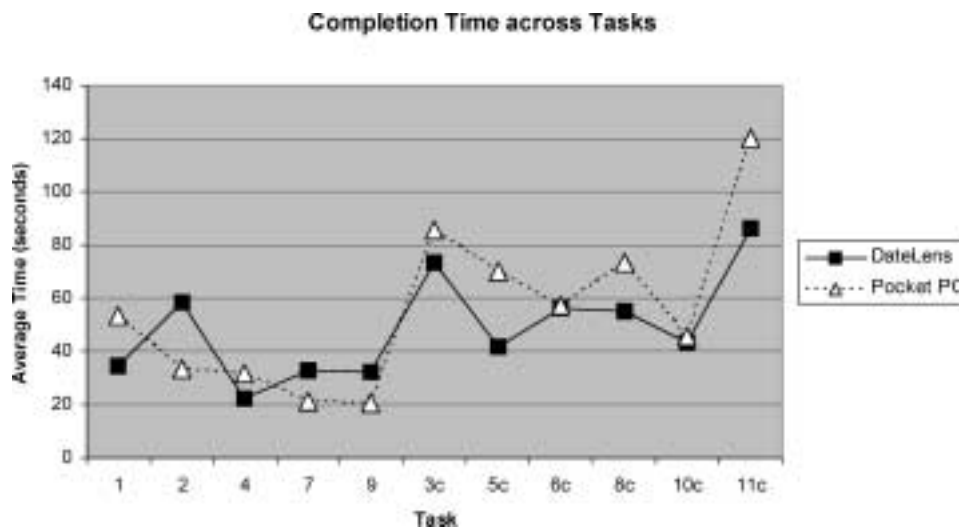


Fig. 9. The time spent by study participants to complete each task using the DataLens and standard Pocket PC calendar interfaces. Tasks are listed with simple tasks on the left, and complex (marked with a ‘c’ in the task number) on the right.

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

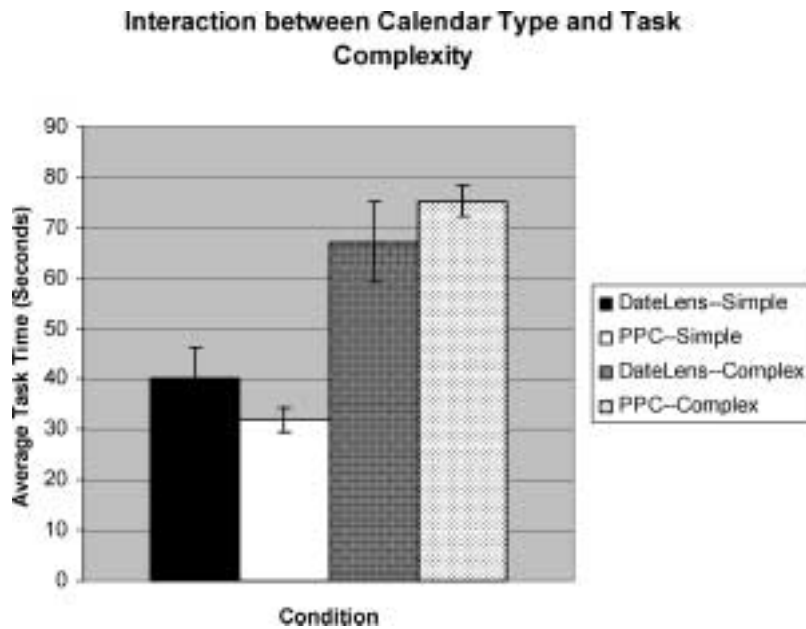


Fig. 10. Effects of calendar type by task complexity on average task times.

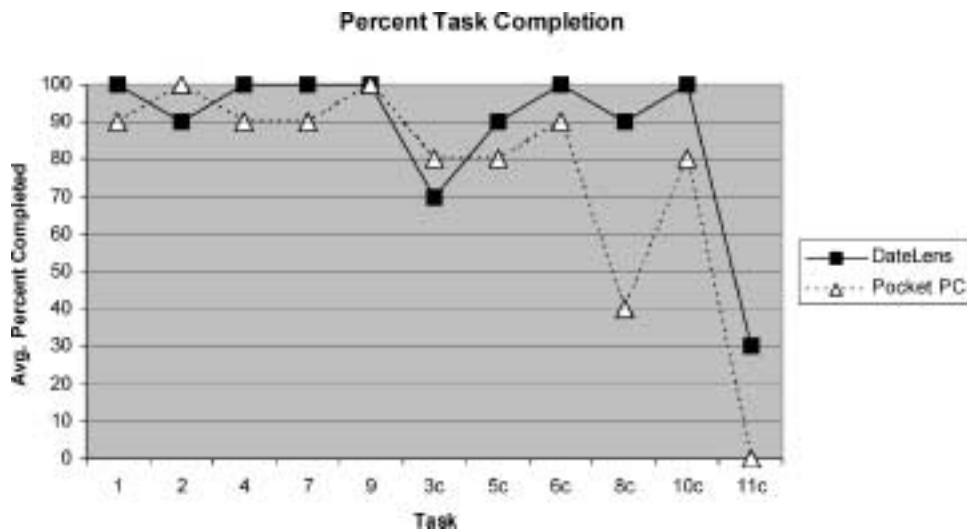


Fig. 11. The percent of tasks completed by study participants for each task. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

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

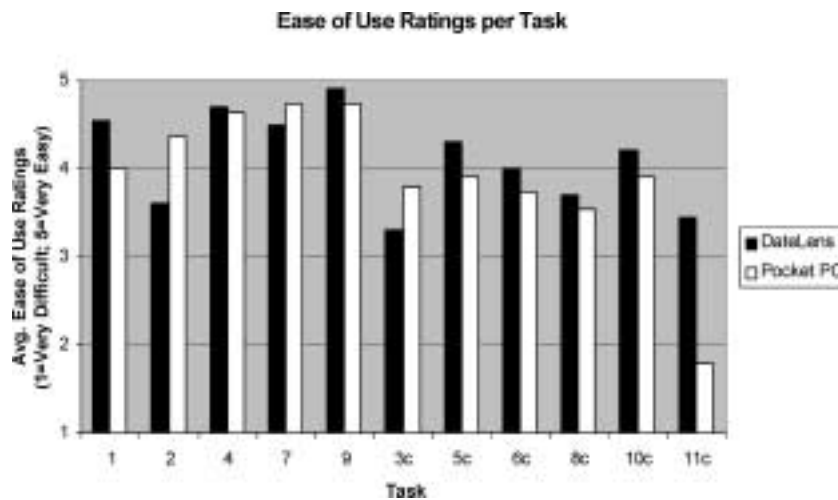


Fig. 12. The ease of use rating for each task by study participants. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

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.

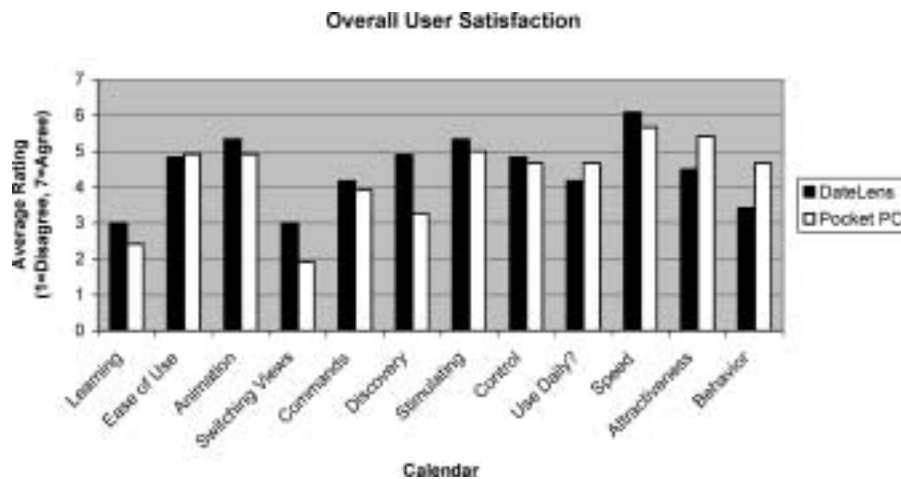


Fig. 13. Overall user satisfaction.

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

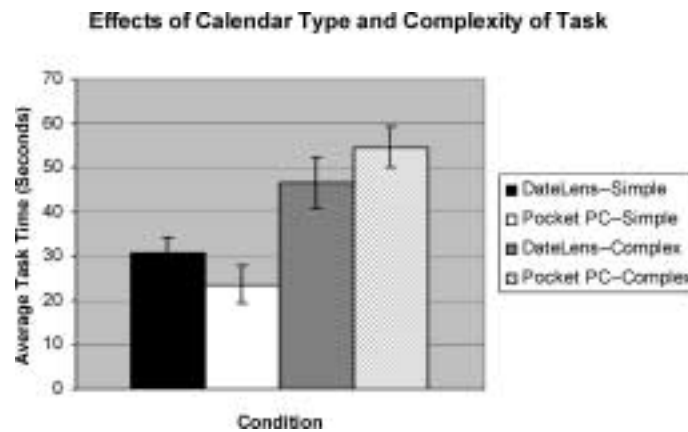


Fig. 15. Effects of calendar type and task complexity on overall average task times in study 2.

