

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, SAMSUNG ELECTRONICS CO. LTD.,
AND SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioner

v.

MULTIMODAL MEDIA LLC,
Patent Owner.

Case No. IPR2024-00064
U.S. Patent No. 10,552,030

**DECLARATION OF DR. BENJAMIN B. BEDERSON IN SUPPORT
OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,552,030**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS	1
III. INFORMATION CONSIDERED	9
IV. RELEVANT LEGAL STANDARDS	9
A. Claim Interpretation	9
B. Perspective of One of Ordinary Skill in the Art.....	10
C. Obviousness.....	10
V. LEVEL OF ORDINARY SKILL IN THE ART	13
VI. SUMMARY OF MY OPINIONS	14
VII. TECHNOLOGICAL BACKGROUND	15
A. Mobile Applications for Recording Media Data On Electronic Devices and the Claimed Recording Modes Were Known.....	15
B. Gesture Recognition Using Touch Interfaces, Including for Recording Media, Was Known	16
1. Press and Hold / Push to Talk	20
2. Slide and hold.....	22
3. Slide and release.....	27
4. Tap to end.....	33
C. Transitioning Between Different Modes Based on Gestures Was Known	34
D. Transmitting (and Not Transmitting) Media to A Server or Device Based on Gestures Was Known.....	39
E. Defining Multiple Interface Regions and Dynamically Changing Functions at Interface Regions Based on Gestures Was Known.....	41
VIII. THE ‘030 PATENT AND ITS FILE HISTORY	47
IX. CLAIM CONSTRUCTION	50
A. “interface regions” (Claims 1, 2, 8, 9)	51
B. “gesture based media recording application” (Claims 1-4, 6, 8,	

TABLE OF CONTENTS

(continued)

	Page
13).....	51
C. “gesture” (Claims 1-4, 6, 8-11, and 13)	51
D. “detect[ing] a [first/second/third] gesture” (Claims 1-3 and 8-10)	52
E. “interface definition module” (Claim 8)	52
F. “detection module” (Claims 8-11)	53
G. “action management module” (Claim 8).....	53
X. OVERVIEW OF THE APPLIED PRIOR ART REFERENCES	54
A. Lee’041 (Ex-1014)	54
B. Ohashi (Ex-1017)	56
C. Lee’804 (Ex-1015)	58
D. D’Souza (Ex-1018).....	60
XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS	62
A. Ground 1: Claims 1, 3-8, and 10-14 are obvious over Lee’041	62
1. Independent Claims 1 and 8.....	62
a. Elements 1[Pre]/8[Pre]: A computer implemented method for recording media data, comprising:	62
b. Element 8[A]: a processor;	63
Element 8[B(i)]: a non-transitory computer readable storage medium communicatively coupled to said processor, ...	63
c. Element 1[A]: providing a gesture based media recording application executable by at least one processor configured to manage a graphical user interface of an electronic device for said recording of media data in one or more recording modes on said electronic device;	64
Element 8[B(ii)]: said non-transitory computer readable storage medium configured to store a gesture based media recording application, ...	64

TABLE OF CONTENTS
(continued)

	Page
Element 8[B(iii)]: wherein said gesture based media recording application, when executed by said processor, manages a graphical user interface of an electronic device for said recording of media data;	64
Element 8[B(iv)]: said gesture based media recording application comprising:.....	64
d. Element 1[B]: defining one or more interface regions on said graphical user interface of said electronic device by said gesture based media recording application;	68
Element 8[C]: an interface definition module configured to define a plurality of interface regions on said graphical user interface of said electronic device;.....	68
e. Element 1[C]: detecting a first gesture from among multiple gestures on a first of said defined interface regions, by said gesture based media recording application, wherein said first gesture is a press and hold gesture;.....	70
Element 8[D]: a detection module configured to detect a first gesture from among multiple gestures on a first of said defined interface regions, wherein said first gesture is a press and hold gesture;.....	70
f. Element 1[D]: starting said recording of media data in a push to talk (PTT) recording mode, by said gesture based media recording application, on said detection of said first gesture on said first of said defined interface regions;.....	72
Element 8[E]: an action management module configured to start said recording of media data in a push to talk (PTT) recording mode, on said detection of said first gesture on said first of said defined interface regions;	72
g. Element 1[E]: dynamically changing, by said gesture based media recording application,	

TABLE OF CONTENTS

(continued)

	Page
functions associated with said one or more defined interface regions based on said detected first gesture on said one of said defined interface regions; and.....	74
Element 8[F]: said interface definition module configured to dynamically change functions associated with said one or more defined interface regions based on said detected first gesture on said one of said defined interface regions; and	74
h. Element 1[F(i)]: detecting a second gesture from among said multiple gestures, by said gesture based media recording application, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and	77
Element 8[G(i)]: said detection module further configured to detect a second gesture from among said multiple gestures, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and	77
i. Element 1[F(ii)]: wherein said recording of said media data continues uninterrupted during and after performance of said second gesture.	81
Element 8[G(ii)]: wherein said recording of said media data continues uninterrupted during and after performance of said second gesture.....	81
2. Dependent Claims 3-7 and 10-14	81
a. Element 3[A]: The computer implemented method of claim 1, further comprising: detecting a third gesture from among said multiple gestures on said second of said defined interface regions, by said	

TABLE OF CONTENTS
(continued)

Page

	gesture based media recording application, wherein said third gesture is a release gesture from said second of said interface regions, and Element 10[A]: The computer implemented system of claim 8, further comprising: said detection module further configured to detect a third gesture from among said multiple gestures on said second of said defined interface regions, by said gesture based media recording application, wherein said third gesture is a release gesture from said second of said interface regions, and.....	81
b.	Element 3[B]: switching said recording of said media data to a tap to end (TTE) mode, wherein said recording of said media data continues uninterrupted during and after performance of said third gesture. Element 10[B]: switching said recording of said media data to a tap to end (TTE) mode, wherein said recording of said media data continues uninterrupted during and after performance of said third gesture.	84
c.	Element 4[A]: The computer implemented method of claim 3, further comprising: detecting a fourth gesture from among said multiple gestures on said one of said defined interface regions, by said gesture based media recording application, wherein said fourth gesture is a tap gesture, and Element 11[A]: The computer implemented system of claim 10, further comprising: said detection module further configured to detect a fourth gesture from among said multiple gestures on said one of said defined interface regions, by said gesture based media recording application, wherein said fourth gesture is a tap gesture, and.....	85
d.	Element 4[B]: wherein said recording of said media data is ended on performing said fourth	

TABLE OF CONTENTS
(continued)

	Page
gesture. Element 11[B]: wherein said recording of said media data is ended on performing said fourth gesture.....	88
e. Claim 5: The computer implemented method of claim 4, wherein said recorded media data is one or more of saved locally, deleted, or transmitted to another electronic device or the server. Claim 12: The computer implemented system of claim 11, wherein said recorded media data is one or more of saved locally, deleted, or transmitted to another electronic device or the server.	89
f. Claim 6: The computer implemented method of claim 1, further comprising sending said media data to the server via a network at one of: (a) continuously while recording said media data, (b) at predefined time intervals after initiation of said recording of said media data, and (c) after completion of said recording of said media data by said gesture based media recording application. Claim 13: The computer implemented system of claim 8, further comprising sending said media data to the server via a network at one of: (a) continuously while recording said media data, (b) at predefined time intervals after initiation of said recording of said media data, and (c) after completion of said recording of said media data by said gesture based media recording application.	91
g. Claim 7: The computer implemented method of claim 1, wherein said media data comprises one of audio data, video data, audiovisual data, image data, multimedia data, and any combination thereof. Claim 14: The computer implemented system of claim 8], wherein said media data comprises one of audio data, video data, audiovisual data, image data, multimedia data, and	

TABLE OF CONTENTS
(continued)

	Page
any combination thereof.....	92
B. Ground 2: Claims 1, 3-8, and 10-14 are obvious over Lee'041 in view of Ohashi.....	93
1. A POSITA would have been motivated to combine Lee'041 and Ohashi and had a reasonable expectation of success.....	93
2. Independent Claims 1 and 8.....	98
a. Element 1[F(i)]: detecting a second gesture from among said multiple gestures, by said gesture based media recording application, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and	98
Element 8[G(i)]: said detection module further configured to detect a second gesture from among said multiple gestures, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and	98
3. Dependent Claims 3-7 and 10-14	104
C. Ground 3: Claims 2 and 9 are obvious over Lee'041 in view of Ohashi in further view of Lee'804 and D'Souza	104
1. Dependent Claims 2 and 9	104
a. Element 2[A]: The computer implemented method of claim 1, further comprising: detecting a third gesture from among said multiple gestures, by said gesture based media recording application, wherein said third gesture is a slide and release gesture, or a swipe and release gesture from said second of said defined interface regions to a third	

TABLE OF CONTENTS
(continued)

Page

of said defined interface regions proximate to said first of said defined interface regions, and Element 9[A]: The computer implemented system of claim 8, further comprising: said detection module further configured to detect a third gesture from among said multiple gestures, wherein said third gesture is a slide and release gesture, or a swipe and release gesture from said second of said defined interface regions to a third of said defined interface regions proximate to said first of said defined interface regions, and.....	104
b. Element 2[B]: wherein said recording of said media data is canceled on performing said third gesture, and Element 9[B]: wherein said recording of said media data is canceled on performing said third gesture, and Element 2[C]: wherein there is no transmission of already recorded media data to another electronic device or a server. Element 9[C]: wherein there is no transmission of already recorded media data to another electronic device or a server.....	111
XII. CONCLUSION.....	118

LIST OF EXHIBITS

Ex-1001	U.S. Patent No. 10,552,030
Ex-1002	Declaration of Dr. Benjamin B. Bederson
Ex-1003	Curriculum Vitae of Dr. Benjamin B. Bederson
Ex-1004	File History of U.S. Patent No. 10,552,030
Ex-1005	INTENTIONALLY LEFT BLANK
Ex-1006	INTENTIONALLY LEFT BLANK
Ex-1007	INTENTIONALLY LEFT BLANK
Ex-1008	INTENTIONALLY LEFT BLANK
Ex-1009	INTENTIONALLY LEFT BLANK
Ex-1010	INTENTIONALLY LEFT BLANK
Ex-1011	INTENTIONALLY LEFT BLANK
Ex-1012	INTENTIONALLY LEFT BLANK
Ex-1013	INTENTIONALLY LEFT BLANK
Ex-1014	U.S. Patent No. 8,812,041 (“Lee’041”)
Ex-1015	U.S. Patent No. 8,948,804 (“Lee’804”)
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	U.S. Patent Publication No. 2012/0229410 (“Ohashi”)
Ex-1018	U.S. Patent Publication No. 2012/0014684 (“D’Souza”)
Ex-1019	INTENTIONALLY LEFT BLANK
Ex-1020	Claim Mapping Table
Ex-1021	<i>Multimodal Media LLC v. Guangdong Oppo Mobile Telecommunications Corp., Ltd.</i> , 2:21-cv-00436, D.I. 65 (E.D. Tex. Apr. 24, 2023) (Claim Construction Order) (“ <i>Oppo</i> Litigation”)

Ex-1022	<i>Multimodal Media LLC v. TCL Technology Group Corporation et al.</i> , 2:22-cv-00463, D.I. 23 (E.D. Tex. Feb. 22, 2023) (Stay Order) (“TCL litigation”)
Ex-1023	<i>Multimodal Media LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2:22-cv-00462, Infringement Contentions (E.D. Tex. May 30, 2023) (“Samsung Infringement Contentions”)
Ex-1024	INTENTIONALLY LEFT BLANK
Ex-1025	INTENTIONALLY LEFT BLANK
Ex-1026	Benjamin B. Bederson & James D. Hollan, Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, UIST ‘94 Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology 17 (1994), DOI: http://dx.doi.org/10.1145/192426.192435
Ex-1027	INTENTIONALLY LEFT BLANK
Ex-1028	INTENTIONALLY LEFT BLANK
Ex-1029	INTENTIONALLY LEFT BLANK
Ex-1030	INTENTIONALLY LEFT BLANK
Ex-1031	David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson, Multi-modal text entry and selection on a mobile device (2010), In Proceedings of Graphics Interface 2010 (GI ‘10), Canadian Information Processing Society, CAN, 19–26
Ex-1032	INTENTIONALLY LEFT BLANK
Ex-1033	INTENTIONALLY LEFT BLANK
Ex-1034	INTENTIONALLY LEFT BLANK
Ex-1035	INTENTIONALLY LEFT BLANK
Ex-1036	INTENTIONALLY LEFT BLANK
Ex-1037	INTENTIONALLY LEFT BLANK
Ex-1038	INTENTIONALLY LEFT BLANK

Ex-1039	INTENTIONALLY LEFT BLANK
Ex-1040	INTENTIONALLY LEFT BLANK
Ex-1041	INTENTIONALLY LEFT BLANK
Ex-1042	INTENTIONALLY LEFT BLANK
Ex-1043	INTENTIONALLY LEFT BLANK
Ex-1044	INTENTIONALLY LEFT BLANK
Ex-1045	Ben Shneiderman & Catherine Plaisant, <i>Designing the User Interface</i> (4th ed., 2005)
Ex-1046	<i>Multimodal Media LLC v. Guangdong Oppo Mobile Telecommunications Corp., Ltd.</i> , 2:21-cv-00436, D.I. 74 (E.D. Tex. Sept. 6, 2023) (Order of Dismissal)
Ex-1047	INTENTIONALLY LEFT BLANK
Ex-1048	INTENTIONALLY LEFT BLANK
Ex-1049	INTENTIONALLY LEFT BLANK
Ex-1050	INTENTIONALLY LEFT BLANK
Ex-1051	INTENTIONALLY LEFT BLANK
Ex-1052	INTENTIONALLY LEFT BLANK
Ex-1053	INTENTIONALLY LEFT BLANK
Ex-1054	INTENTIONALLY LEFT BLANK
Ex-1055	INTENTIONALLY LEFT BLANK
Ex-1056	Ivan Edward Sutherland, <i>Sketchpad: A man-machine graphical communication system</i> , University of Cambridge Technical Report UCAM-CL-TR-574, ISSN 1476-2986, Number 574 (2003), available at: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf
Ex-1057	S.K. Lee, William Buxton, & K.C. Smith, A multi-touch three-dimensional touch-sensitive tablet, <i>Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '85)</i> , 21-

	25 (Apr. 1985), available at: DOI: https://doi.org/10.1145/317456.317461
Ex-1058	U.S. Patent No. 3,482,241
Ex-1059	U.S. Patent No. 10,140,301
Ex-1060	U.S. Patent Publication No. 2013/0019175
Ex-1061	U.S. Patent No. 9,767,681
Ex-1062	Tarmo Anttalainen, Introduction to Telecommunications Network Engineering, 77-84 (2d ed., 2003), available at: https://ieeexplore.ieee.org/document/9106127
Ex-1063	U.S. Patent Publication No. 2013/0227456
Ex-1064	U.S. Patent Publication No. 2014/0063313
Ex-1065	Rick Broida & Dave Johnson, How to Do Everything Palm® Pre™ (2010)
Ex-1066	Ken Hinckley et al., Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli, CHI 2005 (Apr. 2-7, 2005)
Ex-1067	PCT Publication WO2011/151501 (“Dolenc”)
Ex-1068	Kay Hofmeester & Jennifer Wolfe, Self-Revealing Gestures: Teaching New Touch Interactions in Windows 8, CHI 2012 (May 5-12, 2012)
Ex-1069	Xiangshi Ren & Shinju Moriya, Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks, 7 ACM Trans. Comput.-Hum. Interact. 3, 384-416 (Sept. 2000), available at: https://doi.org/10.1145/355324.355328
Ex-1070	Andrew Sears & Ben Shneiderman, High Precision Touchscreens: Design Strategies and Comparisons with a Mouse (Jan. 23, 1989), available at: https://api.drum.lib.umd.edu/server/api/core/bitstreams/9352f26a-e793-429a-80f8-65f606f41969/content
Ex-1071	Seon Joo Kim et al., Brush-and-Drag: A Multi-touch Interface for

Declaration of Dr. Benjamin B Bederson
U.S. Patent No. 10,552,030

	Photo Triaging, MobileHCI'12 (Sept. 21-24, 2012)
Ex-1072	U.S. Patent No. 9,195,330
Ex-1073	U.S. Patent Publication No. 2009/0015703
Ex-1074	U.S. Patent Publication No. 2013/0063645
Ex-1075	U.S. Patent Publication No. 2014/0078371
Ex-1076	U.S. Patent Publication No. 2010/0177157
Ex-1077	Microsoft Computer Dictionary 499 (5th ed., 2002)
Ex-1078	U.S. Patent Publication No. 2011/0099513
Ex-1079	U.S. Patent Publication No. 2012/0072866
Ex-1080	Newsroom, Apple Presents iPhone 4 (June 7, 2010), available at: https://www.apple.com/newsroom/2010/06/07Apple-Presents-iPhone-4/
Ex-1081	Wikipedia, Walkie-talkie, available at: https://en.wikipedia.org/wiki/Walkie-talkie

1. I, Dr. Benjamin B. Bederson, declare as follows.

I. INTRODUCTION

2. I have been retained by Google LLC, Samsung Electronics Co. Ltd., and Samsung Electronics America, Inc. (collectively, “Petitioner”) as an independent expert consultant in this *inter partes* review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”).

3. I have been asked by Petitioner Counsel (“Counsel”) to consider whether certain references disclose, teach and/or suggest the features recited in Claims 1-14 of U.S. Patent No. 10,552,030 (“the ’030 Patent”) (Ex-1001), which I understand is currently assigned to Patent Owner Multimodal Media LLC (“PO”).¹ My opinions and the bases for my opinions are set forth below

4. I am being compensated at my ordinary and customary consulting rate for my work, which is \$600 per hour. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other direct financial interest in this proceeding

II. BACKGROUND AND QUALIFICATIONS

5. All of my opinions stated in my Declaration are based on my own

¹ Where appropriate, I refer to exhibits that I understand are attached to the petition for IPR of the ’030 Patent.

personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in designing, developing, researching, and teaching the technology referenced in my Declaration.

6. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth in my Declaration. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as Ex-1003. The following provides a brief overview of some of my experience that is relevant to the matters set forth in my Declaration.

7. I am currently Professor Emeritus of Computer Science at the University of Maryland (“UMD”). From 2014 to 2018, I was the Associate Provost of Learning Initiatives and Executive Director of the Teaching and Learning Transformation Center at the UMD. I am a member and previous director of the Human-Computer Interaction Lab (“HCIL”), the oldest and one of the best known Human-Computer Interaction (“HCI”) research groups in the country. I am co-founder of the J.S. Bryant School, a therapeutic high school planning to open in 2025. I was also co-founder and Chief Scientist of Zumobi, Inc. from 2006 to 2014, a Seattle-based startup that is a publisher of content applications and advertising platforms for smartphones. I am also co-founder and co-director of the International Children’s Digital Library (“ICDL”), a web site launched in 2002 that provides the

world's largest collection of freely available online children's books from around the world at www.childrenslibrary.org with an interface aimed to make it easy for children and adults to search and read children's books online. I was also co-founder and prior Chief Technology Officer of Hazel Analytics, a data analytics company to improve food safety and better public health whose product sends alerts in warranted circumstances. In addition, I have for more than 15 years consulted for numerous companies in the area of user interfaces, including Microsoft, Logitech, the Palo Alto Research Center, Sony, Lockheed Martin, Hillcrest Labs, and NASA Goddard Space Flight Center.

8. The claims of the '030 Patent generally relate to multimedia and touch interface technologies. For more than 30 years, I have studied, designed, and worked in the field of computer science and HCI. My experience includes 30 years of teaching and research, with research interests in HCI and the software and technology underlying today's interactive computing systems. This includes the design and implementation of hardware and software systems including the use of interactive voice interfaces as well as many other kinds of sensors and interfaces for electronic devices on a range of devices, including embedded systems, controllers, smart phones and PDAs.

9. At UMD, I have been focused primarily on the area of HCI, a field that relates to the development and understanding of computing systems to serve users'

needs. Researchers and practitioners in this field are focused on making universally usable, useful, efficient, and appealing systems to support people in their wide range of activities. My approach is to balance the development of innovative technology that serves people's practical needs. Example systems following this approach that I have built include Cortex-I (1992 embedded computer vision system that sensed licensed plates with custom motor, camera and controller), Audio Augmented Reality (1995 embedded system for sensing a user's location and playing audio suited to that location), Fisheye Menus (2000 software for sensing movement within and selection of linear list of items in a menu), PhotoMesa (2001 software for end users to browse personal photos), DateLens (2002 software for end users to use their mobile devices to efficiently access their calendar information), SlideBar (2005 linear sensor to control scrolling), LaunchTile (2005 "home screen" software for mobile devices to allow users to navigate apps in a zoomable environment), SpaceTree (2001 software for end users to efficiently browse very large hierarchies), ICDL (as I describe above), and StoryKit (a 2009 iPhone app for children to create stories).

10. I built a voice interface system in 2009, described in a 2010 paper², to

² David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson, Multi-modal text entry and selection on a mobile device (2010), In Proceedings of Graphics Interface 2010 (GI '10), Canadian Information Processing Society, CAN, 19–26 (Ex-1031).

support user's rich text entry with multi-modal input on mobile devices to support communications including text messages. As illustrated in this figure from the paper, we enabled users to enter formatted text (bold, colored, etc.) using a touch screen, voice recognition, tilt, and a foot pedal and we measured which approach was most effective.

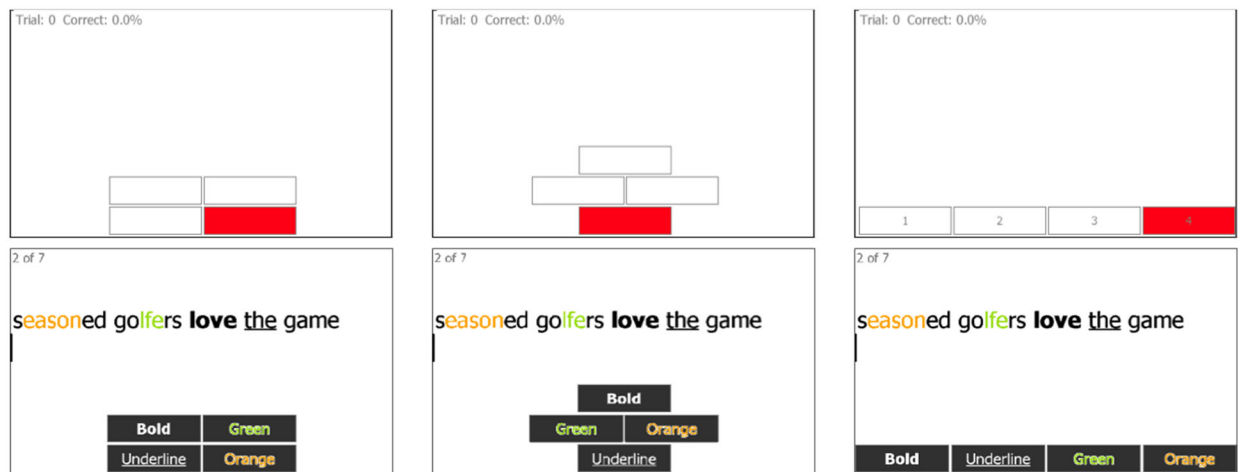


Figure 2. The software interface for the Target Selection (top) and Text Formatting (bottom) tasks. Presented are the Foot (left), Tilt (middle) and Touch/Voice (right) interfaces. The target to select for the Target Selection task is red. The phrase used in the Text Formatting task includes all four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

11. Throughout the 1990s and 2000s, I worked on a range of “zoomable user interfaces,” which are systems that support the multi-scale and spatial organization of and magnification-based navigation among multiple documents or visual objects. I built several different “ZUI” systems over the years, including Pad++, Jazz and Piccolo. In those systems, I used a range of solutions to allow users

to control zooming through the information space.³

12. My work with ZUIs led to my creation of Zumobi in 2006, where I was responsible for investigating new software platforms and developing new user interface designs that provide efficient and engaging interfaces to permit end users to access a wide range of content on mobile platforms (including the iPhone and Android-based devices). For example, I designed and implemented software called “Ziibii,” a “river” of news for iPhone that used a capacitive sensor for controlling linear movement through news, software called “ZoomCanvas,” a zoomable user interface for several iPhone apps, and iPhone apps including “Inside Xbox” for Microsoft and Snow Report for REI.