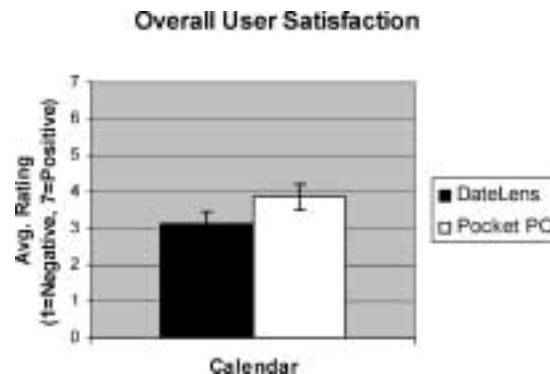


Fig. 16. Overall average satisfaction for each calendar based on the experimental tasks.

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.

ACM Transactions on Computer-Human Interaction, Vol. 11, No. 1, March 2004.

- 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

APPENDIX 6

Understanding One Handed Use of Mobile Devices

Amy K. Karlson
Human-Computer Interaction Lab
Department of Computer Science
A.V. Williams Building
University of Maryland, College Park, MD 20742
Phone: (301) 405-2662 Fax: (301) 405-6707
akk@cs.umd.edu

Benjamin B. Bederson
Human-Computer Interaction Lab
Department of Computer Science
A.V. Williams Building
University of Maryland, College Park, MD 20742
Phone: (301) 405-2764 Fax: (301) 405-6707
bederson@cs.umd.edu

Jose L. Contreras-Vidal
Cognitive-Motor Behavior Laboratory
Department of Kinesiology
Health and Human Performance Building
University of Maryland, College Park, MD 20742
Phone: (301) 405-2495 Fax: (301) 405-6707
pepeum@umd.edu

Karlson, A., Bederson, B. B. (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.

Understanding One Handed Use of Mobile Devices

ABSTRACT

Mobile phones are poised to be the world's most pervasive technology, already outnumbering land lines, personal computers and even people in some counties! Unfortunately, solutions to address the usability challenges of using devices on the move have not progressed as quickly as the technology or user distribution. Our work specifically considers situations in which a mobile user may have only a single hand available to operate a device. To both motivate and offer recommendations for one-handed mobile design, we have conducted three foundational studies: a field study to capture how users currently operate devices; a survey to record user preference for the number of hands used for a variety of mobile tasks, and an empirical evaluation to understand how device size, interaction location, and movement direction influence thumb agility. In this chapter we describe these studies, their results, and implications for mobile device design.

INTRODUCTION

The handheld market is growing at a tremendous rate; the technology is advancing rapidly and experts project that annual mobile phone sales will top 1 billion by 2009 (Pittet et al., 2005). To meet customer demand for portability and style, device manufacturers continually introduce smaller, sleeker profiles to the market. Yet advances in battery power, processing speed and memory allow these devices to come equipped with increasing numbers of functions, features, and applications. Unfortunately these divergent trends are at direct odds with usability: richer content accessed via shrinking input and output channels simply makes devices harder to use. The unique requirements for mobile computing only compound the problem, since mobile use scenarios can involve to unstable environments, eyes-free interaction, competition for attention resources, and varying hand availability (Pascoe, Ryan, & Mores, 2000). While each of these constraints requires attention in design, we are currently interested in issues of usability when a user only has only one hand available to operate a mobile device.

Devices that accommodate single-handed interaction can offer a significant benefit to users by freeing a hand for the host of physical and attentional demands common to mobile activities. But there is little evidence that current devices are designed with this goal in mind. Small, light phones that are easy to control with one hand are unfriendly to thumbs due to small buttons and crowded keypads. Larger devices are not only harder to manage with a single hand, they tend to feature more (rather than larger) buttons, as well as stylus-based touchscreens whose rich interface designs maximize information content, but offer targets too small, and/or too distant, for effective thumb interaction.

While it may seem obvious which features inhibit single-handed use, there has been relatively little systematic study of enabling technologies and interaction techniques. Most commercial and research efforts in one-handed device interaction have focused primarily on either a specific technology or task. For example, accelerometers have been explored to support tilt as a general input channel for handheld devices (Dong, Watters, & Duffy, 2005; Hinckley, Pierce, Sinclair, & Horvitz, 2000; Rekimoto, 1996), while media control (Apple, 2006; Pirhonen, Brewster, & Holguin, 2002) and text entry (Wigdor & Balakrishnan, 2003) have been popular tasks to

consider for one-handed device operation. But in the varied landscape of mobile devices and applications, one-handed design solutions must ultimately extend to a wide range of forms and functions. We began our investigation of this problem by looking at the fundamental human factors involved in operating a device with a single hand.

In this chapter, we report on three studies conducted to understand different aspects of one-handed mobile design requirements. We first ran a field study to capture the extent to which single-handed use is currently showing up “in the wild”. Second, we polled users directly to record personal accounts of current and preferred device usage patterns. The results from these studies help motivate one-handed interface research, and offer insight into the devices and tasks for which one-handed techniques would be most welcomed. Finally, we performed an empirical evaluation of thumb tap speed to understand how device size, target location, and movement direction influence performance. From these results we suggest hardware-independent design guidelines for the placement of interaction objects. Together our findings offer foundational knowledge in user behavior, preference, and motor movement for future research in single-handed mobile design.

BACKGROUND

The physical and attentional demands of mobile device use was perhaps first reported for fieldworkers (Kristoffersen & Ljungberg, 1999; Pascoe, Ryan, & Mores, 2000), from which design recommendations for minimal-attention and one-handed touchscreen interface designs emerged (Pascoe et al., 2000). Though well suited to the directed tasks of fieldwork, the guidelines do not generalize to the varied and complex personal information management tasks of today’s average user. Research of the effects that mobility has on attention and user performance continues (Oulasvirta, Tamminen, Roto, & Kuorelahti, 2005), as well as how these factors can be replicated for laboratory study (Barnard, Yi, Jacko, & Sears, 2005).

Several approaches for one-handed device interaction have been proposed. Limited gestures sets have been explored for mobile application control with both the thumb (Apple, 2006; Karlson, Bederson, & SanGiovanni, 2005; Pascoe et al., 2000) and index finger (Pirhonen et al., 2002), but none have specifically considered ergonomic factors. Since text entry remains the input bottleneck for mobile devices, many are working on improvements, and some targeting one-handed use. Peripheral keyboards for one-handed text entry are available, such as the Twiddler (Lyons et al., 2004), but the mobile device itself must be supported by another hand, desk or lap, which violates our definition of one-handed device control. Text entry on phone keypads is generally performed with a single thumb, but methods to improve input efficiency have focused on reducing the number of key presses required, such as T9 word prediction, rather than by improving ergonomics by optimizing button sizes, locations, or movement trajectories. Accelerometer-augmented devices allow for the device’s spatial orientation to serve as an input channel, and have been shown to support one-handed panning (Dong, Watters, & Duffy, 2005), scrolling (Rekimoto, 1996), and text entry (Wigdor & Balakrishnan, 2003). However, the coarse level of control tilt offers, and the potential for confusion with the normal movements of mobile computing necessarily limit the viability of tilt for generalized input.

Scientists in the medical community have studied the biomechanics of the thumb extensively for the purposes of both reconstruction and rehabilitation. The structure of the thumb is well

understood (Barmakian, 1992), but only now are scientists beginning to reliably quantify the functional capabilities of the thumb. Strength has been the traditional parameter used to assess biomechanical capabilities, and recent research has established the effect movement direction has on thumb strength (Li & Harkness, 2004). Unfortunately, only standard anatomical planes have been considered, which excludes movements toward the palm that are typical of mobile device interaction. As a complement to force capabilities, others have looked at range as a characteristic of thumb movement. Kuo, Cooley, Kaufman, Su and An (2004) have developed a model for the maximal 3D workspace of the thumb and Hirota (2003) has quantified an average angle for thumb rotation. The experimental conditions for these studies, however, do not account for constraints imposed by holding objects of varying size, such as alternative models of handheld device.

FIELD STUDY

One motivation for our research in single-handed mobile designs was our assumption that people already use devices in this manner. Since current interaction patterns, whether by preference or necessity, are predictive of future behavior, they are likely to be transferred to new devices. This suggests that designs should become more accommodating to single-handed use, rather than less, as the current trend seems to be. To capture current behavior, we conducted an *in situ* study of user interaction with mobile devices. The study targeted an airport environment for the high potential of finding mobile device users and ease of access for unobtrusive observation.

Field Study Method

We observed 50 travelers (27 male) at Baltimore Washington International Airport's main ticketing terminal during a six hour period during peak holiday travel. Because observation was limited to areas accessible to non-ticketed passengers, seating options were scarce. We expected to observe the use of both Personal Digital Assistants (PDAs) and cell phones since travelers are likely to be coordinating transportation, catching up on work, and using mobile devices for entertainment purposes. Since most users talk on the phone with one hand, we recorded only the cell phone interactions that included keypad interaction as well. All observations were performed anonymously without any interaction with the observed.

Note that while any subject observation without consent presents a legitimate question for ethical debate, in our research we follow the federal policy on the protection of human research subjects (Department of Health and Human Services, 2005) as a guideline. The policy states that the observation of public behavior is not regulated if the anonymity of the subjects is maintained and that disclosure of the observations would not put the subjects at risk in terms of civil liability, financial standing, employability, or reputation. Since we were interested in capturing natural behavior, did not record identifying characteristics, and consider phone use while standing, walking and sitting relatively safe activities, we did not obtain subject consent.

Field Study Measures

For each user observed, we recorded sex, approximate age, and device type used: candy bar phone, flip phone, Blackberry, or PDA. A "candy bar" phone is the industry term for a traditional-style cellular phone with a rigid rectangular form, typically about 3 times longer than

wide. For phone use, we recorded the hand(s) used to dial (left, right or both) and the hand(s) used to speak (left, right or both). We also noted whether users were carrying additional items, and their current activity (selected from the mutually exclusive categories: walking, standing, or sitting).

Field Study Results

Only two users were observed operating devices other than mobile phones - one used a PDA and the other a Blackberry. Both were seated and using two hands. The remainder of the discussion focuses on the 48 phone users (62.5% flip, 37.5% candy bar). Overall, 74% used one hand for keypad interaction. By activity, 65% of one handed users had a hand occupied, 54% were walking, 35% were standing, and 11% were sitting. Figure 1 presents the distribution of subjects who used one vs. two hands for keypad interaction, categorized by the activity they were engaged in (walking, standing, or sitting). The distribution of users engaged in the three activities reflects the airport scenario where many more people were walking or standing than sitting. It is plain from Figure 1 that the relative proportion of one handed to two handed phone users varied by activity; the vast majority of walkers used one hand, about two-thirds of standers used one hand, but seated participants tended to use two hands. However, we also recorded whether one hand was occupied during the activity, and found walkers were more likely to have one hand occupied (60%), followed by standers (50%), and finally sitters (25%), which may be the true reason walkers were more likely than standers to use one hand, as well as why standers were more likely than sitters to use one hand. Regardless of activity, when both hands were available for use, the percentage of one vs. two handed phone users was equal.

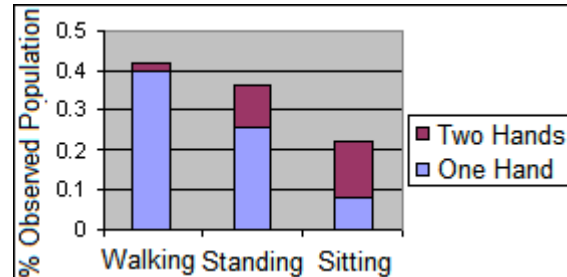


Figure 1. Airport Field Study - number of hands used for keypad interaction by activity.

Analysis of Field Study

Although Figure 1 suggests a relationship between user activity and keypad interaction behavior, it is unclear whether activity influences the number of hands used, or vice versa. Furthermore, since the percentage of users with one hand occupied correlates with the distribution of one-handed use across activities, hand availability, rather than preference, may be the more influential factor in the number of hands used to interact with the keypad. While use scenario certainly impacts usage patterns, the fact that users were as likely to use one hand as two hands when both hands were available suggests that preference, habit and personal comfort also play a role. Regardless of scenario, we can safely conclude that one-handed phone use is quite common, and thus is an essential consideration in mobile phone design.

Generalizability. The choice of observation location may have biased our results from those found in the general population since travelers may be more likely to be: 1) carrying additional items; 2) standing or walking; and 3) using a phone vs. PDA. Different environments, information domains, populations, and scenarios will yield unique usage patterns. Our goal was not to catalogue each possible combination, but to learn what we could from a typical in-transit scenario.

WEB SURVEY

While informative for a preliminary exploration, shortcomings of the field study were a) a lack of knowledge about motivation for usage style; b) the limited types of devices observed (phones); and c) the limited tasks types observed (assumed dialing). To broaden our understanding of device use over these dimensions, we designed a survey to capture user perceptions of, preferences for and motivations surrounding their own device usage patterns.

Survey Method

The survey consisted of 18 questions presented on a single web page which was accessed via an encrypted connection (SSL) from a computer science department server. An introductory message informed potential participants of the goals of the survey and assured anonymity. Notification that results would be posted for public access after the survey period was over provided the only incentive for participation. Participants were solicited from a voluntary subscription mailing list about the activities of our laboratory. In addition the solicitation was propagated to one recipient's personal mailing list, a medical informatics mailing list, and a link to the survey was posted on two undergraduate CS course web pages.

Survey Measures

For each participant, we collected age, sex and occupation demographics. Users recorded all styles of phones and/or PDAs owned, but were asked to complete the survey with only one device in mind - the one used for the majority of information management tasks. We collected general information about the primary device, including usage frequency, input hardware, and method of text entry. We then asked a variety of questions to understand when and why people use one vs. two hands to operate a device. We asked users to record the number of hands used (one and/or two) for eighteen typical mobile tasks, and then to specify the number of hands (one *or* two) they would *prefer* to use for each task. Three pairs of activities were designed to distinguish between usage patterns for different tasks within the same application, which we differentiated as “read” (email reading, calendar lookup, and contact lookup) vs. “write” (email writing, calendar entry, and contact entry) tasks. Users then recorded the number of hands used for the majority of device interaction and under what circumstances they chose one option over the other. Finally, users were asked how many hands they would prefer to use for the majority of interactions (including no preference), and were also asked to record additional comments.

Survey Results

Two hundred twenty-nine participants (135 male) responded to the survey solicitation. One male participant was eliminated from the remaining analysis because his handheld device was specialized for audio play only, leaving 228. Median participant age was 38.5 years. Participant

occupations reflected the channels for solicitation, with 25% in CS, IT or engineering, 23% students of unstated discipline, 20% in the medical field, 10% in education, and the remainder (21%) from other professional disciplines.

Devices owned. The three most common devices owned were flip phones (52%), small candy bar phones (23%) and Palm devices without a Qwerty keyboard (20%). Palm devices with an integrated Qwerty keyboard were as common as Pocket PCs without a keyboard (14%). Since interaction behavior may depend on device input capabilities, we reclassified each user's primary device into one of four general categories based on the device's input channels: (i) *keypad-only* (51%) are devices with a 12-key numeric keypad but no touchscreen, (ii) *TS-no-qwerty* (23%) are devices with a touchscreen but no Qwerty keyboard, (iii) *TS-with-qwerty* (21%) are devices with a touchscreen as well as an integrated Qwerty keyboard, and finally (iv) *qwerty-only* (5%) are devices with an integrated Qwerty keyboard but no touchscreen. For users with multiple devices, we derived their primary device type from the text entry method reported.

Current usage patterns. Of the 18 activities users typically perform with devices, 9 were performed more often with one hand, 6 more often with two hands, and 3 were performed nearly as often with one vs. two hands. Figure 2a displays these results, with the shaded backgrounds grouping the activities preferred with one, either or two hands. Upon inspection, all of the "reading" activities were performed more often with one hand (top) and all "writing" activities with two hands (bottom). Considering users' device types, we notice that with the exception of gaming, owners of *keypad-only* devices were more likely to use one hand regardless of activity, owners of *TS-no-qwerty* were more likely to use two-hands for most activities, and those owning Qwerty based devices were more likely to use two hands when performing writing tasks, but not reading tasks.

Overall, 45% of participants stated they use one hand for nearly all device interactions, as opposed to only 19% who responded similarly for two hands. Considering device ownership, however, users of touchscreen-based devices were more likely to use two hands "always" than they were one hand (Figure 3). When participants use one hand, the majority (61%) perceive they do so whenever the interface supports it, the reason cited by only 10% of those who use two hands. Device form dictated usage behavior when the device was too small for two hands, too large for one hand, or when large devices could be supported by a surface and used with one hand. Participants cited task type as a reason for hand choice, primarily as a trade off between efficiency and resources usage: 14% of users selected one hand only for simple tasks (conserving resources), while 5% selected two hands for entering text, gaming, or otherwise for improving the speed of interaction (favoring efficiency). Finally, according to respondents, the majority of two-handed use occurs when it is the only way to accomplish the task given the interface (63%).