13. At the ICDL, I have since 2002 been the technical director responsible for the design and implementation of the web site, www.childrenslibrary.org (originally at www.icdlbooks.org). In particular, I have been closely involved in designing the user interface as well as the software architecture for the web site since its inception in 2002. Beginning in the mid-1990s, I have been responsible for the design and implementation of numerous other web sites in addition to the ICDL. For example, I designed and built my own professional web site when I was an Assistant

³ Benjamin B. Bederson & James D. Hollan, Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, UIST ‘94 Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology 17 (1994), DOI: <http://dx.doi.org/10.1145/192426.192435> (Ex-1026).

Professor of Computer Science at the University of New Mexico in 1995 and have continued to design, write the code for, and update both that site (which I moved to the UMD in 1998, currently at <http://www.cs.umd.edu/~bederson/>) as well as numerous project web sites, such as Pad++, <http://www.cs.umd.edu/hcil/pad++/>.

14. I received the Janet Fabri Memorial Award for Outstanding Doctoral Dissertation for my Ph.D. work in robotics and computer vision. My work has been published extensively in more than 160 technical publications, and I have given about 100 invited talks, including 9 keynote lectures. I have won a number of awards including the Brian Shackel Award for “outstanding contribution with international impact in the field of HCI” in 2007, and the Social Impact Award in 2010 from the Association for Computing Machinery’s (“ACM”) Special Interest Group on Computer Human Interaction (“SIGCHI”). ACM is the primary international professional community of computer scientists, and SIGCHI is the primary international professional HCI community. I have been honored by both professional organizations. I am an “ACM Distinguished Scientist,” which “recognizes those ACM members with at least 15 years of professional experience and 5 years of continuous Professional Membership who have achieved significant accomplishments or have made a significant impact on the computing field.” I am a member of the “CHI Academy,” which is described as follows: “The CHI Academy is an honorary group of individuals who have made substantial contributions to the

field of HCI. These are the principal leaders of the field, whose efforts have shaped the disciplines and/or industry, and led the research and/or innovation in human-computer interaction.” The criteria for election to the CHI Academy are: (1) cumulative contributions to the field; (2) impact on the field through development of new research directions and/or innovations; and (3) influence on the work of others. I have received two “Test of Time” awards from IEEE InfoVis for my 2001 and 2002 work on visualizing hierarchies, including those containing digital photos.⁴

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

16. I have designed, programmed, and publicly deployed dozens of user-facing software products that have cumulatively had millions of users. My work is cited by several major companies, including Amazon, Apple, Facebook, Google, and Microsoft. I am a named inventor on 14 U.S. patents and 20 U.S. patent applications.

⁴ <https://ieevis.org/year/2021/info/awards/test-of-time-awards;>
[https://ieevis.org/year/2022/info/awards/test-of-time-awards.](https://ieevis.org/year/2022/info/awards/test-of-time-awards)

The patents are generally directed to user interfaces/experience.

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

III. INFORMATION CONSIDERED

18. In preparation for my Declaration, I have considered the materials I discuss in my Declaration, including, for example, the '030 Patent, the references cited by the '030 Patent, the prosecution history of the '030 Patent (including the references cited therein), various background articles and materials referenced in my Declaration, and the prior art references I identify in my Declaration. In addition, my opinions are further based on my education, training, experience, and knowledge in the relevant field.

IV. RELEVANT LEGAL STANDARDS

19. I am not an attorney and offer no legal opinions. For the purposes of my Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Claim Interpretation

20. I have been informed and understand that in an IPR proceeding, claims are to be interpreted according to the *Phillips* claim construction standard. I

understand that under this standard, the meaning of claim terms is considered from the viewpoint of a hypothetical “person of ordinary skill in the art” (“POSITA”) at the time of the alleged invention. I have been informed and understand that claim construction is a matter of law and that the final claim constructions for this proceeding will be determined by the Patent Trial and Appeal Board (“PTAB”).

21. I provide my opinions regarding terms that may require explicit construction to resolve the grounds I present in my Declaration in Section IX, ¶¶78-85.

B. Perspective of One of Ordinary Skill in the Art

22. I have been informed and understand that a patent is to be understood from the perspective of a hypothetical POSITA. Such an individual is considered to possess normal skills and knowledge in a particular technical field (as opposed to being a genius). I understand that in considering what the claims of a patent require, what was known prior to that patent, what a prior art reference discloses, and whether an invention is obvious or not, one must use the perspective of such a POSITA.

C. Obviousness

23. I have been informed and understand that a patent claim is obvious under 35 U.S.C. §103, and therefore invalid, if the claimed subject matter, as a whole, would have been obvious to a POSITA as of the priority date of the patent based on one or more prior art references and/or the knowledge of a POSITA.

24. I understand that an obviousness analysis must consider (1) the scope and content of the prior art, (2) the differences between the claims and the prior art, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations, if any, of non-obviousness (such as unexpected results, commercial success, long-felt but unmet need, failure of others, copying by others, and skepticism of experts).

25. I understand that a prior art reference may be combined with other references to disclose each element of the invention under 35 U.S.C. § 103. I understand that a reference may also be combined with the knowledge of a POSITA, and that this knowledge may be used to combine multiple references. I further understand that a POSITA is presumed to know the relevant prior art. I understand that the obviousness analysis may take into account the inferences and creative steps that a POSITA would employ.

26. In determining whether a prior art reference would have been combined with other prior art or other information known to a POSITA, I understand that the following principles may be considered:

- a. whether the references to be combined involve non-analogous art;
- b. whether the references to be combined are in different fields of endeavor than the alleged invention in the Patent;
- c. whether the references to be combined are reasonably pertinent to the problems to which the inventions of the Patent are directed;

- d. whether the combination is of familiar elements according to known methods that yields predictable results;
- e. whether a combination involves the substitution of one known element for another that yields predictable results;
- f. whether the combination involves the use of a known technique to improve similar items or methods in the same way that yields predictable results;
- g. whether the combination involves the application of a known technique to a prior art reference that is ready for improvement, to yield predictable results;
- h. whether the combination is “obvious to try”;
- i. whether the combination involves the known work in one field of endeavor prompting variations of it for use in either the same field or a different one based on design incentives or other market forces, where the variations are predictable to a POSITA;
- j. whether there is some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill in the art to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- k. whether the combination requires modifications that render the prior

art unsatisfactory for its intended use;

- l. whether the combination requires modifications that change the principle of operation of the reference;
- m. whether the combination is reasonably expected to be a success; and
- n. whether the combination possesses the requisite degree of predictability at the time the invention was made.

27. I understand that in determining whether a combination of prior art references renders a claim obvious, it is helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in doing so. I understand, however, that a teaching, suggestion, or motivation to combine is not required.

V. LEVEL OF ORDINARY SKILL IN THE ART

28. I am familiar with the level of ordinary skill in the art with respect to the '030 Patent around its filing date. Based on my experience working, teaching, and conducting research in the relevant field, and based on my review of the '030 Patent specification, claims, file history, and prior art, I believe one of ordinary skill in the art around the time of the alleged invention of the '030 Patent would have been someone with at least a bachelor's degree in electrical engineering, computer science, computer engineering, or a related field, and two-three years of experience with multimedia and touch interface technologies, or the equivalent, with additional

education substituting for experience and vice versa.

29. In determining the level of ordinary skill in the art, I considered, for example, the type of problems encountered in the art, prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field.

30. I met or exceeded the knowledge and experience that a POSITA would have had in 2012. In my work as a professor and entrepreneur, I worked with graduate students and others who had the knowledge and experience of a POSITA in 2012, and I am able to render opinions from the perspective of a POSITA based on that knowledge and experience. My opinions concerning the '030 Patent claims and the prior art are from the perspective of a POSITA, as set forth above.

VI. SUMMARY OF MY OPINIONS

31. As I explain below in detail in my Declaration, it is my opinion that Claims 1-14 of the '030 Patent would have been obvious under 35 U.S.C. § 103 on the following grounds:

- **Ground 1:** Claims 1, 3-8, and 10-14 are obvious over Lee'041 (Ex-1014).
- **Ground 2:** Claims 1, 3-8, and 10-14 are obvious over Lee'041 in view of Ohashi (Ex-1017).
- **Ground 3:** Claims 2 and 9 are obvious over Lee'041 in view of Ohashi in further view of Lee'804 (Ex-1015) and D'Souza (Ex-1018).

VII. TECHNOLOGICAL BACKGROUND

32. The '030 Patent is directed to well-known apparatuses and methods for detecting gestures on a user interface, switching recording modes based on detected gestures, and showing different functions and content on the user interface based on detected gestures.

33. To provide background for the element-by-element analysis of the claims to follow, I present an overview of some of the relevant state of the art existing as of the time of the alleged invention relating to gesture detection on user interfaces of electronic devices, switching modes based on detected gestures, and showing different functions and content on user interfaces based on detected gestures. As I will describe below, all of these technologies and techniques for gesture detection and implementing corresponding functions were well-known to those of ordinary skill in the art at the time of the alleged invention claimed in the '030 Patent, and a POSITA would have readily understood the combination of elements of Claims 1-14 to have been obvious.

A. Mobile Applications for Recording Media Data On Electronic Devices and the Claimed Recording Modes Were Known

34. The '030 Patent admits that “as technology developed and electronic devices such as smart phones, tablets, etc., became popular, mobile applications for recording audio data and other media data on such electronic devices gained popularity.” Ex-1001, 1:24-27. Moreover, the '030 Patent admits that, at the time of

filing, “multiple communication modes such as a push-to-talk (PTT) communication mode, a tap to start (TTS) communication mode, a release to stop (RTS) communication mode, a tap to end (TTE) communication mode, etc., were developed for recording audio data.” Ex-1001, 1:27-33.

B. Gesture Recognition Using Touch Interfaces, Including for Recording Media, Was Known

35. Electronic devices with touch-sensitive displays configured to detect user gestures using an application(s) on the device were well known by 2012. Nearly all Graphical User Interfaces (GUIs) are operated with an input device that allows pointing at the screen. While mice are ubiquitous pointing devices for displays that are not touch-sensitive, there have actually been many different kinds of pointing devices through the years. Since the very first GUIs, system designers have continually searched for ways to interact with and control GUIs in ways that would be easier to use and better suited to particular tasks. This includes experimentation with different pointing devices. The very first GUIs supported a range of input devices: even as early as 1963, Sutherland’s SketchPad system⁵ used a “light pen” that allowed a user to select items on the screen by pointing at them. There were also

⁵ Ivan Edward Sutherland, Sketchpad: A man-machine graphical communication system, University of Cambridge Technical Report UCAM-CL-TR-574, ISSN 1476-2986, Number 574 (2003), available at: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf> (Ex-1056).

other kinds of pens (more generally called styli) that could be used to point at touch-sensitive surfaces such as the touch pads described by Lee and Buxton in 1985.⁶

36. In addition to pointing methods that used devices such as pens and mice, researchers also developed a number of more direct pointing methods where a person touched a sensing surface with their fingers. This had been done with both resistive and capacitive sensing techniques by Johnson in 1966.⁷

37. These touch sensing technologies were deployed in a wide variety of research oriented and commercial products. A sample of exemplary systems are depicted in the figure below that show pen and finger-based touch screen and tablet computing devices. There were of course, many more devices and software, but this shows a few devices demonstrating important milestones and announcements.

⁶ S.K. Lee, William Buxton, & K.C. Smith, A multi-touch three-dimensional touch-sensitive tablet, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '85), 21-25 (Apr. 1985), available at: DOI: <https://doi.org/10.1145/317456.317461> (Ex-1057).

⁷ U.S. Patent No. 3,482,241 (Ex-1058).



38. These historical systems are described at the following websites:

- 1979 Apple tablet:

<https://mirrors.apple2.org.za/Apple%20II%20Documentation%20Project/Peripherals/Digitizers/Apple%20Graphics%20Tablet/Photos/Apple%20Graphics%20Tablet%20-%20Front.jpg>

- 1986 Wacom WT-460M:

<https://web.archive.org/web/20111027233900/http://www.wacom.jp/en/company/history/>

- 1991 NCR-3125: <https://oldcomputers.net/ncr-3125.html>

- 1992 Microsoft Windows for PEN Computing:

http://www.pencomputing.com/frames/tablet_pc.html

- 1992 Sun Microsystems Star7: <https://javapapers.com/core-java/java-history/>
- 1993 Apple Newton: <http://oldcomputers.net/apple-newton.html>
- 1993 AT&T EO 440 running GO PenPoint: <http://oldcomputers.net/eo-440.html>
- 1996 Palm Pilot: <http://www.palminfocenter.com/news/8493/pilot-1000retrospective>
- 1996 Microsoft Windows CE: <https://news.microsoft.com/1996/09/16/microsoft-announces-windows-cenewest-member-of-windows-family/>

39. For example, it was well known that electronic devices were configured to translate “the rough finger-based input” of a user “into a precise pointer/cursor position or command for performing the actions desired by the user.” U.S. Patent No. 10,140,301 to Davydov (Ex-1059) at 7:1-12, 10:45-57. Electronic devices could include modules such as a “[c]ontact/motion module” that “detect[s] a gesture input by a user. Different gestures on the touch-sensitive surface have different contact patterns. Thus, a gesture may be detected by detecting a particular contact pattern. For example, detecting a finger tap gesture includes detecting a finger-down event followed by detecting a finger-up (lift off) event at the same position (or substantially the same position) as the finger-down event (e.g., at the position of an icon). As

another example, detecting a finger swipe gesture on the touch-sensitive surface includes detecting a finger-down event followed by detecting one or more finger-dragging events, and subsequently followed by detecting a finger-up (lift off) event.” *Id.*, 13:17-29. Exemplary gestures may also “include a right - to - left swipe gesture ... or a tap gesture.” *Id.*, 26:5-8; *see also* U.S. Patent Publication No. 2013/0019175 to Kotler (Ex-1060) at ¶28 (“touching, swiping, pressing and holding, dragging/sliding or similar” user gestures were disclosed in the prior art and could be used with input mechanisms such as voice commands); U.S. Patent No. 9,767,681 to Hope (“Hope”) (Ex-1061) at 5:65-66 (“User device 12 may have a touch screen that allows device 12 to recognize gestures.”).

1. Press and Hold / Push to Talk

40. The use of a press and hold gesture, including to implement a push to talk recording mode, was well known by 2012.

41. For example, Hope teaches that “[i]f desired, a user may touch a part of the touch screen display 34 and prolong the touch over time. This type of persistent tapping gesture may result in different” functions for the device. Ex-1061 at 15:34-37.

42. In half-duplex systems the channel can bear only one side of a conversation at time, thus in a PTT system, the push-to-talk switch disengages the

receiver and engages the transmitter until the switch is released.⁸ As an example, handheld walkie-talkies, first used in World War II, are legacy half-duplex radio systems requiring PTT switches.⁹ PTT is so ubiquitous that even children playing on toy walkie-talkies can recognize and operate the PTT switch. It has been well known in the art for some time to implement PTT in touch screen devices for recording and capture purposes.

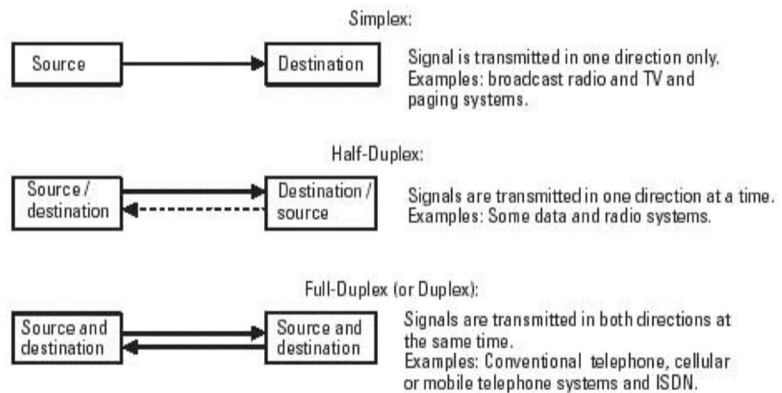


Figure 3.1 Simplex, half-duplex, and full-duplex transmission.

43. For example, U.S. Patent Publication No. 2013/0227456 to Lee ("Lee2013") describes a method of providing capture data such as recording and voice recognition by a mobile device. Ex-1063 at Abstract, ¶80. Lee2013 teaches

⁸ Tarmo Anttalainen, Introduction to Telecommunications Network Engineering, 77-84 (2d ed., 2003), available at: <https://ieeexplore.ieee.org/document/9106127> (Ex-1062).

⁹ Wikipedia, Walkie-talkie, available at: <https://en.wikipedia.org/wiki/Walkie-talkie> (Ex-1081) ("The transceiver is normally in receive mode; when the user wants to talk they must press a 'push-to-talk' (PTT) button that turns off the receiver and turns on the transmitter.").

that if a user inputs a “gesture of pressing a capture key, then external content may be captured when a user presses the capture key for a predetermined time or longer.”

Id. ¶112. The capture key can be “a hardware input” or “a predetermined region on a screen.” *Id.*

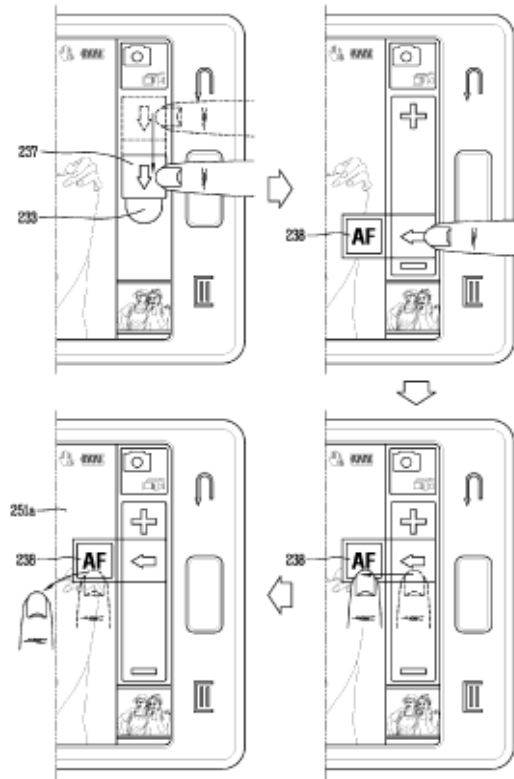
2. Slide and hold

44. The use of slide and hold, swipe and hold, and drag and hold gestures, including to implement a continuous recording mode, were also well known by 2012.

45. Moreover, a POSITA was aware of a number of different ways to implement a “hold” with a slide, swipe, and drag gesture. For example, U.S. Patent Publication No. 2014/0063313 to Choi relates to a mobile device capable of activating prescribed image capture functions based on user gestures on a display unit. Ex-1064 at Abstract. “Referring to FIG. 7A, in a camera mode, a slide button 237 is output to at least one side of the capturing button. The slide button 237 may be moved by a user’s drag. Once the slide button 237 is dragged towards the capturing button 233, a prescribed function is activated.” *Id.* ¶152. “As another example, referring to FIG.7B, a slide bar is output together with a capturing button in a camera mode. If the slide bar is dragged towards the capturing button, the prescribed function may be activated.” *Id.* ¶156. In this example, the user’s constant finger contact with the user interface in combination with the sliding motion would

have been understood by a POSITA to teach a “slide and hold” gesture since the user would sometimes hold their finger at the end position.

FIG. 7A



This is similar to the only example in the '030 Patent that could correspond to the claimed “slide and hold” gesture. Specifically, the '030 Patent states that “the user realizes that the recording will be lengthy, and wishes to continue recording without continually pressing the interface region 301b. The user performs a slide gesture on the GUI 302 to switch from the push-to-talk (PTT) communication mode to the release to record (RTR) communication mode. That is, the user slides his/her finger on the GUI 302 from the interface region 301b to the interface region 301c. The

audio recording application detects the user's slide gesture on the GUI 302 and highlights the interface region 301c as exemplarily illustrated in FIG. 4C. When the user releases his/her finger from the interface region 301c, the audio recording detects the user's release gesture on the GUI 302. In response to the detection of the release gesture, the audio recording application executes the predefined function associated with the interface region 301c where the release gesture was detected. Since the predefined function that was newly associated with the interface region 301c is to continue recording the audio data in the RTR communication mode, the audio recording application executes this predefined function and continues recording the audio data in the RTR communication mode." Ex-1001, 12:39-63.

46. There are also examples in the prior art of a "swipe and hold" gesture that includes a momentary "hold" when the swiping gesture is completed. For example, Hope teaches that "[a] user may perform a swipe and **hold** gesture on touch screen display 34. A swipe and hold gesture may be formed from a combination of tap point 101, swipe motion 102, and hold point 103, as shown in FIG. 10. For example, a user may start by placing their finger at tap point 101. The user may then drag their finger in a swipe motion across touch screen display 34, as indicated by arrow 102. Rather than removing their finger at the end of the gesture, the user may momentarily hold their finger at hold point 103 (e.g., for a second or longer or for a fraction of a second or longer). In one embodiment, the swipe and hold gesture may

continue to generate user input in user device 12 that control media system 14 until the user's finger is removed from hold point 103 of touch screen display 34. In an alternative embodiment in which the swipe and hold gesture is disabled, a swipe and hold gesture may generate the same user input as a swipe gesture." Ex-1061 at 15:65-16:14. "The swipe and hold gesture may generate different user inputs on user device 12 and may subsequently perform different remote control functions on media system 14 based on the direction the swipe motion 102 is performed. As with swipes, swipe and hold gestures may be performed in various directions (e.g., up, down, left, right, or at an angle)." *Id.*, 16:15-20.

47. As another example, the user could slide their finger across the touch screen in the direction that they want the screen to scroll or tap and hold an icon for a few seconds and then drag it where they want to move it. Rick Broida & Dave Johnson, *How to Do Everything Palm® Pre™* (2010) (Ex-1065) at 16, 18.

48. As explained above (and in more detail in my element-by-element analysis below), the '030 Patent's "sliding" example corresponds to the claimed "slide and hold gesture" because it is the only such example provided in the patent. But if the term "slide and hold" is interpreted to require a momentary "hold" when the swiping gesture is completed, the motivation to include such a "hold" when a sliding gesture is completed has been well-documented in the prior art, namely, to ensure that only intentional gestures are implemented and avoid inadvertent

termination and/or transmission. For example, Ken Hinckley at Microsoft in 2005, evaluated techniques for delimiting a selection-action gesture using a pen stylus (instead of a finger) with “a drag-and-hold gesture, which requires holding the pen still for 500ms while dragging.” Ken Hinckley et al., Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in *Scriboli*, CHI 2005 (Apr. 2-7, 2005) (Ex-1066) at 4. This research found that “a 500ms delay between cessation of pen movement and popping up the menu represents a good tradeoff between popping up the menu as rapidly as possible, while not being so short a delay that it leads to excessive accidental activations.” *Id.*

49. This motivation is also taught by WO Publication No. 2011/151501 to Dolenc. Dolenc teaches “forming a hold user interface event in response to a touch input or key press input being held in place for a predetermined time, and using said hold event in forming said gesture information.” Ex-1067 at 2:30-34. Moreover, Dolenc teaches that “[i]f the point of touch or click is moved after the Touch Down user interface event (without lifting the touch), a Move event or state 480 occurs. ... The Move event 480 may be terminated by lifting the touch, and a Release event 460 occurs. The Move event may be terminated also by stopping the move without lifting the touch, in which case a Hold event 470 may occur, if the touch spans a long enough time without moving. A Hold event or state 470 may be generated when a Touch Down or Move event or state continues for a long enough time. The

generation of the Hold event may be done e.g. so that a timer is started at some point in the Touch Down or Move state, and when the timer advances to a large enough value, a Hold event is generated, in case the state is still Touch Down or Move, and the point of touch has not moved significantly.” *Id.*, 16:9-26. Dolenc further teaches that “[a] Hold event or state 470 may be terminated by lifting the touch, causing a Release event 460 to be triggered, or by moving the point of activation, causing a Move event 480 to be triggered. ***The existence of the Hold state or event may bring benefits in addition to just having a Touch Down event in the system, for example by allowing an easier and more reliable detection of gestures.***” *Id.*, 16:26-31.