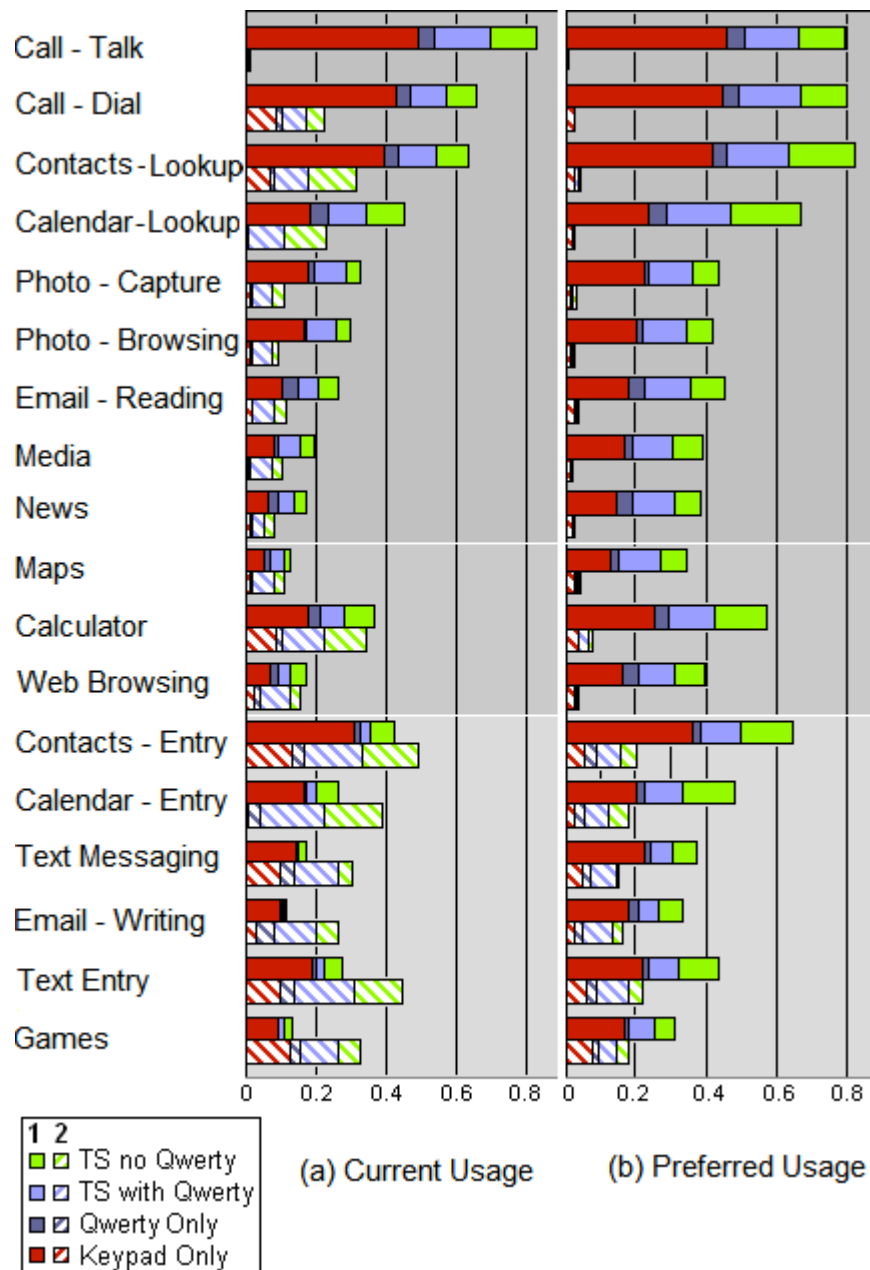


Figure 2. Web Survey – Number of hands (a) currently used and (b) preferred (1 hand is shown as solid, 2 hands is shown as striped) for 18 mobile tasks as a percentage of the observed population. Hand usage for each task is broken down by device type (TS = touchscreen).

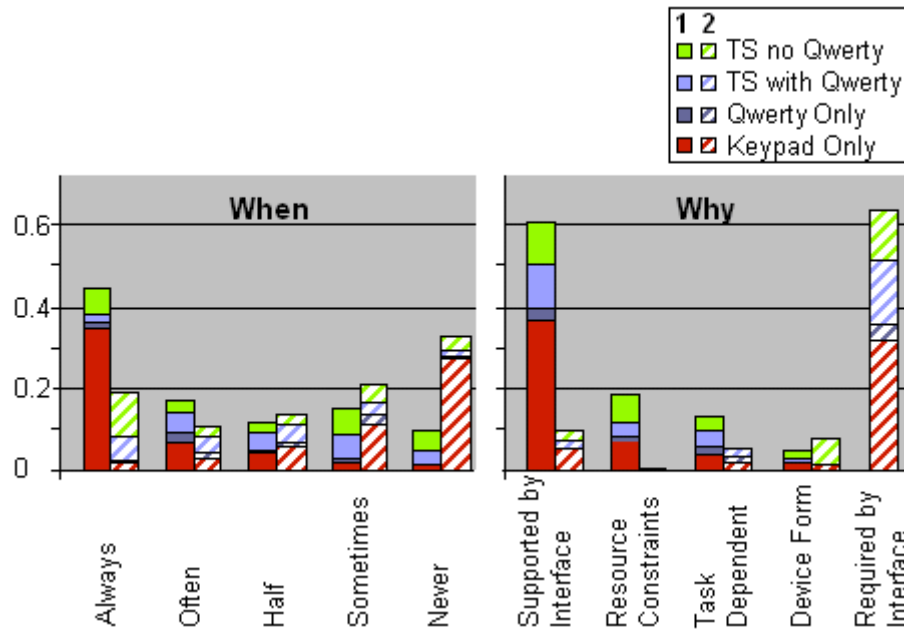


Figure 3. Web Survey - The frequency and reasons for one (solid) and two (striped) handed device use, broken down by device type.

Hand preferences. When asked how many hands users preferred to use while performing the same 18 tasks, one hand was preferred overwhelmingly to two hands for all tasks (Figure 2b). The activities with the closest margin between the number of participants who preferred one vs. two hands were playing games (13%) and composing email (16%). With one exception (gaming), the activities for which more than 14% of users stated a preference for two hands were “writing” tasks (e.g., those that required text entry): text entry, contact entry, calendar entry, email writing, and text messaging, in decreasing order. Even so, except for users of *TS-with-qwerty* devices, the majority of users stated a preference for using one hand, regardless of task or device owned. Users of *TS-with-qwerty* devices preferred two hands for text messaging, email composition, and text entry. Based on these data, it is consistent that 66% of participants stated they would prefer to use one hand for the majority of device interaction, versus 9% who would prefer two hands for all interaction. Twenty-three percent did not have a preference and 6 users did not respond.

Survey Summary

Considering current usage patterns only, there is no obvious winner between one and two handed use. Excluding phone calls, the number of activities for which a majority of respondents use one (7) vs. two hands (6) is nearly balanced. However, device type certainly influences user behavior; users of keypad-only devices nearly always use one hand, while users of touchscreen devices more often favor two hands, especially for tasks involving text entry. But user justifications for hand choice indicate that the hardware/software interface is to blame for much two-handed use occurring today. Most use one hand if at all possible and only use two hands when the interface makes a task impossible to do otherwise. Other than gaming, tasks involving text entry are the only ones for which users may be willing to use two hands, especially when the device used provides an integrated Qwerty keyboard. It seems, therefore, that the efficiency

gained by using two hands for such tasks is often worth the dedication of physical resources, which is also true of the immersive gaming experience.

While most users can imagine the ideal of single-handed text entry, enabling single-handed input may not be enough - *throughput* is also important. Ultimately, it is clear that interface designers of all device types should make one-handed usability a priority, and strive to bridge the gap between current and desired usage patterns.

THUMB MOVEMENT STUDY

The third component of our exploration was an examination of thumb movement in the context of mobile device interaction. As input technologies and device forms come and go, biomechanical limitations of the thumb will remain. Although the thumb is a highly versatile appendage with an impressive range of motion, it is most adapted for grasping tasks, playing opposite the other four fingers (Bourbonnais, Forget, Carrier, & Lepage, 1993). Hence thumb interaction on the surface of today's mobile devices introduces novel movement and exertion requirements for the thumb – repetitive pressing tasks issued on a plane parallel to the palm. We believe a fundamental understanding of thumb capabilities when holding a device can help guide the placement of interaction targets for both hardware and software interfaces designed for one-handed use. Although we can make reasonable guesses about thumb capabilities, empirical evidence is a better guide. Since no strictly relevant studies have yet been conducted, we have developed a study to help us understand how device form and task influences thumb mobility

Since thumb tapping is the predominant means of interaction for keypad-based devices, and has also proven promising for one-handed touchscreen use (Karlson et al., 2005), we focused our investigation on surface tapping tasks. We hypothesized that the difficulty of a tapping task would depend on device size, movement direction, and surface location of the interaction. We captured the impact of these factors on user performance by using movement speed as a proxy for task difficulty – under the assumption that harder tasks would be performed more slowly than easier tasks.