3. Slide and release

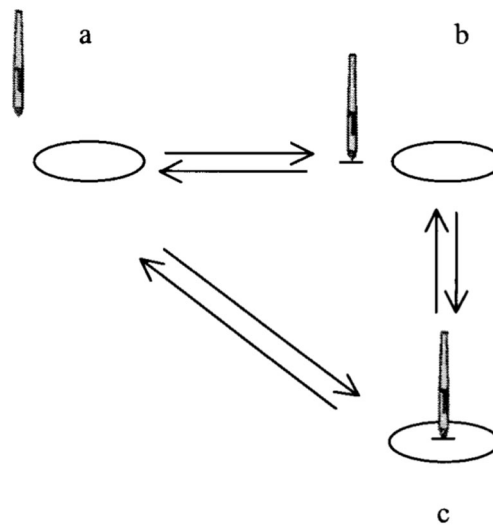
50. The use of slide and release, swipe and release, and drag and release gestures, including to cancel a recording, were well known by 2012.

51. In May 2012, Kay Hofmeester and Jennifer Wolf at Microsoft, defined new gestures for Windows 8 “that worked well for the properties of touch and helped optimize interaction with the system.” Kay Hofmeester & Jennifer Wolfe, Self-Revealing Gestures: Teaching New Touch Interactions in Windows 8, CHI 2012 (May 5-12, 2012) (Ex-1068) at 816. One of those gestures was a “press and hold” gesture.” *Id.*, 822. Another was a new “swipe to select” gesture. The Microsoft team found that “tapping is not the optimal gesture for closing an application due to the ease of accidental activation” in a “vision-based system” and instead “focused on

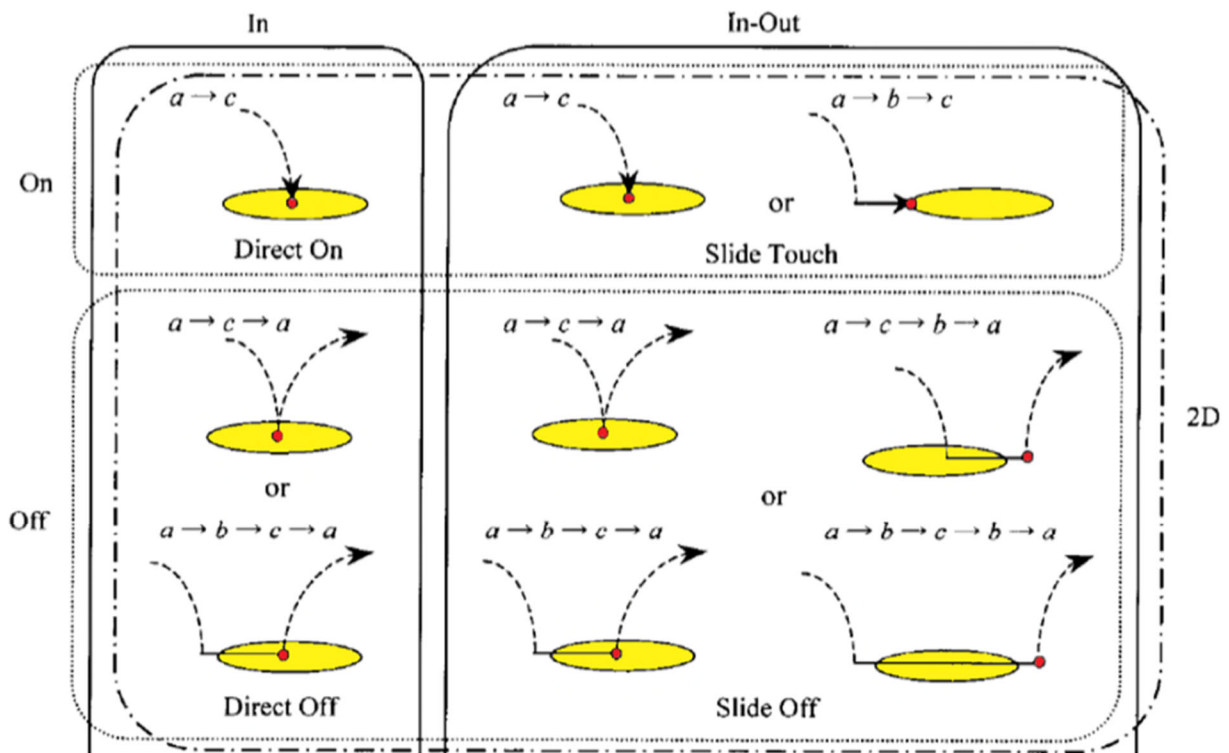
teaching users to use a more deliberate, but equally easy sliding gesture to close applications.” *Id.*, 818.

52. A 2000 article¹⁰ by Ren and Moriya described a number of selection strategies for touch based gesture systems including “Direct Off” where “the target is highlighted only while the pen is touching it. The selection is made at the moment the pen is taken off the target.” (Ex-1069, 390). The following two figures illustrate how various selection strategies can be built by encompassing various basic states of a device on a touch surface. Fig. 1 illustrates a state transition model where the pen is (a) outside/above the screen and target; (b) outside the target, on the screen; and (c) inside the target, on the screen. Fig. 3 illustrates the Direct Off selection strategy.

¹⁰ Xiangshi Ren & Shinju Moriya, Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks, 7 ACM Trans. Comput.-Hum. Interact. 3, 384-416 (Sept. 2000), available at: <https://doi.org/10.1145/355324.355328> (Ex-1069).



Ex-1069, Excerpt of Fig. 1.



Ex-1069, Excerpt of Fig. 3 (emphasis added); *see also* Andrew Sears & Ben Shneiderman, High Precision Touchscreens: Design Strategies and Comparisons

with a Mouse (Jan. 23, 1989), available at:
<https://api.drum.lib.umd.edu/server/api/core/bitstreams/9352f26a-e793-429a-80f8-65f606f41969/content> (Ex-1070).

53. It would have been obvious that a gesture's function is inputted when the user releases their finger from the interface. For example, if a swipe and release gesture corresponds to a cancel recording function, when a user releases their finger from the user interface after a swipe, the recording is canceled.

54. A release is detected when the user's finger is removed from the interface.

55. It was well known in the art that a user could cancel the impending capture of media by dragging (or swiping/sliding) an image element to a recycling bin icon, for example, in the lower right corner of a user interface and releasing their finger from the interface. *See* Seon Joo Kim et al., Brush-and-Drag: A Multi-touch Interface for Photo Triaging, MobileHCI'12 (Sept. 21-24, 2012) ("Kim2012") (Ex-1071) at 5. After this action, the related image element would disappear from the screen, and the associated image would be moved into the recycling bin. *Id.*

56. I note that the '030 Patent does not define what it means to "cancel a recording"—in some instances, it seems to equate it with deleting a recording and in other instances, it seems to distinguish it from the same. For example, the specification states that "[t]he media recording application configured as an audio

recording application allows the user to perform an action, for example, cancel the audio data being recorded, delete the recorded audio data.” Ex-1001, 10:64-65. On the other hand, the background section of the specification states that prior art systems “do not offer a method for the user to cancel the recording once begun ... unless the graphical user interface provides an additional user interaction option, after the recording is completed, that would request the user whether he/she would like to save or cancel the last recording.” Ex-1001, 2:43-46. The most likely meaning of “cancel a recording” in the context of the ’030 Patent is to stop or end the recording. For example, the ’030 Patent states that “the user performs a slide gesture on the GUI 302 ... and [w]hen the user lifts or releases his/her finger from the interface region 301a, the audio recording application detects the user’s release gesture on the GUI 302. In response to detection of the release gesture, the audio recording application executes the predefined function associated with the interface region 301a where the release gesture was detected. Since the predefined function that was newly associated with the interface region 301a is to cancel and delete the recorded audio data which is currently being recorded, the audio recording application executes this predefined function and cancels the recorded audio data. ... Therefore, the audio recording application allows a user to cancel the recording being made by the user ... without being forced to go through another interaction with the GUI 302.” Ex-1001, 11:37-58; *but see id.*, 8:15-20 (distinguishing between

“cancel the recording” and “release to stop”). Regardless of whether canceling a recording means to stop/end the recording or stop/end and delete the recording, it would have been obvious to a POSITA that a swipe and release gesture could be used to “cancel” a recording, because both types of functions (stopping a recording and/or deleting a recording) were well known by 2012 as was the concept of using well-known gestures (such as swipe and release) to implement such defined functions. The motivation for canceling a recording would have been known to a POSITA in all of these scenarios—to avoid transmitting media data to another device or server that the user does not want transmitted.

57. Moreover, there were “eight golden rules of interface design” that were known to a POSITA by 2012, one of them being to “[p]revent errors.” In their fourth edition of “Designing the User Interface” book published in 2004, Ben Shneiderman and Catherine Plaisant explained this golden rule. Ex-1045, 75. In particular, the book teaches POSITA to “[a]s much as possible, design the system such that users cannot make serious errors; for example, grayout menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields. ... If a user makes an error, the interface should detect the error and offer simple, construction, and specific instructions for recovery. For example, users should not have to retype an entire name-address form if they enter an invalid zip code, but rather should be guided to repair only the faulty part. Erroneous actions should leave the system state

unchanged, or the interface should give instructions about restoring the state.” Ex-1045, 75.

4. Tap to end

58. The use of touch to end and tap to end gestures, including to terminate a recording, were well known in the art by 2012. As described above in the Slide to Release section (Section VII.B.3, ¶¶50-57), Ren discloses multiple touch screen based selection gestures including “Direct On” where “Selection is made at the moment the pen makes contact with the bottom of the target area.” Ex-1069 at 391.

59. A POSITA would have understood that a “tap” is essentially a press of short duration. For example, U.S. Patent No. 9,195,330 to Sirpal relates to a multi-display device that includes a screen, touch-sensitive display, configurable area, gesture capture region(s), and camera. Ex-1072 at Abstract. Sirpal teaches that one or more of the buttons on the user interface are capable of supporting different user commands. For instance, “a normal press has a duration commonly of less than about 1 second and resembles a quick tap. A medium press has a duration commonly of 1 second or more but less than about 12 seconds. A long press has a duration commonly of about 12 seconds or more. The function of the buttons is normally specific to the application that is currently in focus on the” display. *Id.*, 9:67-10:9. For example, in a camera or video application, “a normal, medium, or long press can mean ... take photograph or record video.” *Id.*, 10:13-16.

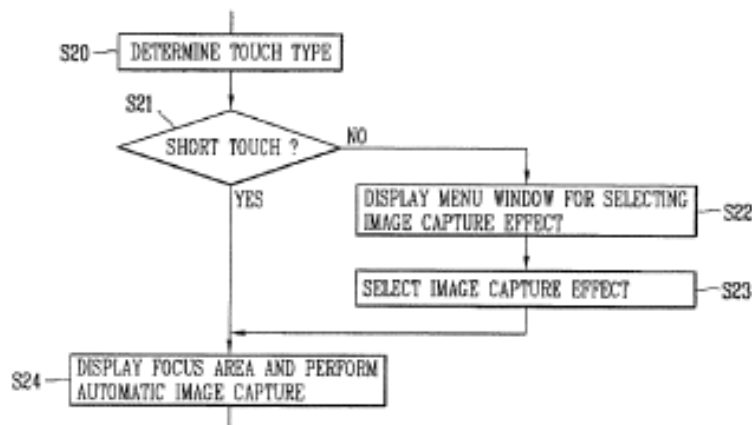
C. Transitioning Between Different Modes Based on Gestures Was Known

60. A controller or processor on an electronic device that transitions between different modes, including recording modes, based on user gestures on a user interface was well known by 2012.

61. For example, U.S. Patent Publication No. 2009/0015703 to Kim (“Kim2009”) relates to a portable terminal having a touch sensing-based image capture function and method that transitions between a general image capture or continuous image capture mode based on detected user gestures. Ex-1073 at Abstract. Kim teaches a controller 180 that automatically performs image capture operation based on detected gestures. *Id.* ¶148. “That is, the controller 180 recognizes the type of the touch inputted by the user, sets an image capture function according to the corresponding touch input, and then displays it on the preview screen display (S30-S32). Here, if there is the **long touch** indicating the general image capture, the controller 180, as shown in FIG.10, displays the separate window on the display module 151.” *Id.* ¶155. “[T]he user may move the corresponding focus by using his finger (**dragging**) or a direction key, and delete the displayed focus by the double-touch. In addition, the user may reduce or enlarge the size of the already displayed focus as well as change its shape by holding an edge of the focus to extend (drag) (S33).” *Id.* ¶156. Moreover, “the controller 180 may control the camera based

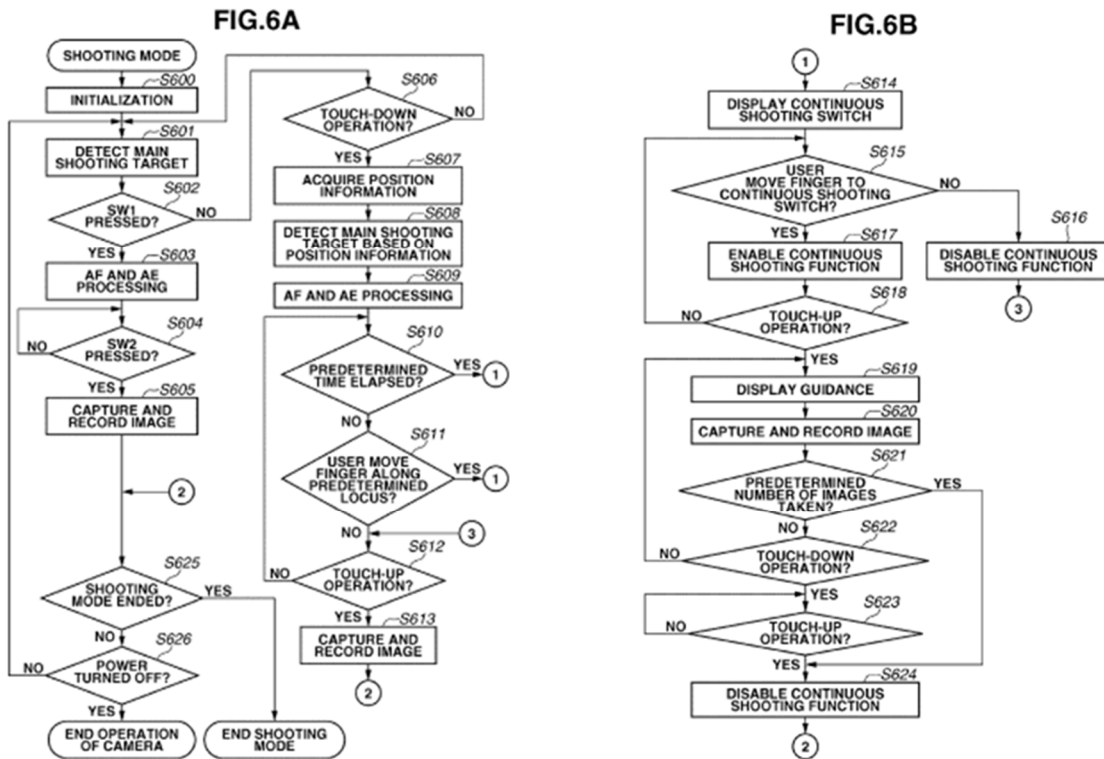
on the determined touch type (i.e., the image capture function set by the user according to the image capture command), thus to perform *either the general image capture or the continuous image capture*. The captured still image is stored in the memory 160 and is transmitted to the external device via the wireless communication unit 110 under the control of the controller 180.” *Id.* ¶157.

FIG. 10



62. Likewise, U.S. Patent Publication No. 2013/0063645 to Aoyama relates to an imaging apparatus, which includes a shooting target detection unit that detects touch operations performed on a touch panel and transitions between single shooting or continuous shooting based on detected user gestures. Ex-1074 at Abstract. Aoyama teaches “an imaging apparatus capable of realizing various types of operations such as an operation for switching *single shooting* and *continuous shooting* and an operation for performing continuous shooting while using a touch panel, a control method therefor, a program, and a recording medium.” *Id.* ¶9. For

example, “the system control device 50 displays the continuous shooting switch, which allows the user to instruct the digital camera 100 to perform continuous shooting, at a position relative to the position where the touch down operation is performed.” *Id.* ¶117. “[T]he system control device 50 determines whether the user moves his/her finger (performs the move operation) to the position of the continuous shooting switch while keeping touching the screen from the touch down operation. ... If the user moves his/her finger (performs the move operation) from the position whether the user performs the touch-down operation to the position of the continuous shooting switch while running his/her finger on the screen (YES in step S615) ... the system control device 50 enables the continuous shooting function.” *Id.* ¶118. “[T]he system control device 50 causes the digital camera 100 to capture an image, and record the captured image data to the external recording medium 91.” *Id.* ¶121. On the other hand, “[i]f the system control device 50 determines that the touch-down operation is performed (YES in step S622), the system control device 50 ends the continuous shooting, and is kept in a standby state until the touch-up operation is performed in step S623. If the touch-up operation is performed (YES in step S623), then in step S624, the system control device 50 disables the continuous shooting function.” *Id.* ¶123.



63. Additionally, U.S. Patent Publication No. 2014/0078371 to Kinoshita teaches an imaging control apparatus and method. Ex-1075 at Abstract. Kinoshita teaches “[i]n the first exemplary embodiment, as a control method of the SW2 button 313, an imaging command to the imaging apparatus 2 is maintained *only during the period that the photographer maintains his/her operation of the SW2 button 313.*” *Id.* ¶79. This corresponds to the “Press and Hold” for PTT recording mode discussed above. “In the second exemplary embodiment, *a lock mechanism is provided for holding a pressed state* of the SW2 button 313. When holding a release for an extended period, such as in bulb imaging, imaging can be continued even if the photographer does not maintain an operation state for an extended period.” *Id.* ¶79.

“An area for locking the SW2 button 313 and holding that state ... represents a display mode that enables the photographer to identify a state held area like that illustrated in FIG. 5B. The photographer can slide the SW2 button 313 to a lock area 316. When the SW2 button 313 overlaps the lock area 316, the SW2 button 313 changes to a pressed state (a locked state) like that illustrated in FIG.5C, and that state is subsequently held.” *Id.* ¶80. This corresponds to the slide and hold for continuous recording mode discussed above. Kinoshita further teaches that “[w]ith the SW2 button 313 in a locked state, when the photographer performs **a release operation**, such as pressing the SW2 button 313, an imaging stop command is transmitted to the imaging apparatus 2, and imaging is stopped.” *Id.* ¶80. Kinoshita’s “CPU 101 determines whether an operation for releasing the lock of the SW2 button 313 has been performed based on whether the SW2 button is not at the lock position, whether the SW2 button has been moved **by a slide operation to its original position**, or whether the SW2 button has been touched again after being released once. If it is determined that the lock has been released ... the processing proceeds to step S213.” *Id.* ¶91. This corresponds to the swipe and release to cancel recording mode discussed above.

D. Transmitting (and Not Transmitting) Media to A Server or Device Based on Gestures Was Known

64. Transmitting (and not transmitting) media to a server or other electronic device based on gestures was well known by 2012.

65. For example, U.S. Patent Publication No. 2010/0177157 to Stephens relates to a video communication system including a kiosk for recording video messages created by a user, storing them in a database, and providing access to the video messages. Ex-1076 at Abstract. “The kiosk further includes a message-recording device for recording a user video message. The video message and user information are uploaded to a database.” *Id.* ¶6. Moreover, “[t]he kiosk 12 may periodically upload the video communications and message data from the memory component to the database 14 via the remote network connection. The kiosk 12 may be configured to upload the video communications and message database on the number of video communications stored on the memory component, set time intervals, or any other parameter or set of parameters.” *Id.* ¶23.

66. A cancel gesture means that no media is transmitted to a server or device. As explained above, regardless of whether to “cancel a recording” means to stop/end the recording or to stop/end and delete the recording, these functions associated with well known gestures would have been known to a POSITA by 2012. It also would have been understood that if a recording is stopped and/or deleted, the

associated media data for the recording would not be transmitted to a server or device because that is the purpose of canceling a recording—to avoid transmitting media data that the user does not want transmitted. For example, if the recording is being streamed from the user’s device to another device or server, a POSITA would have understood that the media transmission stream would stop/end upon the user implementing a cancel gesture and no further media data would be transmitted to a device or server.

67. A POSITA would also have understood that non-streaming media is not transmitted until after a voice recording is completed. If non-streaming media is canceled before the recording is completed, it won’t be transmitted to a device or server. Therefore, if the recorded media data was stored locally and not yet transmitted to another device or server, a POSITA would have understood that a cancel gesture would stop the recording and no media data would be transmitted to the server or other device after the cancel gesture because that is the purpose of a canceling a recording—to avoid transmitting media data that the user does not want transmitted. The term “streaming,” according to the 2002 Microsoft Computer Dictionary, is defined as “On the Internet, the process of delivering information, especially multimedia sound or video, in a steady flow that the recipient can access as the file is being transmitted.” Microsoft Computer Dictionary 499 (5th ed., 2002) (Ex-1077). On the other hand, a POSITA would understand that non-streaming

media is one that is processed on a single computer. So, if a non-streaming voice recording is made, then it would stay on the computer it was being recorded at while it was being recorded. Thus, if it were cancelled before the recording was completed, it would not be transmitted off the device it was being recorded at.

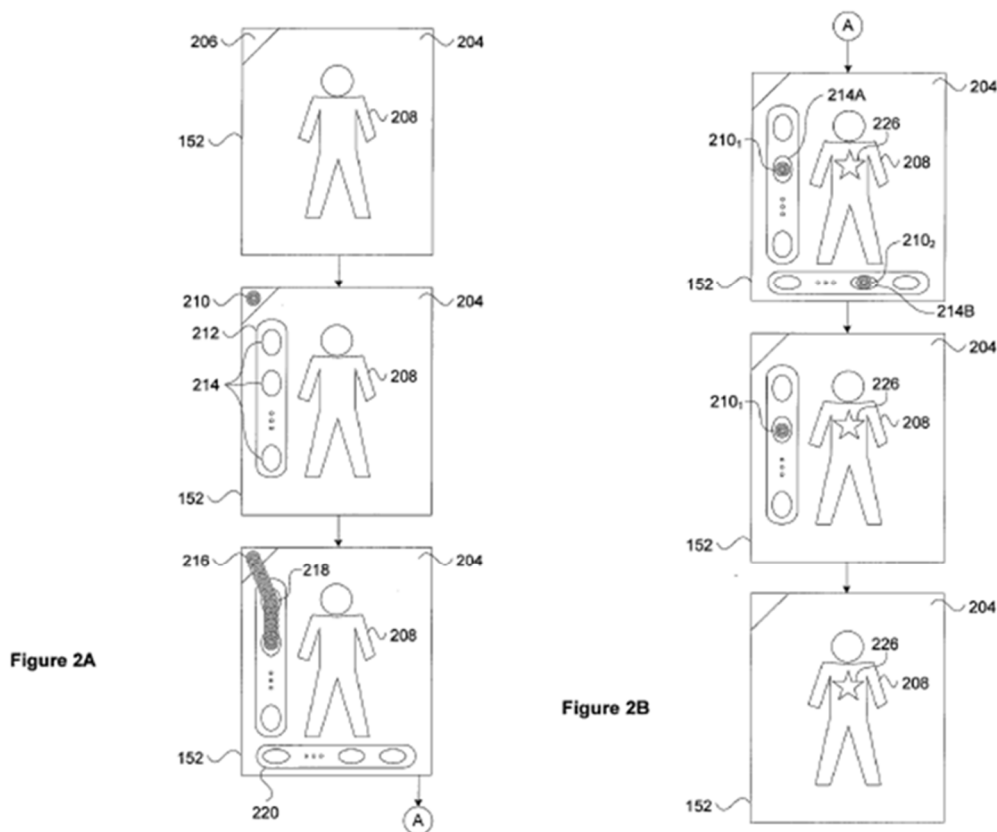
E. Defining Multiple Interface Regions and Dynamically Changing Functions at Interface Regions Based on Gestures Was Known

68. Defining multiple interface regions and dynamically changing functions at interface regions based on gestures was well known by 2012.