Equipment

Device models. For real devices, design elements such as buttons and screens communicate to the user the “valid” input areas of the device. We instead wanted outcomes of task performance to *suggest* appropriate surface areas for thumb interaction. We identified four common handheld devices to represent the range of sizes and shapes found in the market today: (1) a Siemens S56 candy bar phone measuring 4.0 x 1.7 x 0.6 in (10.2 x 4.3 x 1.5 cm); (2) a Samsung SCH-i600 flip phone measuring 3.5 x 2.1 x 0.9 in (9 x 5.4 x 2.3 cm); (3) an iMate smartphone measuring 4 x 2.0 x 0.9 in (10.2 x 5.1 x 2.3 cm) and (4) an HP iPAQ h4155 Pocket PC measuring 4.5 x 2.8 x 0.5 in (11.4 x 7.1 x 1.3 cm). These devices are shown in the top row of Figure 4. We refer to these as simply SMALL, FLIP, LARGE, and PDA. To remove the bias inherent in existing devices, we created a 3D model of each device, removing all superficial design features. The models were developed using Z Corp.'s (<http://www.zcorp.com/>) ZPrinter 310 3D rapid prototyping system. Device models were hollow, but we reintroduced weight to provide a realistic feel. Once “printed” and cured, the models were sanded and sealed to achieve a smooth finish.

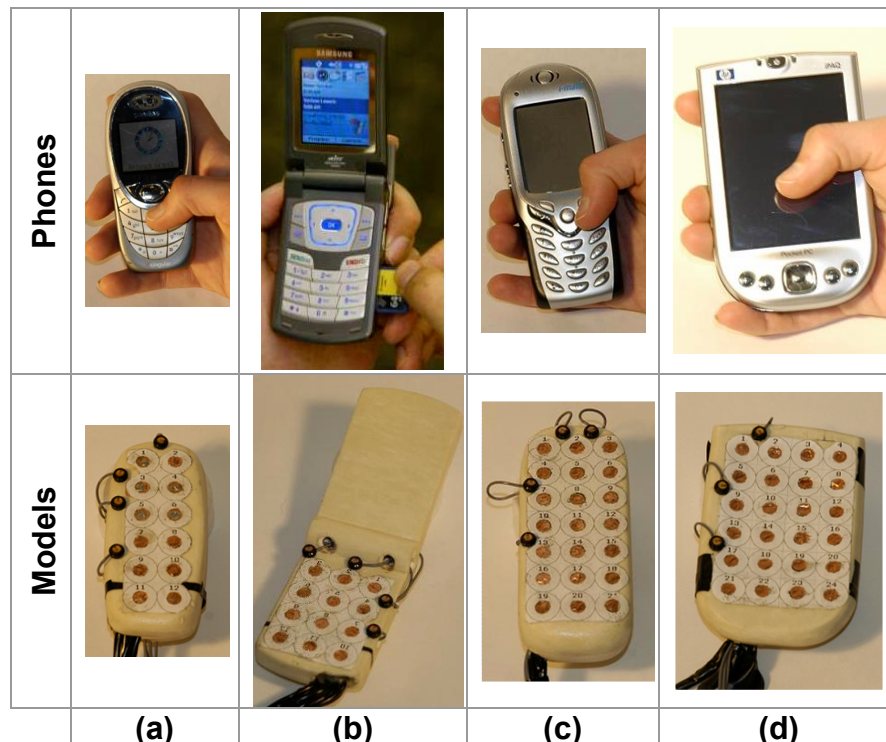


Figure 4. Devices we chose to represent a range of sizes and forms (top row) together with their study-ready models (bottom row): (a) SMALL, (b) FLIP, (c) LARGE, and (d) PDA.

Target design. A grid of circular targets 1.5 cm in diameter was affixed to the surface of each device. Circles were used for targets so that the sizes would not vary with direction of movement (MacKenzie & Buxton, 1992). The target size was selected to be large enough for the average-sized thumb, while also providing adequate surface coverage for each device. The grid dimensions for each device were: SMALL (2x5), FLIP (3x4), LARGE (3x7) and PDA (4x6), as shown in the bottom row of Figure 4.

Measurement. A typical measurement strategy for tapping tasks would involve a surface-based sensor to detect finger contact. Unfortunately, due to the number and variety of device sizes investigated, no technical solution was found to be as versatile, accurate or affordable as required. Instead we used Northern Digital Inc.'s OPTOTRAK 3020 motion analysis system designed for fine-grained tracking of motor movement. The OPTOTRAK uses 3 cameras to determine the precise 3D coordinates of infrared emitting diodes (IREDs). Three planar IREDs attached to the surface of each device defined a local coordinate system, and a fourth IRED provided redundancy (see Figure 4, bottom row). The spatial positions of two markers affixed to each participant's right thumb were then translated with respect to the coordinate system of the device to establish relative movement trajectories. Diode positions were sampled at 100Hz, and data were post processed to derive taps from thumb minima.

Software. Data collection and experiment software was run on a Gateway 2000 Pentium II with 256 MB of RAM running Windows 98.

Participants

Twenty participants were recruited via fliers posted in our Department of Computer Science, with the only restriction that participants be right-handed. Participants (15 male) ranged in age from 18 to 35 years with a median age of 25 years. Participants received \$20 for their time.

Design

For each target on each device (SMALL, FLIP, LARGE, and PDA), users performed all combinations of *distance* (1 or 2 circles) x *direction* (↑, ↔, ↗, ↘) tasks that could be supported by the geometry of the device. For example, SMALL could not accommodate trials of distance 2 circles in the directions (↔, ↗, ↘). Note that the grid layout results in *actual* distances that differ between *orthogonal* trials (↑, ↔) and *diagonal* trials (↗, ↘), which we consider explicitly in our analysis. For LARGE and PDA, trials of distance 4 circles were included as the geometry permitted. Finally each device included a ↗ and ↘ trial to opposite corners of the target grid. For each device, a small number of trials (1 for SMALL, LARGE and PDA, 3 for FLIP), selected at random, were repeated so as to make the total trial count divisible by four. The resulting number of trials for each device were: SMALL (32), FLIP (48), LARGE (108), and PDA (128). Since the larger devices had more surface targets to test, they required more trials.

Tasks

Users performed reciprocal tapping tasks in blocks as follows. For SMALL and FLIP, trials were divided equally into 2 blocks. For the LARGE and PDA, trials were divided equally into 4 blocks. Trials were assigned to blocks to achieve roughly equal numbers of *distance* x *direction* trials, distributed evenly over the device. Trials were announced by audio recording so that users could focus attention fully on the device. Users were presented with the name of two targets by number. For example, a voice recording would say “1 and 3”. After 1 second, a voice-recorded “start” was played. Users tapped as quickly as possible between the two targets, and after 5 seconds, a “stop” was played. After a 1.5 second delay the next trial began. Trials continued in succession to the end of the block, at which point the user was allowed to rest as desired, with no user resting more than 2 minutes. Device and block orders were assigned to subjects using a Latin Square, but the presentation of within-block trials was randomized for each user.

Procedure

Each session began with a brief description of the tasks to be performed and the equipment involved. Two IRED markers were then attached to the right thumb with two-sided tape. One diode was placed on the leftmost edge of the thumb nail, and a second on the left side of the thumb. The orthogonal placement was intended to maximize visibility of at least one of the diodes to the cameras at all times. The two marker wires were tethered loosely to the participant’s right wrist with medical tape.

The participant was seated in an armless chair, with the device held in the right hand, and the OPTOTRAK cameras positioned over the right shoulder. At this point the participant was given more detailed instruction about the tasks, and informed of the error conditions that might occur during the study: if at any point fewer than three of the device-affixed IREDs or none of the thumb IREDs were visible to the cameras, an out-of-sight error sound would be emitted, at which point he or she should continue the trial as naturally as possible while attempting to make

adjustments to improve diode visibility. Next, the participant was given the first device and performed a practice session of 24 trials, selected to represent a variety of distances, directions, and surface locations. During the practice trials, the administrator intentionally occluded the diodes to give the participant familiarity with the out-of-sight error sound and proper remedies. After completion of the practice trials and indication that the participant was ready, the study proper was begun.

During trials, participants were allowed to hold the devices in whatever manner supported their best performance. Since the instructions were presented audibly and with a short pause before the trial began, users could prepare their grip if desired. We chose not to control for grip in our study under the reasoning that it resembled real world settings, in which users have the freedom to adjust their grips to best suit the environment and task.

After all trials for a device were completed, users were allowed to rest while the next device was readied, typically 3 to 5 minutes; together with the rest period users were offered between trial blocks and counterbalanced device order, we hoped to minimize as much as possible the impact of fatigue on the results. After completing all trials for the last device, the participant completed a questionnaire, recording demographics and subjective ratings. Total session time was 2 hours, approximately an hour of which was devoted to data collection.

Measures

Raw 3D thumb movement data for each 5 second trial were truncated to the middle 3 seconds to eliminate artifacts resulting from initiation lag and anticipated trial completion, phenomena routinely observed by the administrator. In a post processing phase, taps were identified within the remaining 3 second interval and a single average tap time was computed from the difference in time between the onset of the first tap to the onset of the last tap, divided by one fewer than the total number of taps detected. In a post experiment questionnaire, participants assigned an overall rating of difficulty to each device (1-7, where 1 = easy, 7 = difficult), and indicated the device regions that were both easiest and hardest to interact with.

Data post processing. Since the 3D thumb position (x,y,z) was recorded relative to the device surface, the z -value represented the thumb height above the device. While one might assume that taps were those thumb positions for which the z -distance was 0, the IREDs were mounted on participants' thumbnails, and so never actually reached the surface of the device. Taps were instead defined as points when both the z -value and change in z -value (velocity) were minimal. For example, plotting z -values over time reveals a wave pattern whose valleys indicate taps (Figure 5).

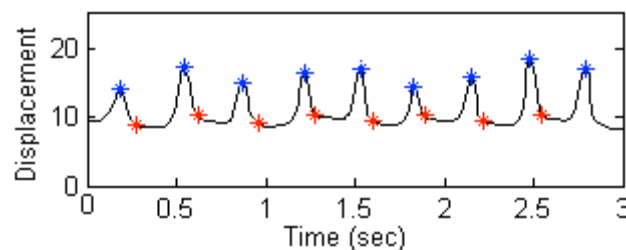


Figure 5. Example MATLAB output of the thumb's distance from the surface of the device. Blue and red stars depict the auto-detected peaks and valleys respectively.

Raw data was first preprocessed to extract the middle 3 seconds of each trial as well as to select the thumb diode with the most complete data set (e.g., the fewest number of missing frames, or if equal, the one with the most compact windows of missing frames). Linear interpolation was performed on missing frames if the gap was less than 100 ms. Missing frames included those lost due to out-of-sight errors, as well as occasional frames dropped by the collection hardware.

The data was then analyzed by the PICKEXTR MATLAB function to identify extrema in a signal. This function is provided with the RelPhase.Box Matlab toolbox for relative phase analysis of oscillatory systems (Dijkstra, Giese, & Schöner, 1997). The accuracy of the tap classifier was verified by inspecting a visual representation (Figure 5) of each trial. When required, corrections were made as follows: (1) valid endpoints were preserved, (2) if intermediate taps were missing, they were added, (3) if intermediate taps were incorrect, they were recoded by hand, and (4) if endpoints were invalid, the entire signal was coded by hand. Since average tap time was calculated as the number, not placement, of intervening taps, this method minimized as much as possible the bias of human annotation. Of the trials included for statistical analysis, 1.3% were discarded because they could not be encoded by machine or human, or had less than 1.5 seconds of encodable signal.

Results

The goal of our analysis was to understand whether user performance was influenced by device size, task region, and movement direction. To maximize comparison among the devices, we limited the analysis to trials with distances of 1 or 2 circles since the geometries of all but the smallest device (SMALL) supported these trials in all four movement directions. To address the fact that actual movement distance differed between orthogonal and diagonal trials, we analyzed these groups separately. For all analyses, Huynh-Felt corrections were used when the sphericity assumption was violated, and Bonferroni corrections were used for post hoc comparisons.

Movement Direction. A 2 (*distance*) x 2 (*direction*) repeated measures analysis of variance (RM-ANOVA) was performed on mean task time data for both orthogonal trials (distances: 1, 2; directions: $\updownarrow$, $\leftrightarrow$) and diagonal trials (distances: 1.4, 2.8; directions: $\nearrow$, $\searrow$) for the three largest devices. Since SMALL did not support distance 2 trials in all four directions, a one-way RM-ANOVA was performed on mean task time for trials of distance 1 and 1.4.

SMALL: A main effect of direction was observed for diagonal trials ($F(1,19) = 65.1$, $p < .001$). Post hoc analyses showed that trials in the $\nearrow$ direction were performed significantly faster than those in the $\searrow$ direction (0.26 v. 0.28 ms, $p < .001$).

FLIP, LARGE, and PDA: Results were similar across the analyses of the three largest devices. Unsurprisingly, a main effect of distance was observed for both orthogonal and diagonal trials, with shorter trials significantly faster than longer trials. There were no further effects of direction or interaction between direction and distance for orthogonal trials ($\updownarrow$, $\leftrightarrow$). However, for diagonal trials, a main effect of direction was observed, with trials in the $\nearrow$ direction significantly faster than those in the $\searrow$ direction for all devices. In addition, a *distance x direction* interaction

showed performance differences between the diagonal trials were more pronounced for longer trials than shorter trials (Table 1).

Table 1. Mean time for movement direction and distance x direction for FLIP, LARGE, and PDA.

	Direction (↖ v. ↘)	F_{1,19}	p	Dist. x Dir. (↖ v. ↘)	F_{1,19}	p
FLIP	.31 v. .35 ms	50.5	<.001	(1.4) .27 v. .30 ms (2.8) .34 v. .41 ms	14.6 34.5	< .001 < .001
LARGE	.31 v. .36 ms	46.1	<.001	(1.4) .27 v. .28 ms (2.8) .35 v. .42 ms	28.0 44.2	< .001 < .001
PDA	.32 v. .36 ms	46.5	<.001	(1.4) .28 v. .30 ms (2.8) .36 v. .43 ms	23.0 38.0	< .001 < .001

Device Size

To determine if device size impacted comparable tasks across devices, we analyzed all trials performed in the lower right 3x4 region of the three largest devices using a 3 (*devices*) x 43 (*trials*) RM-ANOVA. While a main effect of trial was observed, this was expected, as trials of every distance and direction were included for analysis. Yet no effects of *device* or *device x trial* were found.

Target Location

To determine if target location affected performance, we analyzed task time for the shortest tasks for each device. We chose short tasks because they provide high granularity for discriminating among device locations. Since direction was shown to affect task time for diagonal trials, only orthogonal tasks could be considered. For each device, a one-way RM-ANOVA was performed on mean trial time, with the number of trials varying by device.

A main effect of target location was observed for SMALL ($F(8.6, 163.3) = 2.1$, $p = .032$), FLIP ($F(11.5, 218.4) = 3.5$, $p < .001$) and PDA ($F(9.8, 188.1) = 3.9$, $p < .001$), but not for LARGE. However, in post hoc analyses, only PDA had a reasonable number of trials that differed significantly from one another. Since it is difficult to draw helpful conclusions from specific pairs of trials, we explored two aggregation techniques.

Subject-derived regions. Based on subjective opinion of which regions were easiest for each device, we divided tasks into 3 groups (E)asy, (M)edium, and (H)ard. Tasks for SMALL and FLIP were assigned to only E and M groups. A one-way RM-ANOVA on mean group task time was performed for each device. A main effect of group was found for FLIP ($F(5.5, 105.1) = 11.3$, $p < .001$), LARGE ($F(3.1, 58.7) = 8.4$, $p < .001$), and PDA ($F(4.8, 91.0) = 22.0$, $p < .001$). Post hoc analyses showed all groups differed significantly from each other for FLIP and PDA. For LARGE, E and M were significantly faster than H, but were indistinguishable otherwise, so we collapsed them to E (Table 2, third column).

Data-derived regions. For each device we ordered tasks by mean tap time, and then segmented them into seven groups. If the number of trials was not divisible by 7, the remainder trials were included in the middle group. A one-way RM-ANOVA on mean group task time was performed for each device. A main effect of group was found for FLIP ($F(5.5, 105.1) = 11.3, p < .001$), LARGE ($F(3.1, 58.7) = 8.4, p < .001$), and PDA ($F(4.8, 91.0) = 22.0, p < .001$). From these results, groups were labeled *fastest* and *slowest* such that all groups in *fastest* were significantly faster than all groups in *slowest*, according to post hoc analyses. Trials in these groups are shown visually in the rightmost column of Table 2. Mean task time for *fastest* v. *slowest* trials for each device were FLIP (0.26 v. 0.28 ms), LARGE (0.25 v. 0.28 ms), and PDA (0.26 v. 0.29 ms).

Table 2. Preference and movement time maps for each device. Depth of color in columns 1 and 2 indicate stronger user agreement.

	Subjective Difficult Regions	Subjective Easy Regions	Preference -Derived Regions Easy — Medium ▨ Hard —	Data- Derived Regions Fastest — Slowest —
SMALL				
FLIP				
LARGE				
PDA				

Subjective Preferences

After completing all trials, users were presented with diagrams of each device similar to those in the first two columns of Table 2 and asked to identify the targets they found most easy and most difficult to interact with. Aggregating results across users yielded a preference “map” for the least and most accessible targets of each device (columns 1 and 2 of Table 2), with darker regions indicating more agreement among participants. We see that for each device the two representations are roughly inverses of one another.