69. For example, U.S. Patent Publication No. 2011/0099513 to Ameline describes a computer-implemented method for interacting with a graphical user interface using touch gestures. Ex-1078 at Abstract. Ameline teaches that software “application programs” such as “photo capture applications” “are stored in the system memory. ... The user may interact with the application programs 112 by touching the multi-touch screen 152 (with a stylus, finger, etc.), thereby causing a touch event to occur. The application program 112 may update the application databased on touch events received from the multi-touch Screen 152.” *Id.* ¶23. “The user may establish touch events through press-and-hold interactions 210 and/or press-and-drag interactions 216.” *Id.* ¶24. And “the CPU 130 could be configured to cause the GUI menu strip 212 to be displayed upon determining that the (x,y) coordinate pair of a press-and-hold interaction 210 is within boundaries of the

activation area 206.” *Id.* ¶25. “The GUI menu strip 212 may be configured to display any GUI elements known to those skilled in the art, including text, images, videos, or the like. As shown in FIG. 2A, the GUI menu strip 212 includes one or more GUI menu strip elements 214, with each having their own borders to enable individual selection through a press-and-hold interaction 210 and/or a press-and-drag interaction 216. Further, each of the GUI menu strip elements 214 is configured to, upon selection, cause an associated function to be executed.” *Id.* ¶26. Ameline teaches that “the user, upon invoking the GUI menu strip 212 using a press-and-hold interaction 210, advances his or her finger across the GUI and down the multi-touch screen 152, thereby converting the press-and-hold interaction 210 into a press-and-drag interaction 216 ... the functions associated with the GUI menu strip elements 214 are executed, as described above. In the example shown in FIG. 2A, one GUI menu strip element 214 is associated with a function that causes another GUI menu strip 220 to be displayed.” *Id.* ¶27. Moreover, Ameline teaches that “the user ends his or her press and-drag interaction 216 at GUI menu strip element 214A, whereupon the press-and-drag interaction 216 is converted back into a press-and-hold interaction 210, denoted press and-hold interaction 210. As also shown, the user establishes a second press-and-hold interaction 210, denoted by press and-hold interaction 210, within boundaries of a GUI menu strip item included in the GUI menu strip 220, i.e., GUI menu strip element 214B. In the example illustrated in

FIG.2B, the GUI menu strip item 214B that is selected by the press-and hold interaction 210 is associated with a function that causes a GUI element 226 to appear within the GUI element 208. When the user terminates the press-and-hold interaction 210, the GUI menu strip 220 is no longer displayed in the GUI 204. Similarly, when the user terminates the press-and-hold interaction 210, the GUI menu strip 212 is no longer displayed in the GUI 204.” *Id.* ¶28, Figs. 2A-AB.



70. Ameline further teaches with Figure 5 that “the CPU 130 receives a first multi-touchscreen event, such as a press-and-hold interaction 210 or a press-and-drag interaction 216. At step 504, the CPU 130 determines whether the first multi-touch screen event is within boundaries of a GUI activation area.” *Id.* ¶35. “[T]he

CPU 130 causes a first level GUI that includes a first set of GUI elements to be displayed, such as the GUI menu strip 212 and the GUI elements 214, respectively. At step 510, the CPU 130 receives a selection of one of the GUI elements included in the first set of GUI elements. As described, the selection may be made with a finger, a stylus, or any other user input mechanism.” *Id.* ¶36. “[T]he CPU 130 causes a second level GUI that includes a second set of GUI elements to be displayed. At step 514, the CPU 130 receives a selection of one of the GUI elements included in the second set of GUI elements. Again, as described, the selection may be made with a finger, a stylus, or any other user input mechanism. At step 516, the CPU 130 performs an action associated with the selected GUI element included in the second set of GUI elements.” *Id.* ¶37. “[T]he CPU 130 determines that the first multi-touch screen event has terminated. As previously described, in some embodiments, a user may terminate a multi-touch screen event by eliminating all physical contact with the multi-touchscreen 152.” *Id.* ¶38.

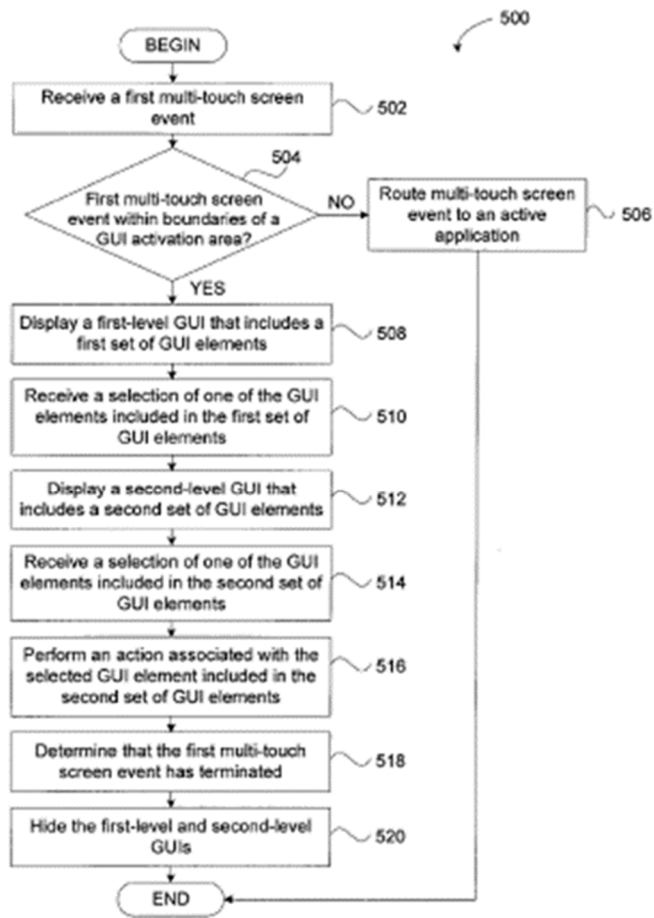


Figure 5

71. Moreover, Lee2013 teaches that “capture data 500 obtained by the capturing unit 120 may be displayed in *a first region of a screen*, a share panel 400 may be displayed in *a second region of the screen*, and a capture panel 300 may be displayed in *a third region of the screen*. In this case, a user may drag the capture data 500 and drop it onto ID information of an external device included in the share panel 400.” Ex-1063 ¶172. “[T]he sensing unit 110 senses the drag and-drop gesture performed from the first region to the second region. Then, the communication unit

160 transmits the capture data 500 to the external device corresponding to the location in the second region where the capture data 500 is dropped.” *Id.* ¶173.

72. Additionally, U.S. Patent Publication No. 2012/0072866 to Imai describes a game apparatus as an information processing apparatus including a CPU that executes a program as to a memo application. Ex-1079 at Abstract. Imai teaches dynamically changing functions associated with defined interface regions based on detected gestures. For example, Imai teaches that “when the icon 440 is long-pushed on the memo creating screen 400 shown in FIG. 30(B), four button images 420, 422, 424, 426 are displayed so as to enclose the icon 440 on the display region 402 of the memo creating screen 400 as shown in FIG. 31(A). That is, a transition to a mode of changing the display position of the icon 440 is made.” Ex-1079 ¶232. “When the icon 440 is dragged in a state that the button images 420-426 are displayed, the icon 440 can be moved in a desired position.” *Id.* ¶233. “Additionally, when the operating region 4302 is touched, the icon 440 is erased. That is, in creating a new memo, not storing the captured image data is selected, or in editing the memo, erasing the captured image data is selected.” *Id.* ¶238.

FIG. 30

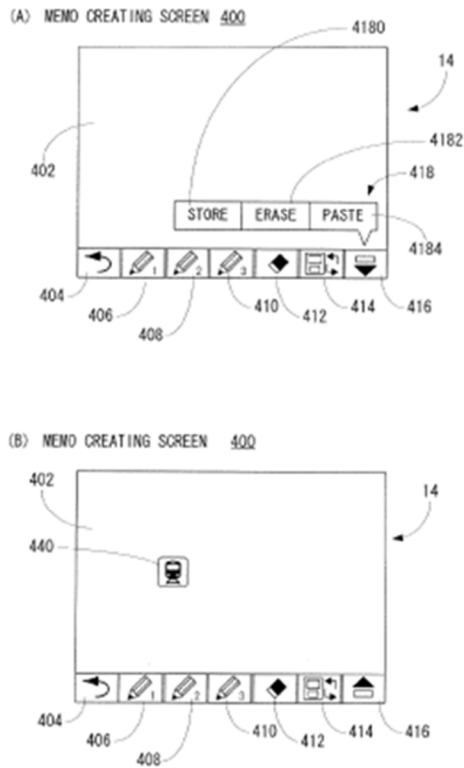
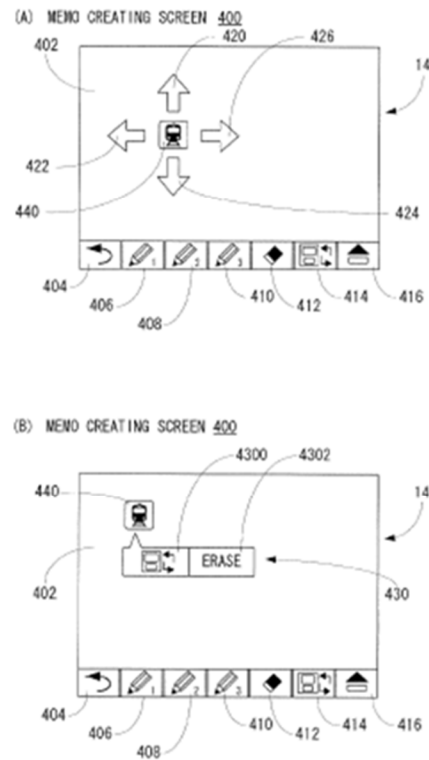
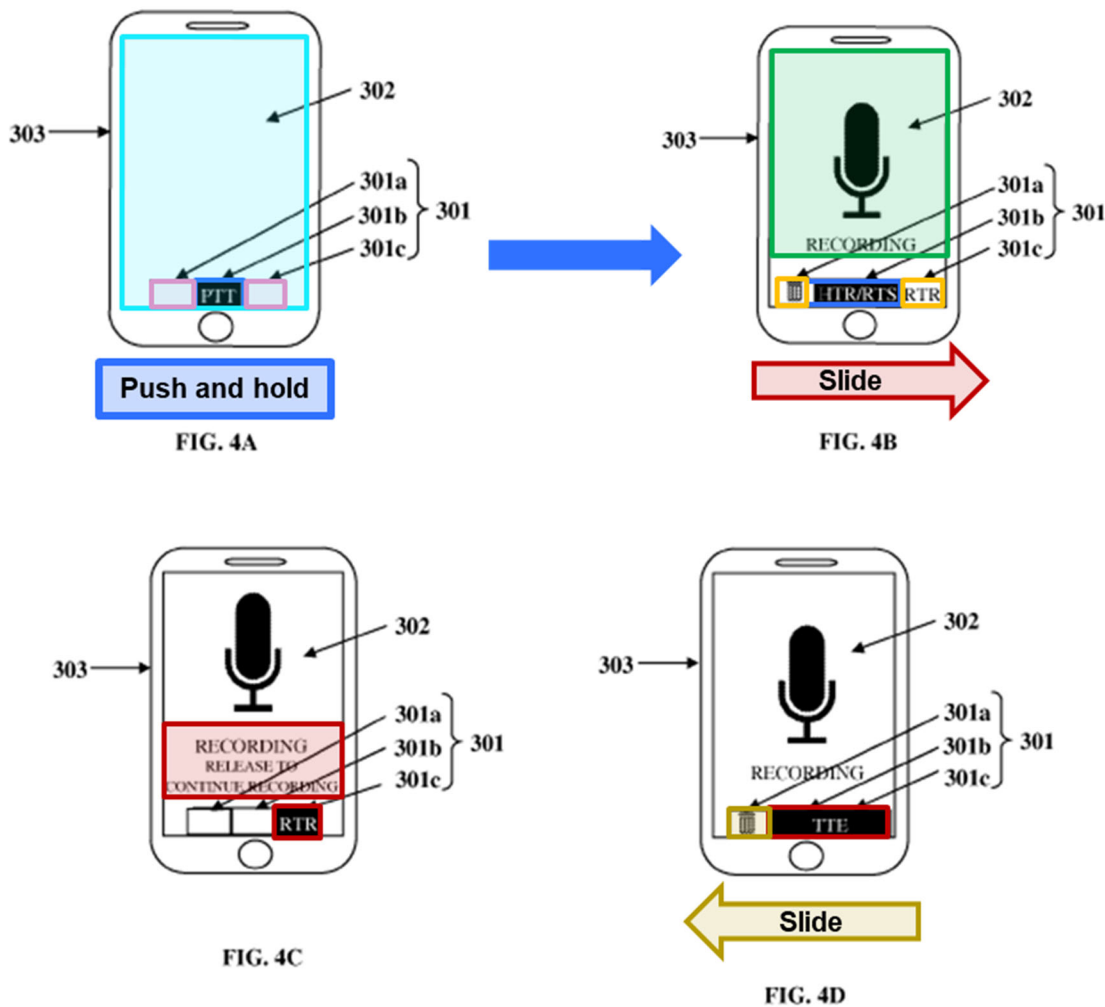


FIG. 31



VIII. THE '030 PATENT AND ITS FILE HISTORY

73. The '030 Patent generally relates to a method for recording media data in a plurality of recording modes based on user gestures that are detected on a display screen of the user's electronic device by a gesture based media recording application, and switching recording modes when different gestures are detected. Ex-1001, Abstract, Cl. 1. Figures 4A-4D (annotated) illustrate the method of the invention:



74. As the figures illustrates, a “user performs a press and hold gesture” on interface region 301b (blue), which the application detects and begins recording the media. Ex-1001, 12:22-29, Fig. 4A. A new predefined “hold to record (HTR)/release to stop (RTS)” function associated with region 301b appears on the interface along with new “cancel and delete” and “release to record (RTR)” functions for regions 301a and 301c, respectively. *Id.*, 12:29-39, Fig. 4B (yellow). If the user wants to continue recording without having to continually press region 301b, they perform a

slide gesture from region 301b to 301c to switch recording modes from Push-to-Talk¹¹ to RTR. *Id.*, 12:39-63, Figs. 4B-4C (red). A new predefined Tap-to-End function is then associated with regions 301b and 301c, which the user can tap to terminate the recording. *Id.*, 12:63-13:2, Fig. 4D (red). The user may also slide and release to cancel the recording. *Id.*, Cl. 2, Fig. 4D (gold).

75. The '030 Patent was filed on October 11, 2013. The claims were rejected in five office actions as anticipated by Cranfill (US2011/0249074) and/or obvious over Cranfill in view of four other references. On April 22, 2019, applicant canceled all pending claims and added new claims, as the examiner suggested in an interview, to “highlight the switching of recording mode from [PTT] recording mode to [RTR] recording mode upon performing the second gesture” in the independent claims. Ex-1004, 452-475. On May 27, 2019, applicant further amended the independent claims, as the examiner suggested in a second interview, to recite “all the transition details between the recording modes and stages within a single modes,” including adding: “wherein said first gesture is a press and hold gesture,” “said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface

¹¹ The '030 Patent and prior art references do not consistently use hyphenation for the recording mode terms. I have hyphenated the recording mode terms in my Declaration for consistency.

regions proximate to said first of said defined interface regions,” and “said recording of said media data continues uninterrupted during and after performance of said second gesture.” Ex-1004, 484-510. Applicant argued the prior art does not teach “a method for recording media data in a plurality of recording modes based on one or more gestures” or switching recording modes “upon detection of the second gesture.” *Id.*, 506-507. The ’030 patent issued on February 4, 2020.

76. For the reasons I discuss below, the prior art teaches the claimed gesture based media recording application—including its well-known configuration of detecting gestures on a user interface, switching recording modes based on detected gestures, and dynamically changing functions associated with defined interface regions based on detected gestures—and renders the challenged claims unpatentable. Accordingly, it is my opinion that Claims 1-14 of the ’030 Patent are invalid as obvious over the prior art.

77. I understand from Counsel that all of the references I discuss in my Declaration are prior art.

IX. CLAIM CONSTRUCTION

78. For the purposes of my Declaration, I do not believe that any term requires explicit construction to resolve the issues I present in my Declaration except for the terms below. Terms not expressly construed should be given their plain and ordinary meaning. I understand from Counsel that various terms were construed or

agreed upon by the parties (Multimodal and Guangdong OPPO Mobile Telecommunications Corp.) in the *Oppo* district court litigation where Patent Owner asserted the '030 Patent against an unrelated party.¹² I apply those constructions in my analysis, except where I note below.

A. “interface regions” (Claims 1, 2, 8, 9)

79. I understand from Counsel that the parties in the *Oppo* litigation agreed that this claim term should be construed to mean “areas provided on the GUI of an electronic device that enable interfacing and interactions with a user of the electronic device.” Ex-1021, 28. I apply this construction for purposes of my Declaration.

B. “gesture based media recording application” (Claims 1-4, 6, 8, 13)

80. I understand from Counsel that the Court in the *Oppo* litigation construed this claim term “to have its plain meaning (apart from the Court’s construction of ‘gesture,’ below).” Ex-1021, 31. I apply this construction for purposes of my Declaration.

C. “gesture” (Claims 1-4, 6, 8-11, and 13)

81. I understand from Counsel that the Court in the *Oppo* litigation construed this claim term to mean “a user action or an actuation by a user performed manually, for example, by use of a finger, a thumb, a hand, etc., on the GUI, or by using an input device, for example, a physical button, a computer mouse, a pointing

¹² I understand from Counsel that the *Oppo* litigation was settled (Ex-1046).

device, a light pen, a touch pad, a touch sensitive display device, a track ball, a pointing stick, any device capable of sensing a tactile input, etc.” Ex-1021, 33. I apply this construction for purposes of my Declaration.

D. “detect[ing] a [first/second/third] gesture” (Claims 1-3 and 8-10)

82. I understand from Counsel that the Court in the *Oppo* litigation construed these claim terms to have their “[p]lain meaning (apart from the Court’s construction of ‘gesture,’ above).” Ex-1021, 33. I apply this construction for purposes of my Declaration.

E. “interface definition module” (Claim 8)

83. I understand from Counsel that the Court in the *Oppo* litigation found that this claim term is subject to 35 U.S.C. §112, ¶6 and should be given the following construction:

Function: “defining a plurality of interface regions on said graphical user interface of said electronic device, and dynamically changing functions associated with said one or more defined interface regions based on said detected first gesture on said one of said defined interface regions.”

Structure: “a software component of the gesture based media recording application that is (a) configured to define multiple interface regions on the graphical user interface and associate a predefined function with each of the defined user interface regions and (b) configured to dynamically change the

functionality of the interface regions to a different predefined function; and equivalents thereof.” Ex-1021, 40-41.

I apply this construction for purposes of my Declaration.

F. “detection module” (Claims 8-11)

84. I understand from Counsel that the Court in the *Oppo* litigation found that this claim term is subject to 35 U.S.C. §112, ¶6 and should be given the following construction:

Function: “detecting a first gesture from among multiple gestures on a first of said defined interface regions, and detecting a second gesture from among said multiple gestures.”

Structure: “a software component of the gesture based media recording application configured to detect one or more physical touch or other gesture on a defined region of a graphical user interface; and equivalents thereof.” Ex-1021, 43.

I apply this construction for purposes of my Declaration.

G. “action management module” (Claim 8)

85. I understand from Counsel that the Court in the *Oppo* litigation found that this claim term is subject to 35 U.S.C. §112, ¶6 and should be given the

following construction:

Function: “starting said recording of media data in a push-to-talk (PTT) recording mode, on said detection of said first gesture on said first of said defined interface regions.”

Structure: “a software component of the gesture based media recording application configured to perform a predefined action based on the detection of a gesture on a defined region of a graphical user interface.” Ex-1021, 45.

I apply this construction for purposes of my Declaration.

X. OVERVIEW OF THE APPLIED PRIOR ART REFERENCES

A. Lee’041 (Ex-1014)

86. Lee’041, titled “Tap-to-End Feature for Unlocking a Virtual Slide-to-Lock Feature in Push-to-Talk Communication Applications,” was filed on July 3, 2012 and issued as US8,812,041 on August 19, 2014. Ex-1014, Cover. I understand from Counsel that Lee’041 is §102(e) prior art.

87. Lee’041 teaches a Push-to-Talk (PTT) application running on the user’s communication (i.e., mobile) device that detects gestures such as Tap-to-End (TTE) and Slide-to-Lock on the device’s display screen and changes recording modes based on the detected gestures. Ex-1014, Abstract, 1:18-21. Figures 2A-2C illustrate Lee’041’s method:

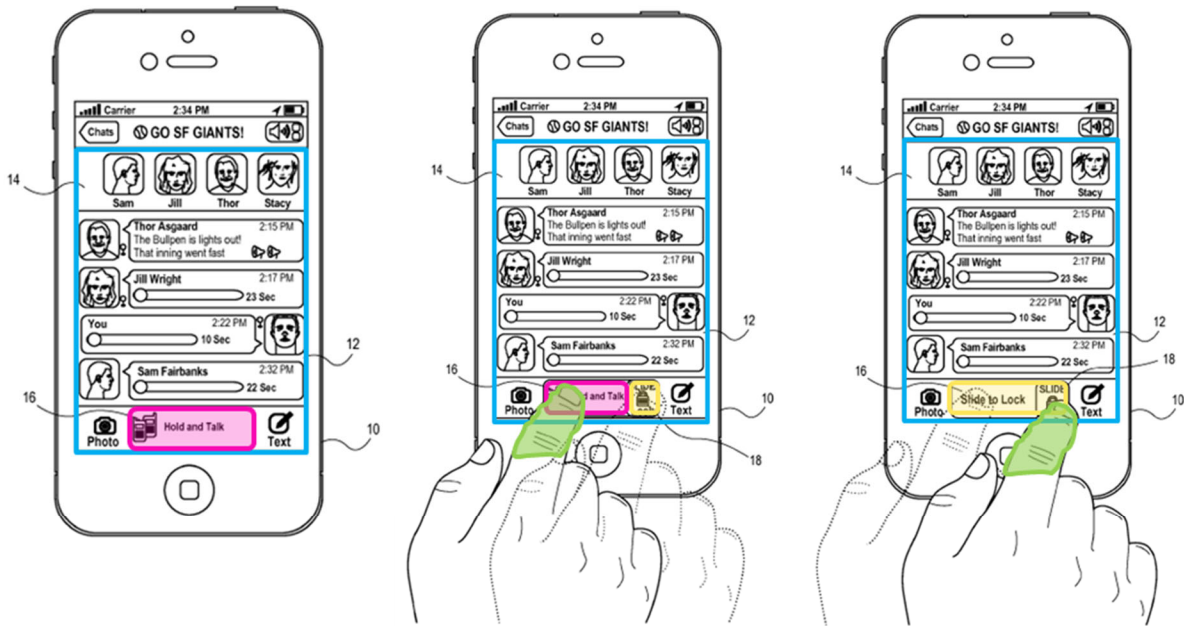


FIG. 2A

FIG. 2B

FIG. 2C

88. As the figures illustrate, the user simply maintains “finger contact with the ‘Hold-and-Talk’”¹³ region 16 “for the duration of the message” to record in a PTT mode. *Id.*, 3:67-4:2, Fig. 2A (pink). When the “Hold-and-Talk” interface region 16 is touched by the user, a “Slide-to-Lock” feature appears in region 18 (Fig. 2B (yellow)). *Ex-1014*, 3:30-35. The user activates the “Slide-to-Lock” feature by “sliding a finger from left to right” over region 16 until region 18 is contacted. *Id.*, 3:37-42. Detection of the sliding motion causes the displayed text in region 16 to change from “Hold-and-Talk” to “Slide-to-Lock” (Fig. 2C (yellow)). *Id.*, 3:42-44.