In addition to region marking, we asked users to rate the overall difficulty of managing each device with one hand on a 7 point scale (7 = most comfortable). Average ratings from most to least comfortable were as follows: SMALL (6.4), FLIP (5.4), LARGE (4.1) and PDA (3.0).

Thumb Movement Summary

The findings from our analysis of thumb movement suggest the following guidelines. First, thumb movement in the ↖ direction is difficult for right-handed users regardless of device size. Presumably the difficulty arises from the considerable flexion required to perform these types of tasks. Under this reasoning, the opposite movement ↗ would be difficult for left-handed users, so conservative designs should constrain repetitive movement to ↓ and ↔ directions, especially for repetitive tasks such as text or data entry.

Second, device region affects both task performance and perceived difficulty. Not only did the slowest trials correspond to those regions users found most difficult, but fastest trials also matched those regions users found most easy (Table 2). In general, regions within reach of the thumb were fastest and most comfortable, favoring those toward the midline of the device - a “sweet spot” that required movement primarily from the base of the thumb. The lower right corners of the devices present an exception in that they are biomechanically awkward to reach because they are “too close” rather than “too far”.

It is important to recognize that the absolute time differences between fastest and slowest regions are not so great as to provide the basis for our design recommendations. Rather, the significant slowdown (7%-12%) suggests that a mechanical and/or physical encumbrance is to blame, which is of concern primarily from an ergonomics perspective. In fact, we believe that the slowdowns we found should be thought of as optimistic, since they capture only localized movement and required substantial changes in user grip; subjective opinion, user observations and practical experience indicate that designers should be cautioned against using the entire surface for thumb interaction, especially for larger devices. We instead recommend placing interaction objects centrally to accommodate both left and right handed users, or offering configurable displays. Since hand size and thumb length will differ by individual, designs should strive to support a range of users.

Finally, the result that users performed trials in the lower right 3x4 sub-grid of the three largest devices equally well suggests that large devices do not inherently impede thumb movement. Rather, larger devices simply have more areas that are out of thumb reach, and so have more regions that are inappropriate for object placement in one-handed designs. Together with user

opinion that larger devices were more difficult to manage suggests that the current trend toward smaller device forms benefits one handed device control.

FUTURE TRENDS

Market forces are driving the sizes of mobile devices down and at the same time driving feature sets up. These concurrent trends *alone* compromise the ease and expressiveness of information interaction and presentation, as shrinking keypads and screens are being used to access ever increasing data sets. Yet at the same time, mobile users often are carrying personal effects, opening doors, holding handrails, or otherwise needing the use of a hand to manage the environment, which then leaves only a single hand available for device operation. Indeed, results from our field study and survey of mobile device use have confirmed that one-handed device operation is widespread - not limited to a niche user segment - and that users would prefer to use one hand more often than current designs allow.

Commercial trends for supporting single handed device operation have focused on thumb operation of touchscreen based devices. Several thumb-based virtual keypads are available from third party vendors, such as the Phrase-It® keypad from Prevalent Devices, www.prevalentdevices.com. One of the few handheld *systems* dedicated to single-handed operation is the touchscreen-based N1m phone by Neonode, www.neonode.com, which supports application navigation and interaction using only thumb taps and sweeps. However, the N1m's primary use is as a phone, camera and media player, rather than a personal data manager. As such, the N1m is not designed to support rich graphical interfaces or data interactions.

As the mobile user base expands, so do device storage capacities and wireless services. Not only are mobile devices accumulating more resident data, but they are increasingly used as front-end interfaces to external data sets. Given the broad range of data sets and tasks users expect of today's devices, single-handed device support will need to generalize beyond specific applications and technologies seen today. Our work seeks to understand some of the basic human factors involved in single handed device use, in order that this knowledge can be applied generally to the variety of tasks and device forms users demand for portable computing.

CONCLUSION

In an effort to understand the one-handed interaction needs of mobile device users, we looked at a broad range of device use. Our field study showed that for at least one class of user (travelers), mobile phones are most often used with one hand, and that this behavior seems to correlate with activity, such as walking or holding items in the other hand. Our survey revealed that the vast majority of users want to use one hand for interacting with mobile devices, but that current interfaces, especially for touchscreens, are not designed to support dedicated single handed use. Finally, an empirical evaluation of thumb interaction on varying-sized devices suggests that 1) mid-device regions are easiest to access; 2) the position of a target with respect to the thumb impacts performance more than device size, and finally 3) ↖ movement is difficult for right-handed users and degrades with movement distance.

REFERENCES

- Department of Health and Human Services. (2005). 45 CFR Part 46, Protection of Human Subjects. In (*Section 46.101(b)*). Retrieved January 16, 2007 from <http://www.hhs.gov/ohrp/humansubjects/guidance/45cfr46.htm>.
- Apple. (2006). iPod. Retrieved January 16, 2007 from <http://www.apple.com/ipod/ipod.html>.
- Barmakian, J. T. (1992). Anatomy of the joints of the thumb. *Hand Clinics*, 8(4), 681-691.
- Barnard, L., Yi, J. S., Jacko, J. A., & Sears, A. (2005). An empirical comparison of use-in-motion evaluation scenarios for mobile computing. *International Journal of Human-Computer Studies*, 62, 487-520.
- Bourbonnais, D., Forget, R., Carrier, L., & Lepage, Y. (1993). Multidirectional analysis of maximal voluntary contractions of the thumb. *Journal of Hand Therapy*, 6(4), 313-318.
- Dijkstra, T. M., Giese, M. A., & Schöner, G. (1997). RelPhase.box: A Matlab toolbox for relative phase analysis of oscillatory systems. Retrieved January 17, 2007, from <http://www.psy.ohio-state.edu/visionlab/dijkstra/Software/RelPhase.sit>
- Dong, L., Watters, C., & Duffy, J. (2005). Comparing two one-handed access methods on a PDA. *International Conference on Human Computer Interaction with Mobile Devices and Services*, (pp. 235-238). ACM Press.
- Hinckley, K., Pierce, J., Sinclair, M., & Horvitz, E. (2000). Sensing techniques for mobile interaction. *ACM Symposium on User Interface Software and Technology*, (pp. 91-100). ACM Press.
- Hirota, N. (2003). Reassessing current cell phone designs: using thumb input effectively. *Extended Abstracts of the ACM Conference on Human Factors in Computing*, (pp. 938-939). ACM Press.
- Karlson, A. K., Bederson, B. B., & SanGiovanni, J. (2005). AppLens and LaunchTile: two designs for one-handed thumb use on small devices. *ACM Conference on Human Factors in Computing*, (pp. 201-210). ACM Press.
- Kristoffersen, S., & Ljungberg, F. (1999). Making place to make it work: empirical exploration of HCI for mobile CSCW. *ACM Conference on Supporting Group Work*, (pp. 276-285). ACM Press.
- Kuo, L., Cooley, W., Kaufman, K., Su, F., & An, K. (2004). A kinematic method to calculate the workspace of the TMC joint. *Proceedings of the Institute of Mechanical Engineering H*, 218(2), 143-149.
- Li, Z.-M., & Harkness, D. A. (2004). Circumferential force production of the thumb. *Medical Engineering and Physics*, 26, 663-670.

- Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., et al. (2004). Twiddler typing: one-handed chording text entry for mobile phones. *ACM Conference on Human Factors in Computing*, (pp. 671-678). ACM Press.
- MacKenzie, I. S., & Buxton, W. (1992). Extending Fitts' Law to two-dimensional tasks. *ACM Conference on Human Factors in Computing*, (pp. 219-226). ACM Press.
- Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. *ACM Conference on Human Factors in Computing*, (pp. 919-928). ACM Press.
- Pascoe, J., Ryan, N., & Mores, D. (2000). Using while moving: HCI issues in fieldwork environments. *Transactions on Computer-Human Interaction*, 7(3), 417-437.
- Pirhonen, P., Brewster, S. A., & Holguin, C. (2002). Gestural and audio metaphors as a means of control in mobile devices. *ACM Conference on Human Factors in Computing*, (pp. 291-298). ACM Press.
- Pittet, S., Hart, T. J., Cosimo, C., Garofano, G., Ingelbrecht, N., & Liew, E. (2005). *Forecast: Mobile Terminals, Worldwide, 2000-2009*: Gartner, Inc.
- Rekimoto, J. (1996). Tilting operations for small screen interfaces. *ACM Symposium on User Interface Software and Technology*, (pp. 167-168). ACM Press.
- Wigdor, D., & Balakrishnan, R. (2003). TiltText: using tilt for text input to mobile phones. *ACM Symposium on User Interface Software and Technology*, (pp. 81-90). ACM Press.