¹³ Lee’041 does not consistently use “Hold-and-Talk” and “Hold-and-Talk.” Figures 2A-2B depict this feature as “Hold-and-Talk.” For consistency purposes, I use “Hold-and-Talk” in my Declaration when discussing both terms.

Activation of the Slide-to-Lock feature results in “activation of a PTT locking function” on the device, which allows the user to “simply speak into the microphone, or generate other types of media,” without having to contact or hold region 16 or 18. *Id.*, 3:45-51. When the user removes their finger following the slide gesture, “a ‘Tap-to-End’ region 90 appears” as shown in Figure 6C (orange). *Id.*, 5:33-34. Tapping region 90 terminates the message and the device “stops transmitting media.” *Id.*, 5:29-36.

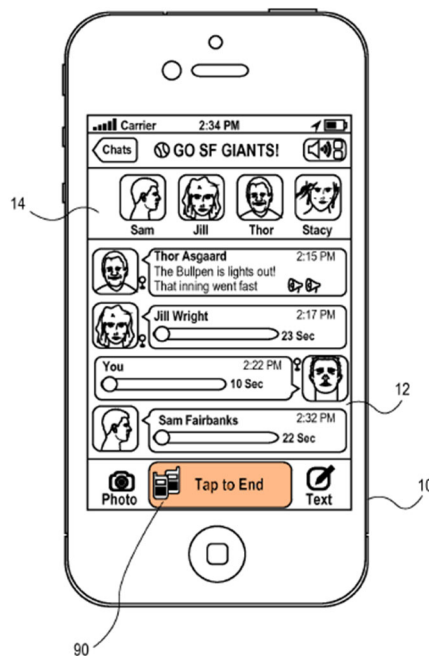


FIG. 6C

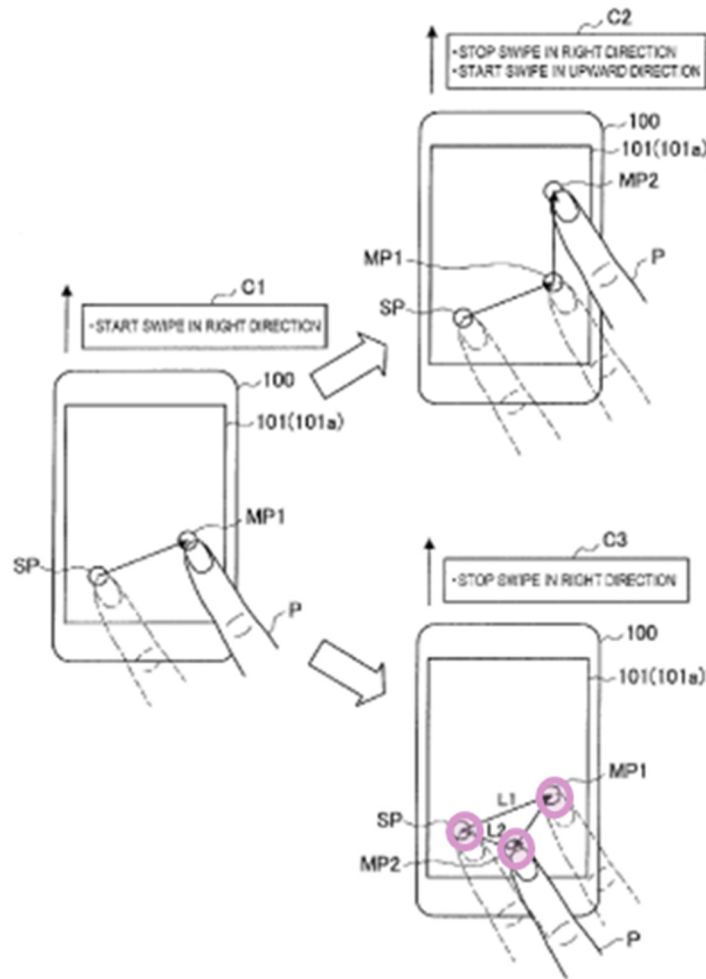
B. Ohashi (Ex-1017)

89. Ohashi, titled “Remote Control Apparatus, Remote Control System, Remote Control Method, and Program,” was filed on May 16, 2012 and published

as US2012/0229410 on September 13, 2012. Ex-1017, Cover. I understand from Counsel that Ohashi is §102(e) prior art.

90. Ohashi teaches detecting a “swipe and hold” gesture on a display panel of an electronic device “in which a finger or the like touches an arbitrary point in the contact detecting region of the touch panel, and the finger or the like is swiped while touching the touch panel and then is held on the touch panel while touching the touch panel.” Ex-1017 ¶3. Ohashi teaches that “[a] continuous control such as a control from the contact starting point SP to a first swipe point MP1, to a second swipe point MP2 ... is done in the swipe and hold control while maintaining the contact.” *Id.* ¶34, Fig. 1 (annotated). Ohashi transmits “a control command according to the detected result” of the swipe and hold gesture “to the electronic device.” *Id.* ¶2. For example, in response to the swipe and hold, “the played content can be ... fast-forwarded/rewound and the displayed content can be ... switched.” *Id.* ¶104.

FIG.1



C. Lee'804 (Ex-1015)

91. Lee'804, titled "Virtual Slide-to-Lock Feature for Push-to-Talk Communication Applications," was filed on May 4, 2012 and issued as US8,948,804 on February 3, 2015. Ex-1015, Cover. I understand from Counsel that Lee'804 is §102(e) prior art.

92. Lee'804 is related to and incorporated by reference in Lee'041 (Ex-1014, 1:9-13), and contains all of the Lee'041 disclosures regarding Figures 2A-2C

I describe above (*see* Section X.A, ¶¶86-88). Ex-1015, 3:19-49. As with Lee’041, when the “Hold-and-Talk” interface region 16 is touched by the user, a “Slide-to-Lock” feature appears in region 18 (Fig. 2B (yellow)). Ex-1015, 3:27-29. The user activates the “Slide-to-Lock” feature by “sliding a finger from left to right” over region 16 until region 18 is contacted. *Id.*, 3:31-34. Detection of the sliding motion causes the displayed text in region 16 to change from “Hold-and-Talk” to “Slide-to-Lock” (Fig. 2C (yellow)). *Id.*, 3:34-38. Activation of the Slide-to-Lock feature results in “activation of a PTT locking function” on the device, which allows the user to “simply speak into the microphone, or generate other types of media,” without having to contact or hold region 16 or 18. *Id.*, 3:39-45. When the PTT locking function is deactivated, the message is terminated. Lee’804 teaches a message module that generates “a message containing media when the PTT locking function is activated using the Slide-to-Lock feature” and “terminates the message when the PTT locking function is de-activated,” as shown in Figure 2D. Ex-1015, Cls. 16-17.

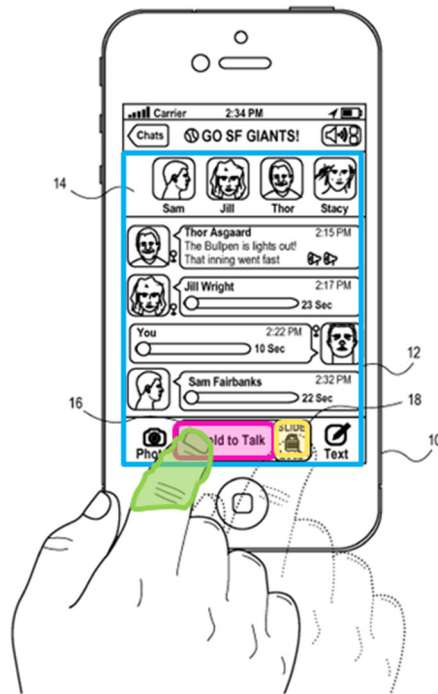


FIG. 2D

D. D’Souza (Ex-1018)

93. D’Souza, titled “Camera Focus and Shutter Control,” was filed on January 5, 2011 and published as US2012/0014684 on January 19, 2012. Ex-1018, Cover. I understand from Counsel that D’Souza is §102(e) prior art.

94. D’Souza teaches a method for canceling an impending capture of media data, such as a photograph, using a swipe and release gesture. Ex-1018 ¶29, Cl. 11. D’Souza teaches that “the taking of a photograph can be canceled,” for example, “when the [user’s] finger is moved to another portion of the touchscreen and then lifted from the touch screen.” Ex-1018 ¶13. For example, if the user decides that they do not want to take a picture, the user slides their “finger away from the shutter

button 300 such that contact is no longer made with any portion of the shutter button 300 but such that contact is maintained with the touch screen 200 throughout the sliding motion” and “[w]hen the user then lifts his or her finger from the touch screen 200, a photograph would not be taken and the autofocus and/or any other photography-related functions that had been activated by the pressing of the finger to the shutter button 300 would be deactivated.” *Id.* ¶17, Fig. 4 (annotated).

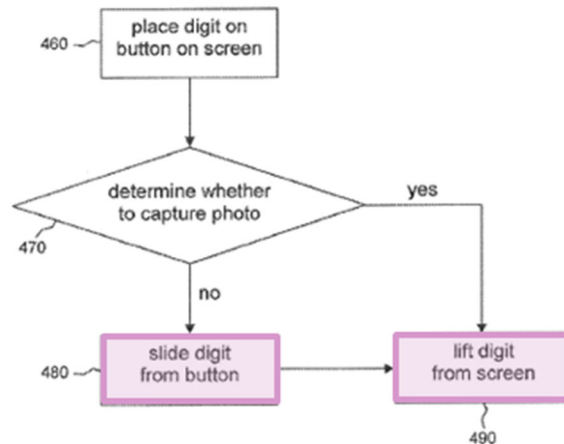


Figure 4

XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

95. Claims 1-14 of the '030 Patent are substantially similar. I discuss like claim elements together in the below element-by-element analysis. *See* Ex-1020 (summary table). To the extent that there are meaningful differences, I individually identify and discuss such differences. As I explain below, the challenged claims are disclosed and/or taught by the prior art.

A. Ground 1: Claims 1, 3-8, and 10-14 are obvious over Lee'041

1. Independent Claims 1 and 8

a. Elements 1[Pre]/8[Pre]: A computer implemented method for recording media data, comprising:

96. In my opinion, Lee'041 teaches the preambles of Claims 1 and 8. Lee'041 teaches “computer code embedded in a non-transitory computer readable medium” that runs on a communication device (Ex-1014, Cl. 1), such as a mobile phone. *Id.*, 6:1-5. The code is configured to display on the screen of the communication device a PTT application with features including Tap-to-End (TTE) and Slide-to-Lock. *Id.*, Cl. 1. As I explain in my technical background section (*see* Section VII.B.1, ¶¶40-43), a POSITA would have understood “Push-to-Talk” or “PTT” as a well-known term of art that dates back to early radio transmission systems over half-duplex communication channels in which the user pushed a switch

on the microphone to begin speaking and then typically concluded with the word “over,” before releasing the switch. *See, e.g.*, Ex-1062.

97. Lee’041’s application allows a user to “speak into the microphone, or generate other types of media,” without having to contact or hold the screen. Ex-1014, 3:49-51. Media includes “voice, video, ... or any other type of streaming or non-streaming media.” *Id.*, 5:57-60.

b. Element 8[A]: a processor;

Element 8[B(i)]: a non-transitory computer readable storage medium communicatively coupled to said processor,

98. In my opinion, Lee’041 teaches Elements 8[A]/8[B(i)], for the same reasons I explain in the previous section. Lee’041 teaches “computer code embedded in a non-transitory computer readable medium” that runs on a communication device (Ex-1014, Cl. 1), such as a mobile phone. *Id.*, 6:1-5. Moreover, Lee’041 teaches a controller¹⁴ and a memory “for storing the PTT application” that is run on the device. *Id.*, 4:65-67, Fig. 5. Furthermore, Lee’041 explains that the Slide-to-Lock detection unit 78 monitors and detects sliding motion “[d]uring execution of the PTT application.” *Id.*, 5:1-4. A POSITA would thus have understood that the communication device would include a processor communicatively coupled to the

¹⁴ The ’030 Patent uses the term microprocessor to include microcontroller (Ex-1001, 16:39-62), and a POSITA would have understood them to be functionally the same in this context.

non-transitory computer readable storage medium to access the memory contents because execution of applications on mobile phones in 2012 could only have occurred on some kind of processor such as a microprocessor and/or microcontroller. Additionally, Lee'041 gives an Apple iPhone as a non-limiting example (*id.*, 5:61-6:1) of the kind of device its invention would run on, and Apple iPhones in 2012 executed applications stored in memory on a microprocessor. For example, a June 7, 2010 Apple announcement describing the iPhone 4 described it as including an “A4 processor.”¹⁵

- c. **Element 1[A]: providing a gesture based media recording application executable by at least one processor configured to manage a graphical user interface of an electronic device for said recording of media data in one or more recording modes on said electronic device;**

Element 8[B(ii)]: said non-transitory computer readable storage medium configured to store a gesture based media recording application,

Element 8[B(iii)]: wherein said gesture based media recording application, when executed by said processor, manages a graphical user interface of an electronic device for said recording of media data;

Element 8[B(iv)]: said gesture based media recording application comprising:

¹⁵ Newsroom, Apple Presents iPhone 4 (June 7, 2010), available at: <https://www.apple.com/newsroom/2010/06/07Apple-Presents-iPhone-4/> (Ex-1080).

99. In my opinion, Lee'041 teaches Elements 1[A]/8[B(ii)]-[B(iv)]. As I describe above, Lee'041's non-transitory computer readable storage medium has embedded computer code that is configured to display certain features on the screen of the communication device running a PTT application (8[B(ii)]). Ex-1014, Cl. 1. Moreover, Lee'041's device includes a memory "for storing the PTT application." *Id.*, 4:65-67.

100. Lee'041 teaches a gesture based media recording application executable by at least one processor configured to manage a graphical user interface of an electronic device for the recording of media data in one or more recording modes on the electronic device (1[A] and 8[B(iii)-(iv)]). Lee'041's PTT application includes a "Slide-to-Lock" feature that is "activated by the user sliding a finger [(Fig. 2B (green))] from left to right over the region 16" on the touch-sensitive display screen of the user's mobile device (pink), "until the Slide-to-Lock feature 18 is contacted" (yellow). *Id.*, 3:37-40. When this feature is activated, the "text displayed in the region 16" of the mobile device display screen "changes from 'Hold-and-Talk' [first recording mode] to 'Slide-to-Lock.'" *Id.*, 3:41-44, Fig. 2C (yellow).

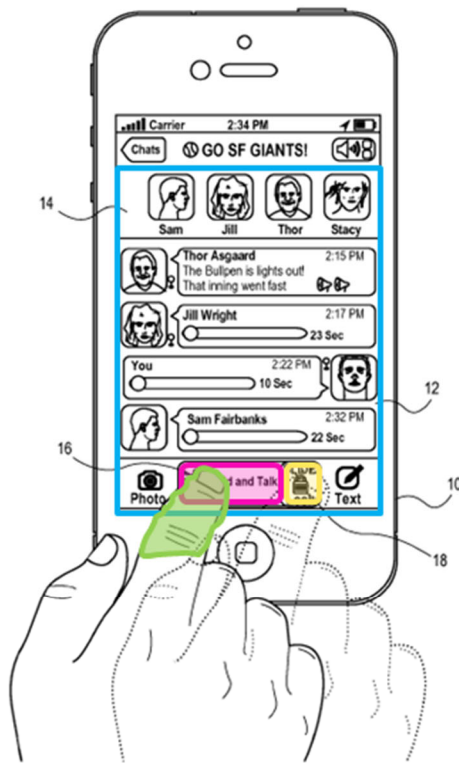


FIG. 2B

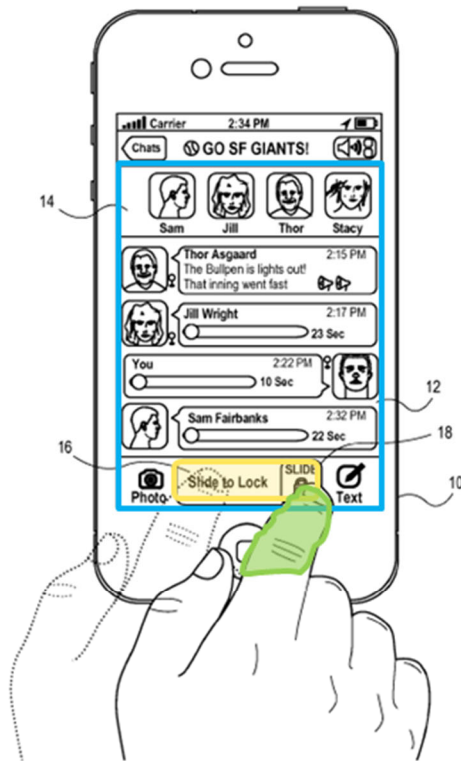


FIG. 2C

101. A recording mode called a “PTT locking function” (second recording mode) is then implemented, which allows the user to “simply speak into the microphone, or generate other types of media,” such as voice or video messages, without having to contact or hold region 16 and/or 18 while creating a message. *Id.*, 3:45-51, 5:57-60.

102. Lee’041 further teaches that a Slide-to-Lock detection unit and controller “operate in cooperation with the touch-sensitive display screen 12 in a well-known manner.” *Id.*, 5:17-19. For example, the detection unit monitors and detects the user’s sliding motion on the display screen when the Slide-to-Lock

feature is activated or deactivated and notifies the controller to activate or deactivate the PTT locking function, respectively. *Id.*, 5:1-7, 5:17-19.

103. Accordingly, a POSITA would have understood that Lee'041's PTT application is a gesture based media recording application that manages a graphical user interface of an electronic device for the recording of media data in one or more recording modes on the electronic device, because it monitors and detects gestures on the display screen such as a sliding motion by the user's finger and activates or deactivates certain media recording modes on the mobile device, such as a PTT locking function, in response to the detected gestures. For instance, when the PTT application detects a sliding motion by the user's finger, it manages the display screen by activating the PTT locking function to allow the user to record a voice message without having to contact or hold a region of the display screen. A POSITA would have understood that the PTT application is executable by at least one processor because Lee'041 teaches that its detection unit and controller "may be implemented in hardware, software, firmware, or any combination thereof" (Ex-1014, 5:17-19) and all mobile phones and computers must have at least a CPU to execute stored computer code or program instructions. Execution of applications on mobile phones in 2012 could only have occurred on some kind of processor such as a microprocessor and/or microcontroller. Additionally, Lee'041 gives an Apple iPhone as a non-limiting example (*id.*, 5:61-6:1) of the kind of device its invention

would run on, and Apple iPhones in 2012 executed applications stored in memory on a microprocessor. For example, a June 7, 2010 Apple iPhone 4 announcement described it as including an “A4 processor.” Ex-1080.

- d. Element 1[B]: defining one or more interface regions on said graphical user interface of said electronic device by said gesture based media recording application;**

Element 8[C]: an interface definition module configured to define a plurality of interface regions on said graphical user interface of said electronic device;

104. In my opinion, Lee’041 teaches Elements 1[B]/8[C]. Lee’041 teaches that its gesture based media recording application defines one or more interface regions on the graphical user interface of the electronic device. Lee’041 teaches that its PTT application defines one or more areas on the touch-sensitive display screen of the mobile device, for example, regions 16 and 18, that enable interfacing and interactions with a user of the mobile device. For example, Lee’041 teaches that a “Slide-to-Lock” feature 18 (Fig. 2B (yellow)) appears on the display screen of the user’s mobile device when the Hold-and-Talk region 16 (pink) is contacted by the user (green). Ex-1014, 3:33-36, Figs. 2A-2B. The Slide-to-Lock feature 18 may be “activated by the user sliding a finger from left to right over the region 16, until the Slide-to-Lock feature 18 is contacted.” *Id.*, 3:37-40. This feature allows a user to “speak into the microphone, or generate other types of media,” without having to contact or hold the screen. *Id.*, 3:49-51.

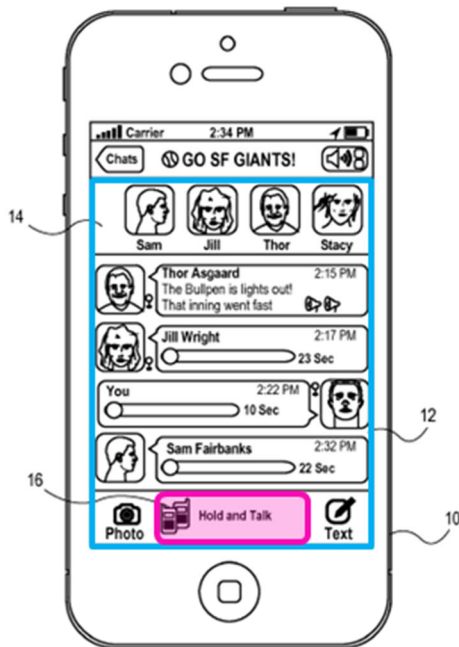


FIG. 2A

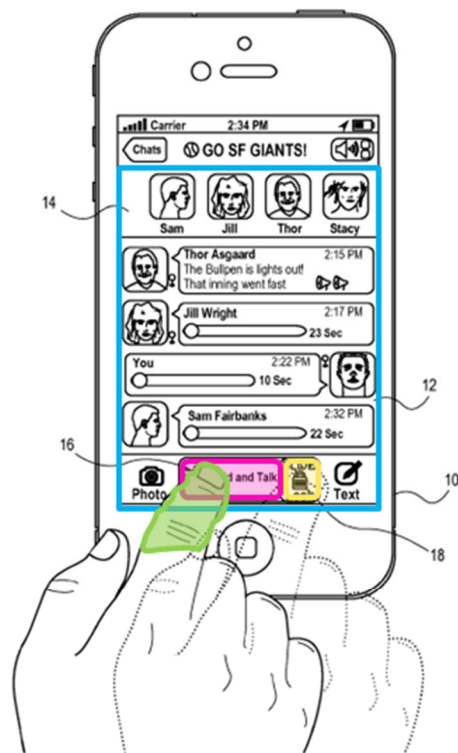


FIG. 2B

105. Thus, similar to the '030 Patent, Lee'041 teaches an interface region (16) that allows the user to Hold-and-Talk and an interface region (18) that enables a Slide-to-Lock feature. Ex-1001, 12:29-39 (interface region 301b enables a “hold to record (HTR)/release to stop (RTS)” function and “a slide gesture” to interface region 301c enables a “release to record (RTR)” function).

106. A POSITA would have understood that Lee'041's PTT application comprises an interface definition module or “a software component” that is “configured to define multiple interface regions on the graphical user interface and associate a predefined function with each of the defined user interface regions”

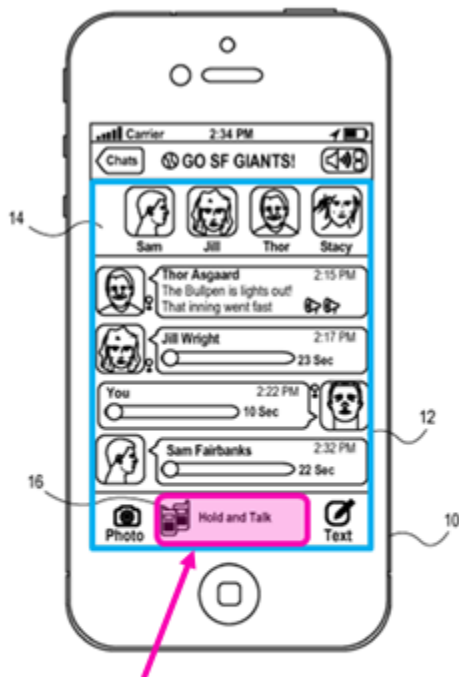
because its display screen includes interface regions 16 and 18 that are associated with predefined functions of Hold-and-Talk (PTT recording mode) and Slide-to-Lock (RTR recording mode), respectively. Ex-1014, 5:17-21 (Slide-to-Lock detection unit and controller “operate in cooperation with the touch-sensitive display screen” and “may be implemented in ... software”).

- e. **Element 1[C]: detecting a first gesture from among multiple gestures on a first of said defined interface regions, by said gesture based media recording application, wherein said first gesture is a press and hold gesture;**

Element 8[D]: a detection module configured to detect a first gesture from among multiple gestures on a first of said defined interface regions, wherein said first gesture is a press and hold gesture;

107. In my opinion, Lee’041 teaches Elements 1[C]/8[D]. Lee’041 teaches that its gesture based media recording application detects a first gesture from among multiple gestures on a first of the defined interface regions, wherein the first gesture is a press and hold gesture. Lee’041 teaches that its PTT application detects a first user action performed manually by the user’s finger on interface region 16 of the touch-sensitive display screen, wherein the first user action is a press and hold gesture (a first gesture). For example, Lee’041 teaches that a “Slide-to-Lock” feature 18 (Fig. 2B (yellow)) appears on the display screen of the user’s mobile device when the Hold-and-Talk region 16 (pink) is contacted by the user (green). Ex-1014, 3:33-36, Figs. 2A-2B (annotated). To perform a press and hold gesture, the user “can

simply maintain finger contact with the ‘Hold-and-Talk’ region 16, or a similar PTT feature, for the duration of the message.” *Id.*, 3:67-4:2. This is one of multiple gestures taught by Lee’041. *Id.*, Abstract (describing Tap-to-End and Slide-to-Lock gestures).



First defined
interface region
(Hold and Talk 16)

FIG. 2A

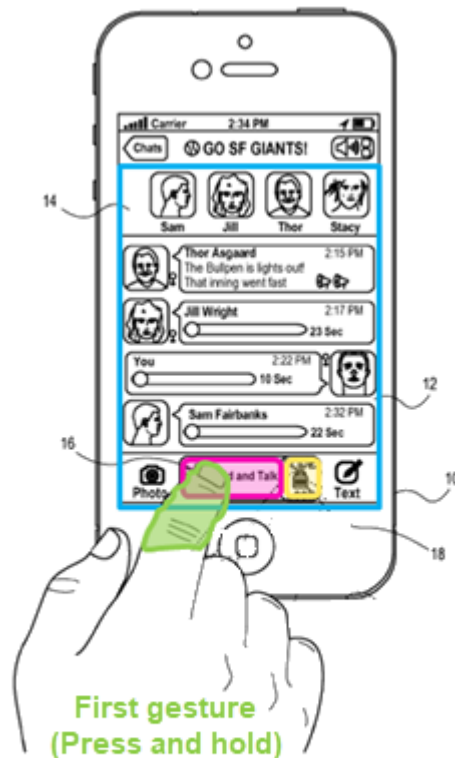


FIG. 2B

108. A POSITA would have understood that Lee’041’s PTT application comprises a detection module or “a software component” that is “configured to detect one or more physical touch or other gesture on a defined region of a graphical user interface” because its controller “operate[s] in cooperation with the touch-sensitive display screen” to detect a user’s press and hold gesture on interface region

16 of the display screen and “may be implemented in ... software.” Ex-1014, 5:17-21.

f. Element 1[D]: starting said recording of media data in a push to talk (PTT) recording mode, by said gesture based media recording application, on said detection of said first gesture on said first of said defined interface regions;

Element 8[E]: an action management module configured to start said recording of media data in a push to talk (PTT) recording mode, on said detection of said first gesture on said first of said defined interface regions;

109. In my opinion, Lee’041 teaches Elements 1[D]/8[E]. Lee’041 teaches that its gesture based media recording application starts the recording of media data in a PTT recording mode on the detection of the first gesture on the first of the defined interface regions. Lee’041 teaches that “a virtual PTT region 16, labeled ‘Hold-and-Talk,’ is provided” on the touch-sensitive display screen of the user’s mobile device. Ex-1014, 3:30-32, Fig. 2B (pink). To record media, the user can simply contact and “maintain finger contact with the ‘Hold-and-Talk’” (a first gesture) on interface region 16 on the display screen (first of the defined interface regions), “or a similar PTT feature, for the duration of the message” to record media in this recording mode. Ex-1014, 3:67-4:2.

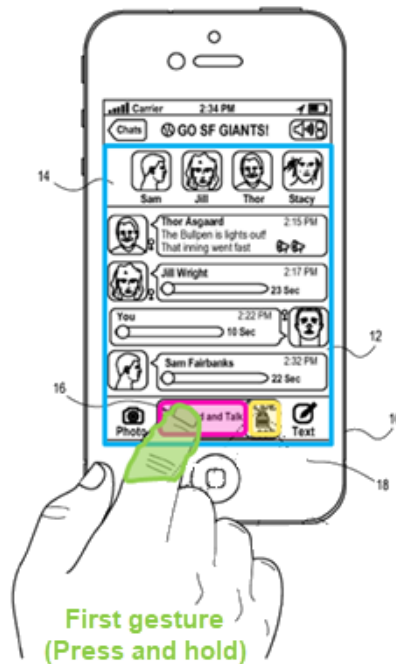


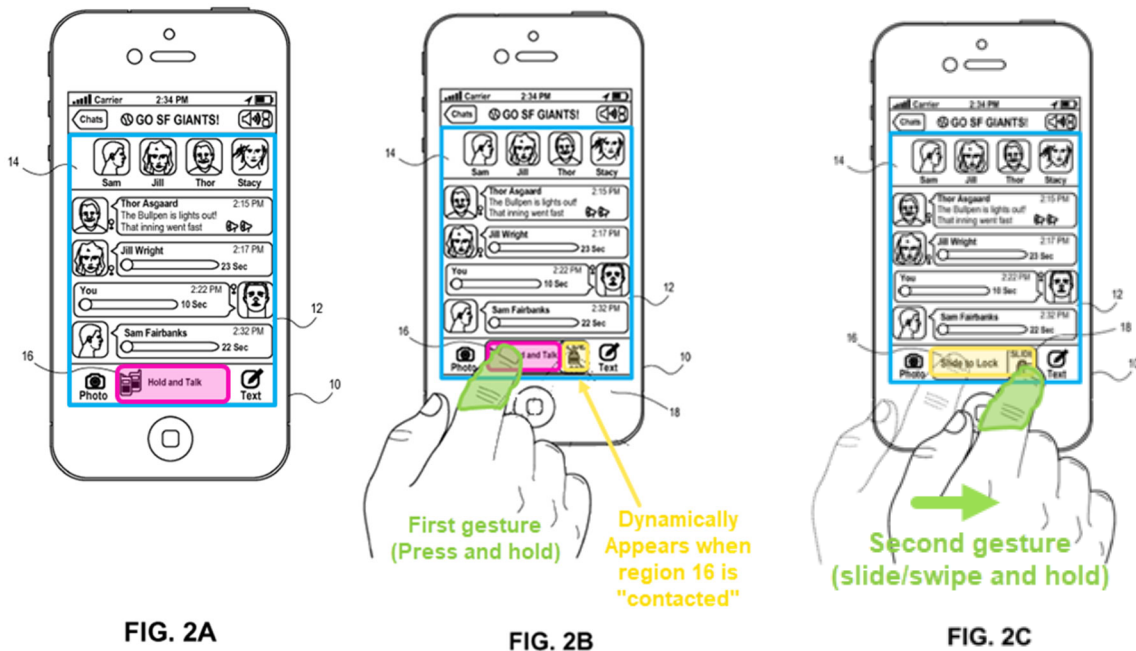
FIG. 2B

110. A POSITA would have understood that Lee'041's PTT application comprises an action management module or "a software component" that is "configured to perform a predefined action based on the detection of a gesture on a defined region of a graphical user interface," because its controller "operate[s] in cooperation with the touch-sensitive display screen" to detect a user's gesture on interface region 16 of the display screen and perform a predefined Hold-and-Talk action based on the detected gesture. Ex-1014, 5:17-21. Further, the detection unit and controller "may be implemented in ... software." As I explain in my technical background section, press and hold gestures and PTT recording modes were also known in the art by 2012. *See* Sections VII.A-VII.B.1, ¶¶34-43.

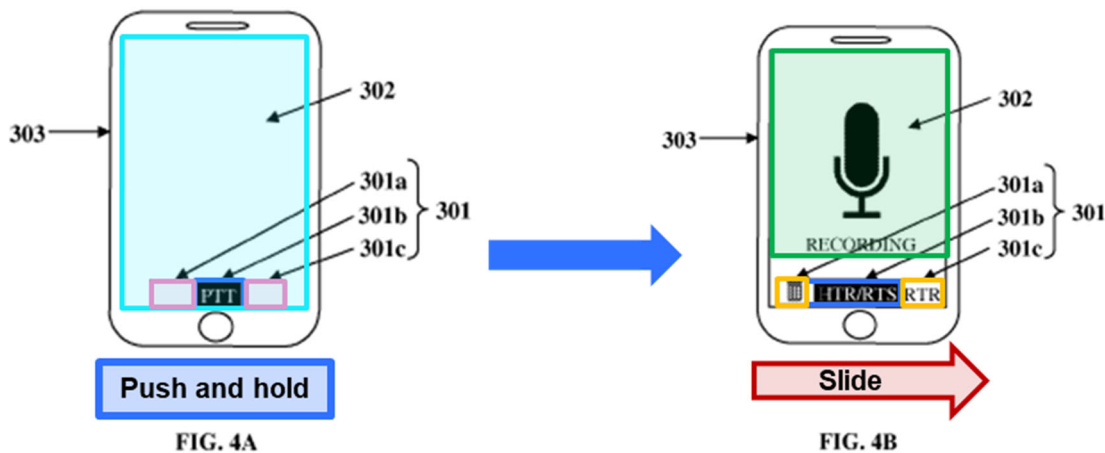
- g. Element 1[E]: dynamically changing, by said gesture based media recording application, functions associated with said one or more defined interface regions based on said detected first gesture on said one of said defined interface regions; and**

Element 8[F]: said interface definition module configured to dynamically change functions associated with said one or more defined interface regions based on said detected first gesture on said one of said defined interface regions; and

111. In my opinion, Lee'041 teaches Elements 1[E]/8[F]. Lee'041 teaches that its gesture based media recording application dynamically changes functions associated with one or more defined interface regions based on the detected first gesture on the defined interface region. Lee'041 teaches that “a virtual PTT region 16, labeled ‘Hold-and-Talk,’ is provided” on the touch-sensitive display screen of the user’s mobile device. Ex-1014, 3:30-32, Fig. 2A (pink). When the Hold-and-Talk region 16 is pressed and held by the user’s finger (Fig. 2B (green)) (first gesture), a “Slide-to-Lock” feature 18 dynamically appears on the display screen (yellow) where a part of region 16 was previously. Ex-1014, 3:33-36. Moreover, when a sliding motion by the user’s finger is detected by the PTT application to interface region 18, “the text displayed in the region 16 changes from ‘Hold-and-Talk’ to ‘Slide-to-Lock.’” *Id.*, 3:40-44, Fig. 2C (yellow).



112. Lee'041's PTT application dynamically changes functions associated with defined interface regions 16 and 18 based on the detected first "Hold-and-Talk" gesture in the same way as the '030 Patent by changing the function of interface region 16 where the user places their finger on it to initiate a recording and adding interface region 18 to which the user may swipe their finger to continuously record. Ex-1001, 12:22-34 (when press and hold gesture is detected, the application "associates new predefined functions to" interface regions 301a, 301b, and 301c, such as "interface region 301b" for "hold to record (HTR)/release to stop (RTS)"), Figs. 4A-4B (301a changes from PTT in Figure 4A to HTR/RTS in Figure 4B).



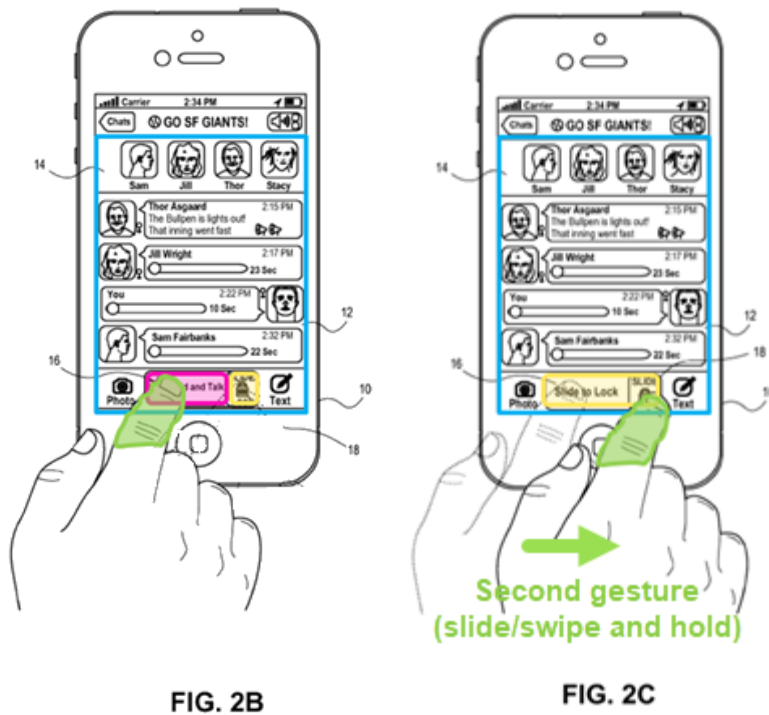
113. A POSITA would have understood that Lee’041’s PTT application comprises an interface definition module or “a software component” that is “configured to dynamically change the functionality of the interface regions to a different predefined function,” as shown in Figures 2A-2C, because the text displayed on the GUI changes from a predefined “Hold-and-Talk” function to a “Slide-to-Lock” function for interface region 16 based on detection of a user’s finger touch or swiping motion. Ex-1014, 5:17-21 (Slide-to-Lock detection unit and controller “operate in cooperation with the touch-sensitive display screen” and “may be implemented in ... software”). As I explain in my technical background section, defining a plurality of interface regions and dynamically changing functions associated with the interface regions based on gestures were also known concepts in the art by 2012. Section VII.E, ¶¶68-72.

- h. Element 1[F(i)]: detecting a second gesture from among said multiple gestures, by said gesture based media recording application, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and**

Element 8[G(i)]: said detection module further configured to detect a second gesture from among said multiple gestures, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and

114. In my opinion, Lee'041 teaches Elements 1[F(i)]/8[G(i)]. Lee'041 teaches that the gesture based media recording application detects a second gesture from among multiple gestures, wherein the second gesture is a slide and hold gesture from a first defined interface region to a second defined interface region proximate to the first defined interface region. Lee'041 teaches that its PTT application detects a second user action performed manually by the user sliding their finger from interface region 16 to interface region 18 of the touch-sensitive display screen while holding their finger on the display screen. In particular, Lee'041's PTT application includes a "Slide-to-Lock" feature that is "activated by the user sliding a finger from left to right over the region 16" on the touch-sensitive display screen of the user's mobile device, "until the Slide-to-Lock feature 18 is contacted" (a second gesture). Ex-1014, 3:37-40, Figs. 2B-2C (annotated). The second interface region 18 is

proximate to the first interface region 16. When this feature is activated, a recording mode called a “PTT locking function” is implemented, which allows the user to “simply speak into the microphone, or generate other types of media,” such as voice or video messages, without having to contact or hold region 16 and/or 18 while creating a message. *Id.*, 3:45-51, 5:57-60.



115. The '030 Patent does not define “hold” in the context of the “slide and hold” and “swipe and hold” gestures claimed. However, Lee'041's sliding motion gesture is implemented in the same manner as the only gesture in the '030 Patent to which the claimed “slide and hold” gesture could possibly refer, where the user slides their finger from interface region 301b to interface region 301c and when the user releases their finger from interface region 301c, the application continues

recording. Ex-1001, 12:39-63. A POSITA would have understood that in both Lee'041 and the '030 Patent, a user's sliding motion from one interface region to another while maintaining contact with the interface at the end of the motion changes the recording mode to allow recording to continue to record without the user having to continue contact. Thus, a POSITA would have understood the terms "slide/swipe and hold" with the same level of disclosure as the '030 Patent to encompass a user holding their finger on the interface briefly after performing the swipe or slide motion. Lee'041 alone thus teaches this limitation.

116. As I explain in my technical background section (*see* Section VII.B.2, ¶¶44-49), a "slide and hold" gesture was known in the art by 2012 as was the POSITA's motivation to implement a hold when completing a slide gesture. For example, Dolenc teaches "forming a hold user interface event in response to a touch input or key press input being held in place for a predetermined time, and using said hold event in forming said gesture information." Ex-1067 at 2:30-34. Moreover, Dolenc teaches that "[i]f the point of touch or click is moved after the Touch Down user interface event (without lifting the touch), a Move event or state 480 occurs. ... The Move event 480 may be terminated by lifting the touch, and a Release event 460 occurs. The Move event may be terminated also by stopping the move without lifting the touch, in which case a Hold event 470 may occur, if the touch spans a long enough time without moving. A Hold event or state 470 may be generated when a

Touch Down or Move event or state continues for a long enough time. The generation of the Hold event may be done e.g. so that a timer is started at some point in the Touch Down or Move state, and when the timer advances to a large enough value, a Hold event is generated, in case the state is still Touch Down or Move, and the point of touch has not moved significantly.” *Id.*, 16:9-26. Dolenc further teaches that “[a] Hold event or state 470 may be terminated by lifting the touch, causing a Release event 460 to be triggered, or by moving the point of activation, causing a Move event 480 to be triggered. ***The existence of the Hold state or event may bring benefits in addition to just having a Touch Down event in the system, for example by allowing an easier and more reliable detection of gestures.***” *Id.*, 16:26-31. Thus, it would have been obvious to a POSITA to include a brief hold at the end of the slide/swipe to ensure the slide/swipe was intended and would have formed a natural ending to the gesture. As I also explain in my technical background section, transitioning between different recording modes based on user gestures was also known in the art by 2012. Section VII.C, ¶¶60-63.

117. A POSITA would also have understood that Lee’041’s PTT application comprises a detection module or “a software component” that is “configured to detect one or more physical touch or other gesture on a defined region of a graphical user interface” because its Slide-to-Lock detection unit and controller “operate in cooperation with the touch-sensitive display screen” to detect a user’s slide and hold

action from interface region 16 to 18 of the display screen and “may be implemented in ... software.” Ex-1014, 5:17-21.

- i. **Element 1[F(ii)]: wherein said recording of said media data continues uninterrupted during and after performance of said second gesture.**

Element 8[G(ii)]: wherein said recording of said media data continues uninterrupted during and after performance of said second gesture.

118. In my opinion, Lee’041 teaches Elements 1[F(ii)]/8[G(ii)]. Lee’041 teaches that the recording of the media data continues uninterrupted during and after performance of the second gesture. Lee’041 teaches that “[w]hen the PTT locking function is activated, the user ... no longer needs to select or hold the region 16 and/or 18 while creating a message”; instead, “the user can simply speak into the microphone, or generate other types of media,” without having to contact or hold interface region 16 and/or 18. Ex-1014, 3:46-51; *see also id.*, 5:31-33 (“after the Slide-to-Lock feature 18 is implemented ... the user no longer has to contact the region 18 to generate and transmit voice media”).

2. Dependent Claims 3-7 and 10-14

- a. **Element 3[A]: The computer implemented method of claim 1, further comprising: detecting a third gesture from among said multiple gestures on said second of said defined interface regions, by said gesture based media recording application, wherein said third gesture is a release gesture from said second of said interface regions, and**

Element 10[A]: The computer implemented system of claim 8, further comprising: said detection module further configured to detect a third gesture from among said multiple gestures on said second of said defined interface regions, by said gesture based media recording application, wherein said third gesture is a release gesture from said second of said interface regions, and

119. In my opinion, Lee'041 teaches Elements 3[A]/10[A]. Lee'041 teaches that the gesture based media recording application detects a third gesture from among multiple gestures on the second defined interface region, wherein the third gesture is a release gesture from the second interface region. Lee'041 teaches that its PTT application detects a third user action performed manually after the user slides their finger from interface region 16 to interface region 18 of the touch-sensitive display screen and releases it from interface region 18. Lee'041 teaches that the "Slide-to-Lock" feature "is activated by the user sliding a finger from left to right over the region 16, until the Slide-to-Lock feature 18 is contacted," which "signifies the selection of the Slide-to-Lock feature 18." Ex-1014, 3:37-42, Fig. 2C (annotated). Once the user releases their finger from region 18 (the second region), the "Tap-to-End" display option (orange) appears. *Id.*, 5:31-34, Fig. 6C (annotated). Display of "Tap-to-End" on the interface is evidence that the release gesture was detected. I also discuss this "release" concept in my technical background section

(see Section VII.B.3, ¶¶50-57), where I explain that a release is detected when the user's finger is removed from the interface.



FIG. 2C

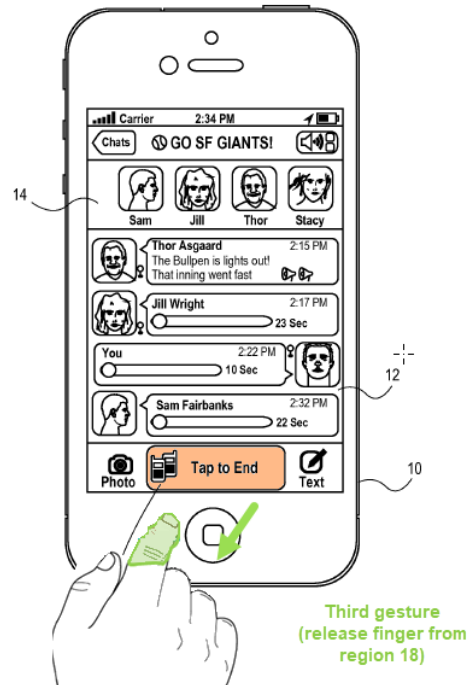


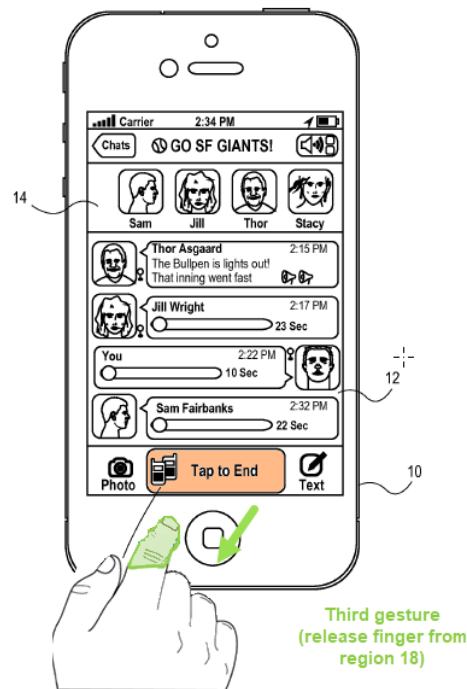
FIG. 6C

120. A POSITA would also have understood that Lee'041's PTT application comprises a detection module or "a software component" that is "configured to detect one or more physical touch or other gesture on a defined region of a graphical user interface" because its Slide-to-Lock detection unit and controller "operate in cooperation with the touch-sensitive display screen" to detect a user's finger release on interface region 18 of the display screen and "may be implemented in ... software." Ex-1014, 5:17-21.

- b. Element 3[B]: switching said recording of said media data to a tap to end (TTE) mode, wherein said recording of said media data continues uninterrupted during and after performance of said third gesture.**

Element 10[B]: switching said recording of said media data to a tap to end (TTE) mode, wherein said recording of said media data continues uninterrupted during and after performance of said third gesture.

121. In my opinion, Lee'041 teaches Elements 3[B]/10[B]. Lee'041 teaches switching the recording of the media data to a Tap-to-End (TTE) mode, wherein recording of the media data continues uninterrupted during and after performance of the third gesture. Lee'041 teaches that “[w]hen the PTT locking function is activated,” the user no longer has to hold region 16 or 18 when creating a message. Instead, “the user can simply speak into the microphone,” or generate and transmit other types of media, without having to contact or hold region 16 and/or 18. Ex-1014, 3:45-51. When the user removes their finger, “a ‘Tap-to-End’ region 90 appears,” as shown in Figure 6C (orange). *Id.*, 5:33-34. Tapping region 90 terminates the message and the device “stops transmitting media.” *Id.*, 5:29-36.