KEY TERMS

Biomechanics – the study of muscular mechanics and activity.

Candybar Phone – a traditional-style mobile phone with a rigid rectangular form, typically about 3 times longer than wide.

Ergonomics – often used synonymously with “human factors”, ergonomics is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance (definition adopted by the International Ergonomics Association in August 2000).

Flip Phone – a popular form factor for a mobile phone that structures the phone into two halves that are hinged like a clamshell: the numeric keypad is placed on the lower half, and the display is placed on the upper half. In the closed position the screen and the keypad are protected from inadvertent damage or activation, while in the open position users can angle the screen for optimal visibility.

Mobile or Cellular Phone – a portable electronic device which, at a minimum, supports long-range telecommunications. Today’s mobile phones support a much broader range of functions for personal data and data management activities, including data exchange with a personal desktop computer, music play, and photo capturing.

PDA – Personal digital assistant. The original handheld device designed expressly for supporting personal data management on the go. PDAs typically have touch sensitive screens and are operated with a stylus. Recent models may also include a miniaturized Qwerty keyboard for text entry.

APPENDIX 7

Audio Augmented Reality: A Prototype Automated Tour Guide

Benjamin B. Bederson*
Bell Communications Research
445 South Street
Morristown, NJ 07960
(bederson@bellcore.com)

KEYWORDS

Audio augmented reality, museum tour guides, virtual reality, social interaction.

ABSTRACT

Augmented reality (or computer augmented environments as it is sometimes called) uses computers to enhance the richness of the real world. It differs from virtual reality in that it doesn't attempt to replace the real world. Our prototype automated tour guide superimposes audio on the world based on where a user is located. We propose this technique for use as an automated tour guide in museums and expect it will enhance the social aspects of museum visits, compared to taped tour guides.

INTRODUCTION

For many types of information retrieval, social interaction is critical to the experience. For instance, why do people go to live music concerts instead of listening to compact discs at home? Partly because of the different quality of sound, but it is also largely due to the social experience of being at the live show, being with your friends, and being part of the audience.

Many people believe that computation enhances our everyday lives. But many of the forms in which computers aid us also work to isolate us. Perhaps the most glaring example of this is *virtual reality*, where the main point is to take us out of the physical world by replacing our senses with computer-generated ones.

Augmented reality, on the other hand, is an attempt to combine our real world interactions with the richness of computational information without isolating people from each other. The basic idea is to superimpose computer generated data on top of the real world, as the person moves within it. This idea was originally described by Myron Krueger [4], but now a number of groups are

beginning to experiment with it (see [2] for a collection of several research articles, or [1] for a survey).

MUSEUM TOURS

One place a low-tech version of augmented reality has long been in the marketplace is museums. It is quite common for museums to rent audio-tape tour guides that viewers carry around with them as they tour the exhibits.

While this technology works reasonably well, many people using it become frustrated because it seems to obstruct some of their social purposes in attending the museum. As with music, part of the reason many people go to museums is to socialize, to be with friends and to discuss the exhibit as they experience it. Taped tour guides conflict with these goals because the tapes are linear, pre-planned, and go at their own pace. This makes it hard to stay with friends because if one person turns off their tape temporarily, it is very difficult to get synchronized again.

In addition, the taped tours typically describe only a relatively small subset of the pieces on exhibit. The pieces described on tape may not be the ones a particular viewer is interested in hearing. But because the tape must be accessed linearly, it is impossible to skip over or access descriptions out of order.

AUDIO AUGMENTED REALITY PROTOTYPE

A more technologically sophisticated tour guide, on the other hand, can offer the benefits of automation without the social conflicts caused by the taped tour guide.

We have built a prototype audio augmented reality-based tour guide. This system replaces analog audio tapes with random access digital audio. In addition, it adds a micro-computer and an invisible spatial locating device that allow much more freedom for the participant.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of ACM. To copy otherwise, or to republish, requires a fee and/or specific permission.
CHI' Companion 95, Denver, Colorado, USA
© 1995 ACM 0-89791-755-3/95/0005...\$3.50

* Current address: bederson@cs.unm.edu,
Computer Science Department, University of New Mexico,
Albuquerque, NM 87131.

The idea is to allow visitors to hear descriptions of pieces just by walking up to them. Descriptions may be heard in any order, and can be cut short by walking away. Friends visiting together can stay in sync very naturally just by walking up to a piece at the same time. When they walk away from a piece the description will stop and they may talk amongst themselves.

Because the digital storage device can hold two and a half hours of audio, the tour planner can describe many more pieces in the exhibit than any viewer will likely see. This gives much more control to the viewer since they can hear descriptions of pieces they want to hear about.

This prototype consists of a few devices that the viewer carries with them: a random-access digital audio source (modified Sony MiniDisc™ player), a microprocessor (Motorola M68332), and a custom infrared receiver that tells the computer where the viewer is. A very small infrared transmitter is placed in the ceiling above each piece to be described. It transmits a unique identifying number that the computer uses to identify the location of the viewer. As the viewer walks around, the computer simply controls the audio source to play or stop playing pre-recorded descriptions. See Figure 1 for a schematic diagram of the system.

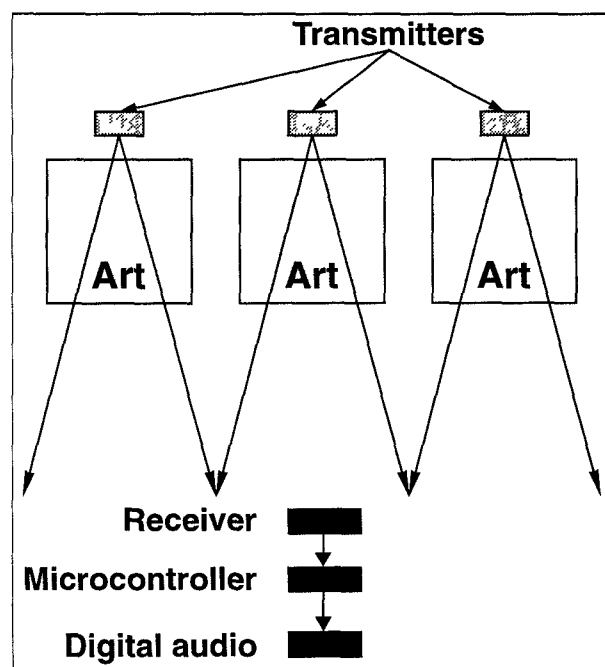


Figure 1: Schematic of automated tour guide prototype.

This type of technology offers the possibility of modifying the descriptions of pieces based on the particular interaction history of this viewer with the exhibit. This notion, more generally referred to as *history-enriched*

digital objects is described in [3]. For example, if the computer noticed that the viewer had looked at several pieces by one artist, it could offer extra background information on that artist. Or it could relate pieces currently being viewed to particular pieces that this viewer recently saw. There could even be a "restroom button" on the device which would give the viewer directions to the nearest restroom.

One group [5] has created a similar technology, but in reverse, to identify the locations of individuals to a centralized computer system. In this system, people carry around small infrared transmitters, and there is a single receiver in the ceiling of each room.

Some museums have begun experimenting with new guide technologies, but they still have some limitations. One popular approach is to have some local audio broadcast technology so that each piece broadcasts a description of itself in continuous cycles. The disadvantage of this approach is that participants often walk up to a piece mid-way through a description. In this case, they hear the second half first, and then the description starts over again. It is vital that the user carry the audio source with them so the descriptions can be heard on the user's time schedule.

CONCLUSION

Augmenting rather than replacing the external world can take advantage of computation without conflicting with social concerns. We introduce a prototype museum tour guide based on augmented reality which shows that more appropriate technology can enrich our experiences without interfering with our social interactions as much.

ACKNOWLEDGEMENTS

I am indebted to Jim Hollan. It was with him that I conceived of this idea on a subway ride to New York City. And I am thankful to Chris Calabrese who wrote much of the software for the prototype.

REFERENCES

- [1] Bederson, B.B and Druin, A. *Computer Augmented Environments: New Places to Learn, Work and Play*, ed. Jakob Nielsen, in *Advances in Human Computer Interaction*, Vol. 5, Ablex Press, In Press.
- [2] *Communications of the ACM*, Special Issue on Augmented Environments, 36 (7), July, 1993.
- [3] Hill, W.C., Hollan, J.D., Wroblewski, D. and McCandless, T., *Edit Wear and Read Wear*, in *Proceedings of Human Factors in Computing Systems (CHI '92)*, ACM Press, pp. 3-9, 1992.
- [4] Krueger, M.W., *Artificial Reality II*, Addison-Wesley, 1991.
- [5] Want, R., Hopper, A., Falcao, V. and Gibbons, J., *The Active Badge Location System*, *ACM Transactions on Information Systems*. Vol. 10 (1), pp. 91-102, Jan. 1992.