- c. **Element 4[A]:** The computer implemented method of claim 3, further comprising: detecting a fourth gesture from among said multiple gestures on said one of said defined interface regions, by said gesture based media recording application, wherein said fourth gesture is a tap gesture, and

Element 11[A]: The computer implemented system of claim 10, further comprising: said detection module further configured to detect a fourth gesture from among said multiple gestures on said one of said defined interface regions, by said gesture based media recording application, wherein said fourth gesture is a tap gesture, and

122. In my opinion, Lee'041 teaches Elements 4[A]/11[A]. Lee'041 teaches that the gesture based media recording application detects a fourth gesture from among multiple gestures on one of the defined interface regions, wherein the fourth

gesture is a tap gesture. Lee'041 teaches that its PTT application detects a fourth user action performed manually when the user taps interface region 90 of the touch-sensitive display screen. Ex-1014, 5:34-36, Fig. 6C (annotated). The “Tap-to-End feature is implemented by tapping the display screen.” Ex-1014, Cl. 2; *see also id.*, Cl. 7 (“[T]he computer code is responsive to the implementation of the Tap-to-End feature by touching the display screen.”).

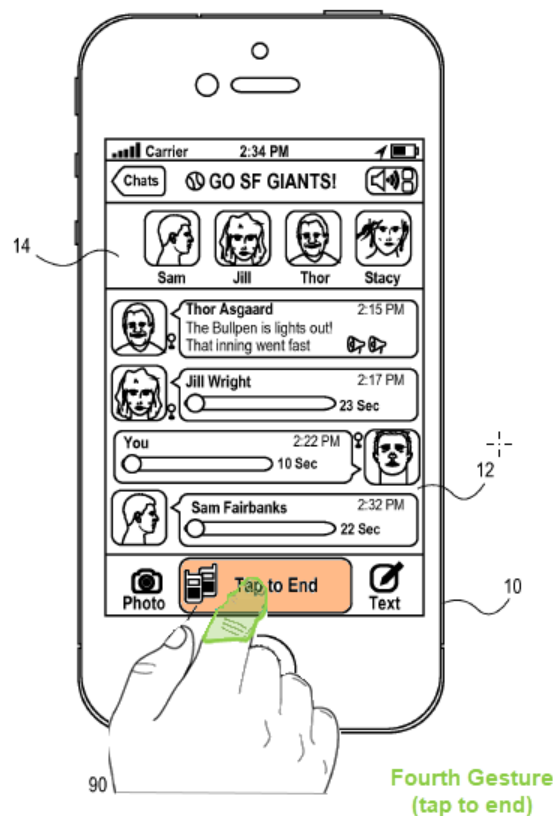


FIG. 6C

123. A POSITA would also have understood that Lee'041's PTT application comprises a detection module or “a software component” that is “configured to

detect one or more physical touch or other gesture on a defined region of a graphical user interface” because its Slide-to-Lock detection unit and controller “operate in cooperation with the touch-sensitive display screen” to detect a user’s finger tap on interface region 90 of the display screen and “may be implemented in ... software.” Ex-1014, 5:17-21.

124. Moreover, as I explain in my technical background section, the difference between a tap and hold was known by 2012 and was not inventive (*see* Section VII.B.2, ¶¶49, 58-59). For example, Dolenc teaches “forming a hold user interface event in response to a touch input or key press input being held in place for a predetermined time, and using said hold event in forming said gesture information.” Ex-1067 at 2:30-34. Moreover, Dolenc teaches that “[i]f the point of touch or click is moved after the Touch Down user interface event (without lifting the touch), a Move event or state 480 occurs. ... The Move event 480 may be terminated by lifting the touch, and a Release event 460 occurs. The Move event may be terminated also by stopping the move without lifting the touch, in which case a Hold event 470 may occur, if the touch spans a long enough time without moving. A Hold event or state 470 may be generated when a Touch Down or Move event or state continues for a long enough time. The generation of the Hold event may be done e.g. so that a timer is started at some point in the Touch Down or Move state, and when the timer advances to a large enough value, a Hold event is generated, in

case the state is still Touch Down or Move, and the point of touch has not moved significantly.” *Id.*, 16:9-26. Dolenc further teaches that “[a] Hold event or state 470 may be terminated by lifting the touch, causing a Release event 460 to be triggered, or by moving the point of activation, causing a Move event 480 to be triggered. *The existence of the Hold state or event may bring benefits in addition to just having a Touch Down event in the system, for example by allowing an easier and more reliable detection of gestures.*” *Id.*, 16:26-31. Thus, it would have been obvious to a POSITA to include a brief hold at the end of the slide/swipe to ensure the slide/swipe was intended and would have formed a natural ending to the gesture.

d. Element 4[B]: wherein said recording of said media data is ended on performing said fourth gesture.

Element 11[B]: wherein said recording of said media data is ended on performing said fourth gesture.

125. In my opinion, Lee’041 teaches Elements 4[B]/11[B]. Lee’041 teaches that the recording of the media data is ended on performing the fourth gesture. Lee’041 teaches that “[w]hen the Tap-to-End region 90 is tapped, the message is terminated and the device 10 stops transmitting media.” Ex-1014, 5:34-36, Fig. 6C (orange).

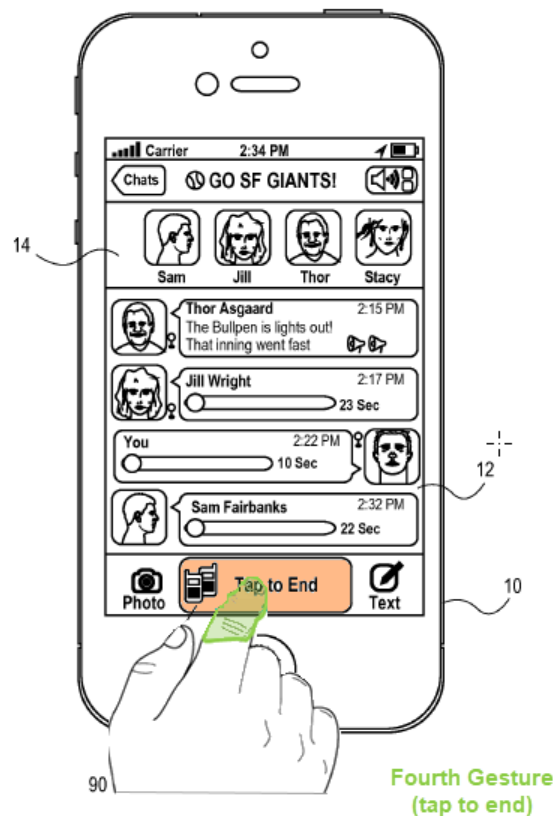


FIG. 6C

- e. **Claim 5:** The computer implemented method of claim 4, wherein said recorded media data is one or more of saved locally, deleted, or transmitted to another electronic device or the server.

Claim 12: The computer implemented system of claim 11, wherein said recorded media data is one or more of saved locally, deleted, or transmitted to another electronic device or the server.

126. In my opinion, Lee'041 teaches Claims 5 and 12. Lee'041 teaches that when the PTT locking function is activated, media created by the media creating devices is “progressively stored and/or transmitted, without the need to hold/select

the Hold-and-Talk region 16 or similar PTT feature.” Ex-1014, 5:7-10. When deactivated, “created media is progressively stored only when the Hold-and-Talk region 16 or similar PTT feature is selected.” *Id.*, 5:10-12. Moreover, Lee’041 teaches that the recorded media may be “progressively stored in memory 24” of the communication device (i.e., locally) “and progressively transmitted by transmitter 80 when the media is created, or initially stored in memory 24 and transmitted out of storage after the message is complete.” Ex-1014, 5:12-17, Fig. 5.

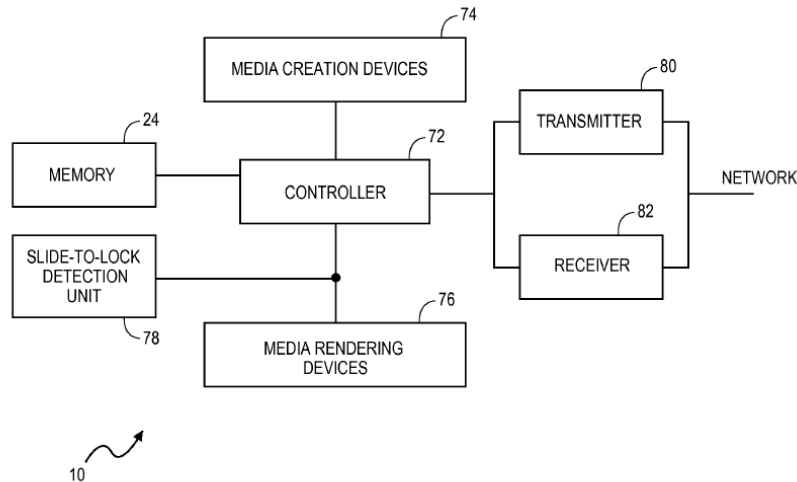


FIG. 5

- f. **Claim 6: The computer implemented method of claim 1, further comprising sending said media data to the server via a network at one of: (a) continuously while recording said media data, (b) at predefined time intervals after initiation of said recording of said media data, and (c) after completion of said recording of said media data by said gesture based media recording application.**

Claim 13: The computer implemented system of claim 8, further comprising sending said media data to the server via a network at one of: (a) continuously while recording said media data, (b) at predefined time intervals after initiation of said recording of said media data, and (c) after completion of said recording of said media data by said gesture based media recording application.

127. In my opinion, Lee'041 teaches Claims 6 and 13. Lee'041 teaches that its gesture based media recording application sends the media data to the server via a network (a) continuously while recording the media data and/or (c) after completion of the recording of the media data. Lee'041 teaches that the Slide-to-Lock feature “may be used with both (i) progressive type PTT applications where a message is transmitted as the media is created and progressively stored and (ii) store and forward type PTT applications where the message is transmitted out of storage after the message is complete.” Ex-1014, 4:6-11; *see also id.*, 5:7-12 (when the PTT lock function is activated, media created by the media creating devices is “progressively stored and/or transmitted, without the need to hold/select the Hold-and-Talk region 16 or similar PTT feature”), Fig. 5. Moreover, Lee'041 teaches that

the communication device is coupled to a network and server (*id.*, 4:58-67) and a “jitter buffer may be optimized or tuned ... for rendering incoming media as it is received over the network in real-time.” *Id.*, 4:40-42. Lee’041 further teaches that “steps can be taken on the network to enhance near real-time or live communication, such as notifying servers ... on the network and other participants ... of the conversation status so that they too can optimize operation and transmission protocols for real-time or live communication.” *Id.*, 4:46-51. A POSITA would thus have understood that the media data is sent to a *server* via the network.

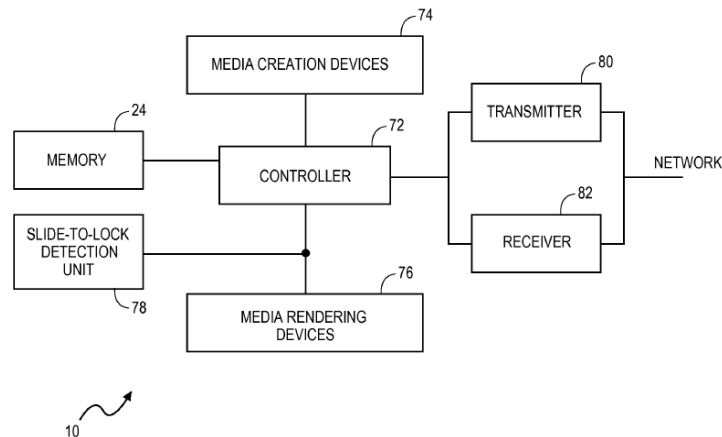


FIG. 5

- g. **Claim 7: The computer implemented method of claim 1, wherein said media data comprises one of audio data, video data, audiovisual data, image data, multimedia data, and any combination thereof.**

Claim 14: The computer implemented system of claim 8], wherein said media data comprises one of audio data, video data, audiovisual data, image data, multimedia data, and any combination thereof.

128. In my opinion, Lee'041 teaches Claims 7 and 14. Lee'041 teaches that the media data comprises at least audio data, video data, image data, and any combination thereof. Lee'041 teaches that “media may include but is not limited to, voice, video, text, photos, sensor data, GPS or positional data, or any other type of streaming or non-streaming media.” Ex-1014, 5:57-60.

B. Ground 2: Claims 1, 3-8, and 10-14 are obvious over Lee'041 in view of Ohashi

1. A POSITA would have been motivated to combine Lee'041 and Ohashi and had a reasonable expectation of success

129. In addition to the reasons set forth below, a POSITA would have been motivated to combine the teachings of Lee-041 and Ohashi (“Lee'041-Ohashi”) and would have had a reasonable expectation of success in doing so. Each relates to methods for detecting gestures on a user interface of an electronic device and implementing predefined actions that are associated with the gestures. Ex-1014, Abstract; Ex-1017, Abstract.

130. Lee'041 teaches a Push-to-Talk application running on a user's communication device, such as a mobile phone, that detects multiple user gestures on a display screen of the device and implements predefined recording actions associated with the gestures such as, for example, changing the recording mode of the device from a Hold-and-Talk mode to a Slide-to-Lock mode. Ex-1014, 3:25-51. For example, when Lee'041's application detects a “sliding motion” from interface

region 16 (Hold-and-Talk) to region 18 (Slide-to-Lock) on the user interface, the recording mode is changed so that “the user can simply speak into the microphone, or generate other types of media,” without having to contact or hold region 16 and/or 18. *Id.*, 3:37-51.

131. Similarly, Ohashi teaches a remote control apparatus for an electronic device, such as a mobile phone (Ex-1017 ¶39), that detects a “swipe and hold” gesture on a display panel of the device (*id.* ¶3) and implements a predefined action associated with the gesture such as, for example, fast-forwarding/rewinding the content and switching the displayed content. *Id.* ¶104. Ohashi teaches that “[a] continuous control such as a control from the contact starting point SP to a first swipe point MP1, to a second swipe point MP2 ... is done in the swipe and hold control while maintaining the contact.” *Id.* ¶34.

132. A POSITA would have been motivated to implement Ohashi’s swipe and **hold** gesture as a detected gesture with Lee’041’s system—to the extent it is not already included with Lee’041’s Swipe-to-Lock feature—to improve the reliability of the user’s selection because doing so would ensure that the user’s gesture is completed without inadvertent detections and that the appropriate associated predefined action is implemented. For example, Ohashi explains that “cancellation of the [user’s] contact” with the display panel “at an improper time [] fails to input a desired command,” which “sometimes decreases the operability” of the device.

Ex-1017 ¶6. Additionally, Ohashi's swipe and hold gesture "enables the user's control without checking the display on the touch panel." *Id.* ¶3. Likewise, Lee'041 states that "a user may inadvertently de-select the PTT feature in the middle of a message, causing an interruption or termination in the progressive storing and/or transmission of the media" resulting in the recipient receiving "an interrupted, partial, message." Ex-1014, 1:50-56. Thus, Lee'041's "requirement of sliding a finger across both regions 16 and 18 helps reduce the incidence of false positives when either activating or de-activating the PTT locking function using the virtual Slide-to-Lock feature." Ex-1014, 3:61-64. A POSITA would thus have recognized that implementing Ohashi's "swipe and *hold*" feature with Lee'041's "Slide-to-Lock" gesture would ensure that Lee'041's predefined action associated with this gesture (i.e., continuous recording) would be executed and avoid inadvertent termination of the recording.

133. As I explain in my technical background section (*see* Section VII.B.2, ¶¶44-49), the motivation to implement a "hold" with a sliding gesture was well known in the art by 2012. For example, Dolenc teaches "forming a hold user interface event in response to a touch input or key press input being held in place for a predetermined time, and using said hold event in forming said gesture information." Ex-1067 at 2:30-34. Moreover, Dolenc teaches that "[i]f the point of touch or click is moved after the Touch Down user interface event (without lifting

the touch), a Move event or state 480 occurs. ... The Move event 480 may be terminated by lifting the touch, and a Release event 460 occurs. The Move event may be terminated also by stopping the move without lifting the touch, in which case a Hold event 470 may occur, if the touch spans a long enough time without moving. A Hold event or state 470 may be generated when a Touch Down or Move event or state continues for a long enough time. The generation of the Hold event may be done e.g. so that a timer is started at some point in the Touch Down or Move state, and when the timer advances to a large enough value, a Hold event is generated, in case the state is still Touch Down or Move, and the point of touch has not moved significantly.” *Id.*, 16:9-26. Dolenc further teaches that “[a] Hold event or state 470 may be terminated by lifting the touch, causing a Release event 460 to be triggered, or by moving the point of activation, causing a Move event 480 to be triggered. ***The existence of the Hold state or event may bring benefits in addition to just having a Touch Down event in the system, for example by allowing an easier and more reliable detection of gestures.***” *Id.*, 16:26-31. Thus, it would have been obvious to a POSITA to include a brief hold at the end of the slide/swipe to ensure the slide/swipe was intended and would have formed a natural ending to the gesture.

134. A POSITA would have had a reasonable expectation of success in implementing the combined teachings of Lee’041-Ohashi. Lee’041 and Ohashi both disclose methods for detecting gestures on user interfaces and implementing

associated actions using familiar elements such as mobile devices, user interfaces, and programming that a POSITA would have understood are amenable to various well-understood and predictable combinations. For example, Lee'041 teaches computer code that manages the user interface to detect and implement actions associated with gestures (Ex-1014, Cl. 1) and Ohashi teaches a control component that does the same (Ex-1017 ¶¶46-51). Ohashi has a “control part 103,” which includes a CPU that uses ROM/RAM to execute the stored program and control each part of the system. Ex-1017 ¶¶46-47. Control part 103 also “decodes the coordinate signal detected from the touch panel 101a to generate coordinate data and controls each part of the commander 100 based on a contact/cancel signal and the coordinate data.” *Id.* ¶¶48 (“The control part 103 reads data stored in the memory 105 to generate display data for a graphical user interface (GUI) screen and supplies the display data to the display panel 101b.”), 49 (“The control part 103 determines the swipe and hold control that has designated a given contact starting point SP, based on the contact/cancel signal and the coordinate data.”), 51 (“After determining that a swipe control has been done, the control part 103 determines that a hold control has been done when recognizing almost-constant coordinate data while a contact signal is continuously input and then sets the coordinate data as the coordinate of a swipe point MP. The control part 103 determines that a swipe and hold control has been terminated when a cancel signal is input after a contact signal is continuously

input.”). As such, it would be well within the skill of a POSITA to implement the swipe and hold gesture on any of the computer systems of Lee’041 or Ohashi.

2. Independent Claims 1 and 8

- a. Element 1[F(i)]: detecting a second gesture from among said multiple gestures, by said gesture based media recording application, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and**

Element 8[G(i)]: said detection module further configured to detect a second gesture from among said multiple gestures, wherein said second gesture is a slide and hold gesture, or a swipe and hold gesture from said first of said defined interface regions to a second of said defined interface regions proximate to said first of said defined interface regions, and

135. In my opinion, Lee’041 in view of Ohashi teaches Elements 1[F(i)]/8[G(i)]. Lee’041 alone teaches all of Claims 1 and 8, including Elements 1[F(i)]/8[G(i)] for the reasons I describe in Section XI.A.1.h, ¶¶114-117. A “Slide-to-Lock” feature is “activated by the user sliding a finger from left to right over the region 16” on the touch-sensitive display screen of the user’s mobile device, and then holding their finger on the display screen until the ““Slide-to-Lock’ feature 18 is activated” (a second gesture). Ex-1014, 3:37-40. When this feature is activated, a recording mode called a “PTT locking function” is implemented, which allows the user to “simply speak into the microphone, or generate other types of media,” such

as voice or video messages, without having to contact or hold region 16 and/or 18 while creating a message. *Id.*, 3:45-51, 5:57-60.

136. The '030 Patent does not define “hold” in the context of the “slide and hold” or “swipe and hold” gestures. However, Lee'041's sliding motion gesture is implemented in the same manner as the only gesture in the '030 Patent to which the claimed “slide and hold” gesture could possibly refer, where the user slides their finger from interface region 301b to interface region 301c and when the user releases their finger from interface region 301c, the application continues recording. Ex-1001, 12:39-63. A POSITA would understand that in both Lee'041 and the '030 Patent, a user's sliding motion from one interface region to another while maintaining contact with the interface at the end of the motion changes the recording mode to allow recording to continue to record without the user having to continue contact. Thus, a POSITA would have understood the terms “slide and hold” and “swipe and hold” in the context of the '030 Patent claims to encompass a user holding their finger on the interface briefly after performing the swipe or slide motion. Lee'041 alone thus teaches this limitation.

137. The “slide and hold” feature is also taught by Ohashi. Ohashi teaches detecting a “swipe and hold” gesture on a display panel of an electronic device “in which a finger or the like touches an arbitrary point in the contact detecting region of the touch panel, and the finger or the like is swiped while touching the touch panel

and then is held on the touch panel while touching the touch panel.” Ex-1017 ¶3; *see also id.* ¶34 (“A continuous control such as a control from the contact starting point SP to a first swipe point MP1, to a second swipe point MP2 ... is done in the swipe and hold control while maintaining the contact.”). Moreover, Ohashi teaches transmitting “a control command according to the detected result” of the swipe and hold gesture “to the electronic device.” *Id.* ¶2. For example, in response to the swipe and hold control, “the played content can be ... fast-forwarded/rewound and the displayed content can be ... switched.” *Id.* ¶104.

138. A POSITA would have been motivated to implement Ohashi’s swipe and hold gesture as a detected second gesture with Lee’041’s system—to the extent it is not already included with Lee’041’s Swipe-to-Lock feature—for the reasons I discuss in Section XI.B.1, ¶¶129-134, namely, to improve the reliability of the user’s selection because doing so would ensure that the user’s gesture is completed without inadvertent detections and that the appropriate predefined action for that gesture is implemented. For example, Ohashi explains that “cancellation of the [user’s] contact” with the display panel “at an improper time [] fails to input a desired command,” which “sometimes decreases the operability” of the device. Ex-1017 ¶6. Additionally, Ohashi’s swipe and hold gesture “enables the user’s control without checking the display on the touch panel.” *Id.* ¶3. Likewise, Lee’041 states that “a user may inadvertently de-select the PTT feature in the middle of a message, causing

an interruption or termination in the progressive storing and/or transmission of the media” resulting in the recipient receiving “an interrupted, partial, message.” Ex-1014, 1:50-56. Thus, Lee’041’s “requirement of sliding a finger across both regions 16 and 18 helps reduce the incidence of false positives when either activating or deactivating the PTT locking function using the virtual Slide-to-Lock feature.” Ex-1014, 3:61-64. A POSITA would thus have recognized that implementing Ohashi’s “swipe and *hold*” feature with Lee’041’s “Slide-to-Lock” gesture would ensure that Lee’041’s predefined action associated with this gesture (i.e., continuous recording) would be executed and avoid inadvertent termination of the recording.

139. As I explain in my technical background section (*see* Section VII.B.2, ¶¶44-49), the motivation to implement a “hold” with a sliding gesture was well known in the art by 2012. For example, Dolenc teaches “forming a hold user interface event in response to a touch input or key press input being held in place for a predetermined time, and using said hold event in forming said gesture information.” Ex-1067 at 2:30-34. Moreover, Dolenc teaches that “[i]f the point of touch or click is moved after the Touch Down user interface event (without lifting the touch), a Move event or state 480 occurs. ... The Move event 480 may be terminated by lifting the touch, and a Release event 460 occurs. The Move event may be terminated also by stopping the move without lifting the touch, in which case a Hold event 470 may occur, if the touch spans a long enough time without moving.

A Hold event or state 470 may be generated when a Touch Down or Move event or state continues for a long enough time. The generation of the Hold event may be done e.g. so that a timer is started at some point in the Touch Down or Move state, and when the timer advances to a large enough value, a Hold event is generated, in case the state is still Touch Down or Move, and the point of touch has not moved significantly.” *Id.*, 16:9-26. Dolenc further teaches that “[a] Hold event or state 470 may be terminated by lifting the touch, causing a Release event 460 to be triggered, or by moving the point of activation, causing a Move event 480 to be triggered. ***The existence of the Hold state or event may bring benefits in addition to just having a Touch Down event in the system, for example by allowing an easier and more reliable detection of gestures.***” *Id.*, 16:26-31. Thus, it would have been obvious to a POSITA to include a brief hold at the end of the slide/swipe to ensure the slide/swipe was intended and would have formed a natural ending to the gesture.

140. A POSITA would have had a reasonable expectation of success in doing so for the reasons I discuss in Section XI.B.1, ¶¶129-134, namely, because both disclose methods for detecting gestures on user interfaces and implementing associated actions using familiar elements such as mobile devices, user interfaces, and programming that a POSITA would have understood are amenable to various well-understood and predictable combinations. For example, Lee’041 teaches computer code that manages the user interface to detect and implement actions

associated with gestures (Ex-1014, Cl. 1) and Ohashi teaches a control component that does the same (Ex-1017 ¶¶46-51). Ohashi has a “control part 103,” which includes a CPU that uses ROM/RAM to execute the stored program and control each part of the system. Ex-1017 ¶¶46-47. Control part 103 also “decodes the coordinate signal detected from the touch panel 101a to generate coordinate data and controls each part of the commander 100 based on a contact/cancel signal and the coordinate data.” *Id.* ¶¶48 (“The control part 103 reads data stored in the memory 105 to generate display data for a graphical user interface (GUI) screen and supplies the display data to the display panel 101b.”), 49 (“The control part 103 determines the swipe and hold control that has designated a given contact starting point SP, based on the contact/cancel signal and the coordinate data.”), 51 (“After determining that a swipe control has been done, the control part 103 determines that a hold control has been done when recognizing almost-constant coordinate data while a contact signal is continuously input and then sets the coordinate data as the coordinate of a swipe point MP. The control part 103 determines that a swipe and hold control has been terminated when a cancel signal is input after a contact signal is continuously input.”). As such, it would be well within the skill of a POSITA to implement the swipe and hold gesture on any of the computer systems of Lee’041 or Ohashi.

3. Dependent Claims 3-7 and 10-14

141. In my opinion, Lee'041 in view of Ohashi teaches dependent Claims 3-7 and 10-14 for the same reasons as Lee'041 alone. *See* Section XI.A.2, ¶¶119-128.

C. Ground 3: Claims 2 and 9 are obvious over Lee'041 in view of Ohashi in further view of Lee'804 and D'Souza

142. In my opinion, Lee'041 in view of Ohashi, Lee'804, and D'Souza ("Lee'041-Ohashi-Lee'804-D'Souza") teaches Claims 2 and 9. In fact, as set forth below, Lee'041-Ohashi teach all of the limitations except for very specific and obvious design choices that would have been obvious to a POSITA.

1. Dependent Claims 2 and 9

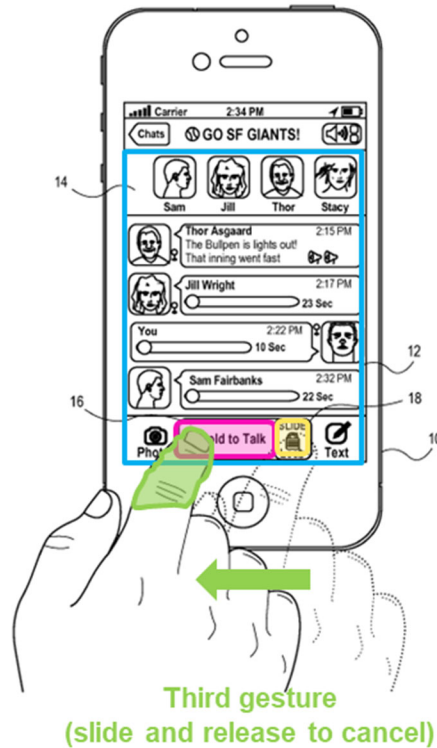
- a. **Element 2[A]: The computer implemented method of claim 1, further comprising: detecting a third gesture from among said multiple gestures, by said gesture based media recording application, wherein said third gesture is a slide and release gesture, or a swipe and release gesture from said second of said defined interface regions to a third of said defined interface regions proximate to said first of said defined interface regions, and**

Element 9[A]: The computer implemented system of claim 8, further comprising: said detection module further configured to detect a third gesture from among said multiple gestures, wherein said third gesture is a slide and release gesture, or a swipe and release gesture from said second of said defined interface regions to a third of said defined interface regions proximate to said first of said defined interface regions, and

143. In my opinion, Lee'041 in view of Ohashi, Lee'804, and D'Souza teaches Elements 2[A]/9[A]. Lee'041 teaches that the gesture based media recording application detects a third gesture from among multiple gestures, wherein the third gesture is a slide and release gesture from the second defined interface region to a third defined interface region proximate to the first defined interface region. Lee'041 teaches that its PTT application detects a third user action performed manually by the user sliding their finger from interface region 18 (the Slide-to-Lock position) to interface region 16 of the touch-sensitive display screen and releasing it at interface region 16. For example, Lee'041 teaches that "[t]he PTT locking function may be de-activated," using the Slide-to-Lock feature "by sliding a finger from region 18 to region 16," as shown in Figure 2D.¹⁶ Ex-1014, 3:52-55. A POSITA would have understood that because Lee'041's application dynamically changes the functions associated with the defined interface regions based on the user's gestures (e.g., region 16 changes from PTT mode to TTE mode, as I discuss in Section XI.A.1.g, ¶¶111-113), region 16 becomes the third defined interface region whereby a right to left swipe deactivates the recording. A POSITA would have further understood that Lee'041-Lee'804's slide and release gesture from interface region 18 to region 16 deactivates the PTT lock function and returns the device to a PTT recording mode

¹⁶ Note that this is an alternative to the Tap-to-End embodiment described in Figure 6C, discussed with respect to Claims 3-4, 10-11. Ex-1014, 5:22-36.

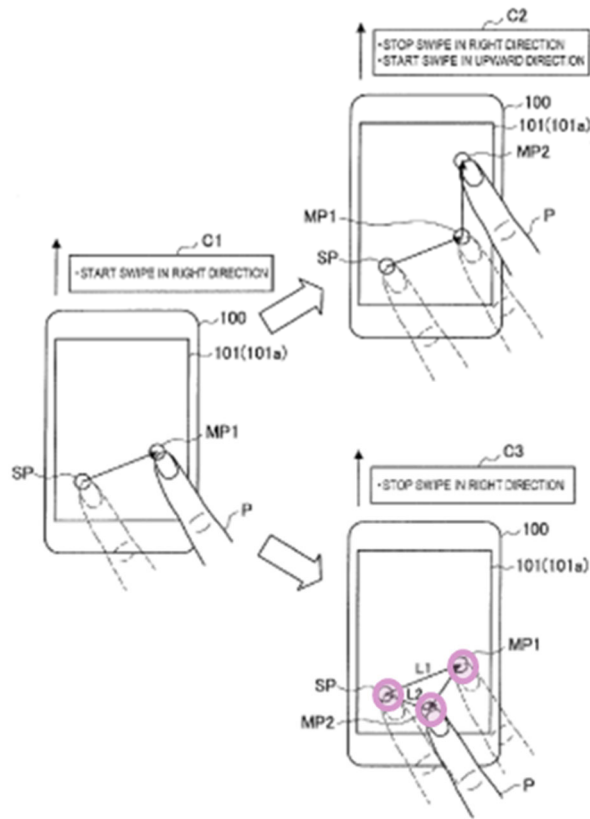
in which the recording is stopped and the message terminated (or canceled) when the user releases their finger from the screen.



144. A POSITA would also have understood that Lee'041's PTT application comprises a detection module or "a software component" that is "configured to detect one or more physical touch or other gesture on a defined region of a graphical user interface" because its Slide-to-Lock detection unit and controller "operate in cooperation with the touch-sensitive display screen" to detect a user's slide and release action from interface region 18 to 16 of the display screen and "may be implemented in ... software." Ex-1014, 5:17-21.

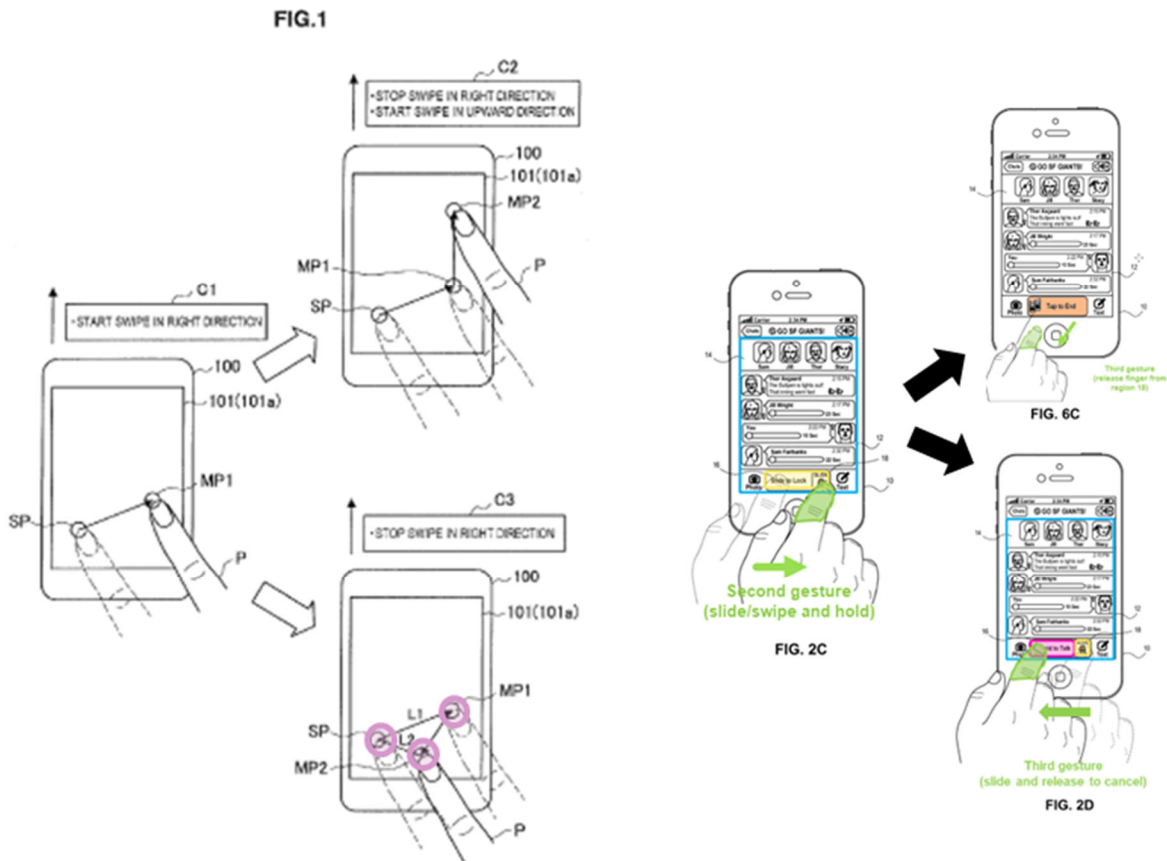
145. To the extent it is required that the swipe and release gesture end on an interface region that is geographically distinct on the display from where the recording started on region 16, this would have been obvious as a matter of routine design choice to a POSITA, as exemplified by Ohashi. Ohashi teaches detecting a gesture from the second defined interface region to a *third defined interface region* proximate to the first defined interface region on a display panel of an electronic device. Ohashi teaches that “[a] continuous control such as a control from the contact starting point SP to a first swipe point MP1, to a second swipe point **MP2** ... is done in the swipe and hold control while maintaining the contact.” Ex-1017 ¶34, Fig. 1 (annotated).

FIG.1



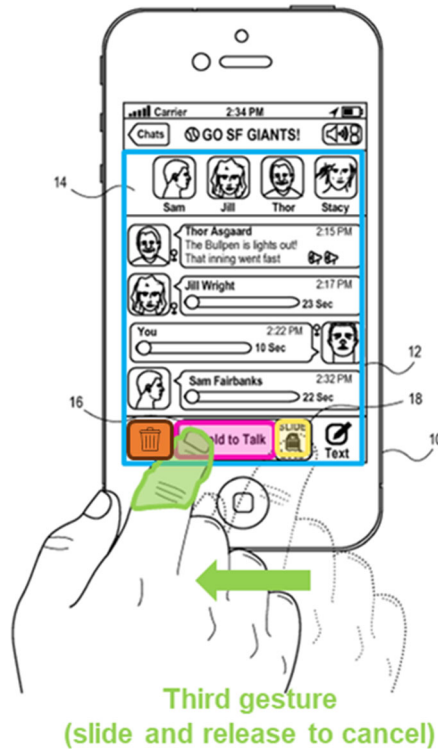
146. A POSITA would have been motivated to implement Ohashi's third interface region with Lee'041's system with a reasonable expectation of success to provide the user more gesture options. For example, Lee'041 teaches two "non-exclusive" embodiments, such as simultaneously enabling the third slide-and-release gesture of this claim (Ex-1014, Figure 2D) and Tap-to-End gesture (*id.*, Fig. 6C). *Id.*, 5:22-25. A POSITA would have implemented a geographically distinct third interface region to ensure accurate differentiation between gestures based on the specific region touched on the interface. For example, Lee'041 notes that the Tap-to-End feature "may be implemented anywhere on the display screen 12" including

in the “vicinity of where the virtual Slide-to-[L]ock 18 feature is implemented.” Ex-1014, 5:49-54. A POSITA would have recognized that third interface region could similarly have been implemented “in the vicinity” of the Slide-to-Lock region 18. A POSITA would have also been motivated to combine Lee’041 with Ohashi for the reasons I describe in Section XI.B.1, ¶¶129-134.



For example, in the Lee’041-Ohashi combination, Lee’041’s interface would include a geographically distinct third interface region with a “trash bin” feature such that the user could slide and release to that region to implement the associated third

gesture function (cancel recording and/or delete it). Fig. 2D (annotated, orange). The motivation to include this feature would be to decrease the possibility of accidental detections on the interface.



147. A POSITA would also have had a reasonable expectation of success in implementing the combined teachings of Lee'041-Ohashi. Lee'041 and Ohashi both disclose methods for detecting gestures on user interfaces and implementing associated actions using familiar elements such as mobile devices, user interfaces, and programming that a POSITA would have understood are amenable to various well-understood and predictable combinations. Ex-1014, Cl. 1; Ex-1017 ¶¶46-51. As such, it would be well within the skill of a POSITA to implement the swipe and

release to cancel feature on any of the computer systems of Lee'041 or Ohashi (*see* Section XI.B.1, ¶¶129-134).

b. Element 2[B]: wherein said recording of said media data is canceled on performing said third gesture, and

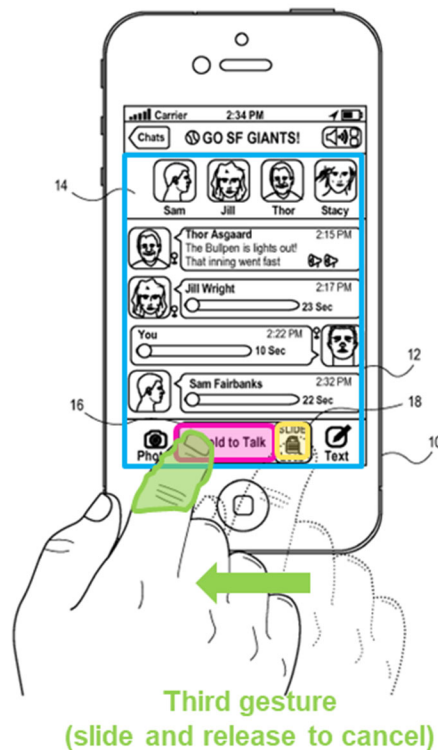
Element 9[B]: wherein said recording of said media data is canceled on performing said third gesture, and

Element 2[C]: wherein there is no transmission of already recorded media data to another electronic device or a server.

Element 9[C]: wherein there is no transmission of already recorded media data to another electronic device or a server.

148. In my opinion, Lee'041 in view of Ohashi, Lee'804, and D'Souza teaches Elements 2[B]-2[C]/9[B]-9[C]. Lee'041-Lee'804 teaches that the recording of the media data is canceled on performing the third gesture. Lee'041 explains that “[t]he Slide-to-lock feature 18 can be used for de-activation by sliding a finger from region 18 to region 16, as illustrated in Fig. 2D.” Ex-1014, 3:52-55. Lee'804, which is related to and incorporated by reference in Lee'041 (Ex-1014, 1:9-13), includes identical disclosures as Lee'041 for Figures 2A-2D. Ex-1015, 3:19-49. Moreover, Lee'804 teaches “a message module configured to generate a message containing media when the PTT locking function is activated using the Slide-to-Lock feature” and “*terminate[] the message* when the PTT locking function is de-activated” by the slide and release gesture in Figure 2D. Ex-1015, Cls. 15-17. A POSITA would have

understood that Lee'041-Lee'804's slide and release gesture from interface region 18 to region 16 deactivates the PTT lock function and returns the device to a PTT recording mode in which the recording is stopped and the message terminated (or canceled) when the user releases their finger from the screen. A POSITA would have recognized that the gesture would include the user releasing their finger because to continue using the interface for additional operations controlled by further gestures, the finger would have to be off of the screen in order to perform those additional gestures. Furthermore, it would be an obvious design choice to require the release of the finger as part of the slide gesture to de-activate the PTT lock function to further "reduce the incidence of false positives when either activating or de-activating the PTT locking function using the virtual Slide-to-Lock feature." Ex-1014, 3:61-64. Additionally, there are a finite number of such design choices (two) since the option here is merely to include or not include the release of the finger as part of the gesture.



149. Lee'041-Lee'804 further teaches that there is no transmission of already recorded media data to another electronic device or a server. Lee'041-Lee'804 teaches a non-streaming embodiment wherein the created media is "initially stored in memory 24 and transmitted out of storage after the message is complete." Ex-1014, 5:12-17, 5:55-60 (media includes "non-streaming media"); Ex-1015, 5:6-11, 5:19-22. Lee'804 also teaches that the message terminated by the messaging module when the PTT locking function is deactivated may contain "non-streaming media." Ex-1015, Cls. 16-17. A POSITA would have understood that a terminated message containing non-streaming media would not be "complete[d]" and, thus,

would not be transmitted to another electronic device or a server. I explain this well-known concept in my technical background section (*see* Section VII.D, ¶¶64-67).

150. To the extent more implementation details were desired by a POSITA, D’Souza also teaches a swipe and release gesture to cancel an impending capture of media data such as a photograph. Ex-1018 ¶29. D’Souza teaches that “the taking of a photograph can be canceled,” for example, “when the [user’s] finger is moved to another portion of the touchscreen and then lifted from the touch screen.” Ex-1018 ¶13. For instance, if the user decides that they do not want to take a picture, the user slides their “finger away from the shutter button 300 such that contact is no longer made with any portion of the shutter button 300 but such that contact is maintained with the touch screen 200 throughout the sliding motion” and “[w]hen the user then lifts his or her finger from the touch screen 200, a photograph would not be taken and the autofocus and/or any other photography-related functions that had been activated by the pressing of the finger to the shutter button 300 would be deactivated.” *Id.* ¶17, Fig. 4 (annotated).

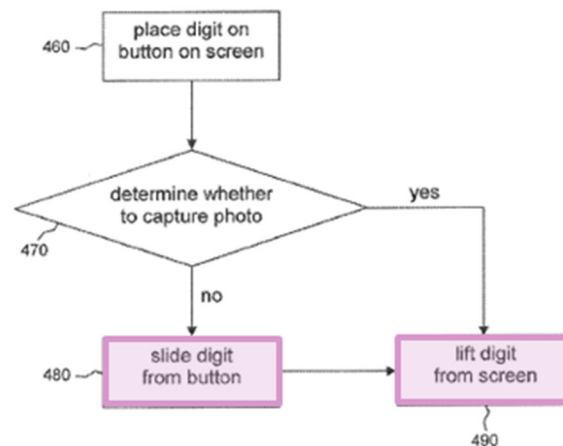


Figure 4

151. A POSITA would have been motivated to combine Lee’804 and D’Souza with the Lee’041-Ohashi combination with a reasonable expectation of success. Lee’804 is related to and incorporated by reference in Lee’041 and contains identical disclosures for Figures 2A-2D to Lee’041; thus, a POSITA’s motivation to combine Lee’804 with Ohashi and expectation of success would be the same as I describe for Lee’041 in Sections XI.B.1, ¶¶129-134 and Section XI.C.1.a, ¶¶143-147. *See* Ex-1014, 1:9-13; Ex-1015, 3:19-49. In particular, a POSITA would have found it beneficial to include Lee’804’s more specific disclosures regarding termination of a message following a swipe and release gesture in the Lee’041-Ohashi combination to ensure that a recording is not inadvertently terminated. Ohashi explains that “cancellation of the [user’s] contact” with the display panel “at an improper time [] fails to input a desired command,” which “sometimes decreases

the operability” of the device. Ex-1017 ¶6. Likewise, Lee’041 and Lee’804 both state that “a user may inadvertently de-select the PTT feature in the middle of a message, causing an interruption or termination in the progressive storing and/or transmission of the media.” Ex-1014, 1:50-56; Ex-1015, 1:42-46. Both explain that the slide gesture “reduce[s] the incidence of false positives when either activating or deactivating the PTT locking function.” Ex-1014, 3:61-64; Ex-1015, 3:55-58. A POSITA would thus have recognized that including a predefined termination function associated with a specific swipe and release gesture (as taught by Lee’804) in the combined Lee’041-Ohashi system would improve the operability of the device and ensure that messages are only intentionally terminated.

152. A POSITA would have been motivated to implement D’Souza’s swipe and release gesture to cancel the capture of media data with the combined Lee’041-Lee’804-Ohashi system—to the extent it is not already included by Lee’804’s termination feature—for similar reasons, because it would ensure that a recording is not inadvertently terminated. On the other hand, it would also ensure that recorded information that the user did *not want* transmitted to another electronic device or server would not be inadvertently transmitted. For example, Lee’041-Lee’804 states that “the Talk feature is often inadvertently selected, causing the creation of ‘false’ messages” and “[i]f the user is unaware of the situation, a potentially long, data intensive, message that consumes bandwidth may be transmitted.” Ex-1014, 2:6-10;

Ex-1015, 1:63-67. Implementing D’Souza’s intentional swipe and release gesture to cancel the capture of media data in the combined system would avoid this inconvenience for the user.

153. A POSITA would have had a reasonable expectation of success in adding Lee’804 and D’Souza to the Lee’041-Ohashi combined system because all disclose methods for detecting gestures on user interfaces and implementing associated actions using familiar elements such as mobile devices, user interfaces, and programming that a POSITA would have understood are amenable to various well-understood and predictable combinations. For example, Lee’041-Lee-804 teaches computer code that manages the user interface to detect and implement actions associated with gestures (Ex-1014, Cl. 1; Ex-1015, Cl. 1), Ohashi teaches a control component that does the same (Ex-1017 ¶¶46-51), and D’Souza teaches a processor that does the same (Ex-1018 ¶¶22-23). D’Souza teaches a processor that “executes instructions, codes, computer programs, or scripts that it might access from network connectivity devices 1320, RAM 1330, ROM 1340, or secondary storage 1350. ... While only one CPU 1310 is shown, multiple processors may be present” such that “the instructions may be executed simultaneously, serially, or otherwise by one or multiple processors.” Ex-1018 ¶22. As such, it would be well within the skill of a POSITA to implement the swipe and release to cancel gesture,

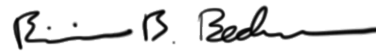
with no transmission of media data to another device or server, on any of the computer systems of Lee'041-Lee'804, Ohashi, or D'Souza.

XII. CONCLUSION

154. For the reasons above, it is my opinion that Claims 1-14 of the '030 Patent are obvious based on each of the grounds specified in my Declaration.

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

Date: December 1, 2023



Dr. Benjamin B. Bederson