

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

SMITH INTERFACE TECHNOLOGIES, LLC

Patent Owner

Case IPR2024-01083

Patent No. 10,642,413

DECLARATION OF CRAIG ROSENBERG, Ph.D.

TABLE OF CONTENTS

I. INTRODUCTION.....	1
II. PROFESSIONAL BACKGROUND.....	1
III. COMPENSATION.....	8
IV. LEGAL CONSIDERATION.....	8
A. Anticipation	8
B. Obviousness	9
C. Claim Interpretation	10
D. Overview of Task & Basis for Opinions	11
E. Level of Skill in the Relevant Art	11
V. OVERVIEW OF THE '413 PATENT.....	13
A. Independent Claims 1 and 101	18
VI. OVERVIEW OF THE PRIOR ART	21
A. U.S. Patent App. No. 2008/0207188 ("Ahn")	22
B. U.S. Patent App. No. 2007/0150842 ("Chaudhri")	32
C. U.S. Patent App. No. 2011/0209093 ("Hinckley")	33
1. Bezel Gestures.....	34
2. Radial Menu	39
D. Radial Menus	41
VII. GROUND 1: DR. TERVEEN'S ANALYSIS FAILS TO SHOW OBVIOUSNESS OF CLAIMS 1-3, 7, 9, 10, AND 101 BY AHN, CHAUDHRI, AND HINCKLEY	49
A. Claims 1 and 101 are Not Obvious in view of Ahn, Chaudhri, and Hinckley	50
1. [1.e.i-iii], [101.e.i-iii]: detecting a first duration of contact on at least one of the one or more first menu items and in response [...]	50
B. Dr. Terveen's Motivations for Modifying Ahn with Hinckley are Incorrect	63
1. The 'Techniques' Identified by Dr. Terveen are Not Related to Hinckley's Radial Menu	64
2. A POSITA Would Have Understood that Hickley's Radial Menu is Not a Simple Substitution for Ahn's Linear Menus.	68
C. Dr. Terveen's Mapping of Claim 101 is Internally Inconsistent	86
1. The Institution Decision Does Not Follow the Theory Laid Out by Dr. Terveen and Contradicts Dr. Terveen's Other Theories.....	90
D. Dr. Terveen's Mapping to the Blurring Limitation of Claim 1 is Deficient	93

1.	Ahn Theory.....	94
2.	Ahn-Chaudhri Theory	101
VIII.	CONCLUSION.....	112

I. INTRODUCTION

1. My name is Dr. Craig Rosenberg, , Ph.D. I have been retained by Smith Interface Technologies, LLC as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”).

II. PROFESSIONAL BACKGROUND

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

3. 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 herein. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as EX2007. The following provides a brief overview of some of my experiences that are relevant to the matters set forth in this Declaration.

4. My qualifications to testify about the patents in this case and the relevant technology are set forth in my curriculum vitae (“CV”), which I have included as Exhibit A. In addition, a brief summary of my qualifications is included below.

5. I hold a Bachelor of Science in Industrial Engineering, a Master of Science in Human Factors, and a Ph.D. in Human Factors from the University of Washington School of Engineering. For 30 years, I have worked in various areas, including human factors, user interface design, software development, software architecture, systems engineering, and modeling and simulation, across a wide range of application areas, such as aerospace, communications, entertainment, and healthcare.

6. I graduated from the University of Washington in 1988 with a Bachelor of Science degree in Industrial Engineering. After graduation, I continued my studies at the University of Washington College of Engineering. In 1990, I obtained an M.S. in Human Factors. Human Factors is an engineering discipline that studies the design of products, processes, equipment, and systems to enhance their efficiency in interaction with humans. The goal of human factors is to reduce human error, increase productivity and efficiency, and enhance safety and comfort, with a specific focus on the interaction and interfaces between humans and the products, processes, or systems with which they interact. In 1994, I earned a Ph.D. in Human Factors, with a focus on advanced interface design. My B.S., M.S., and Ph.D. were all taught at the University of Washington College of Engineering.

7. During my doctoral studies, I worked as an Associate Assistant Professor of Human Factors at the University of Washington's Industrial

Engineering Department. My duties included teaching, writing research proposals, designing and conducting funded human factors experiments for the National Science Foundation, and hiring and supervising students. While studying at the University of Washington, I also worked as a human factors researcher, designing and conducting advanced human factors experiments related to virtual environments, interface design, stereoscopic displays, and advanced visualization research, which the National Science Foundation funded. My duties included user interface design, system design, software development, graphics programming, experimental design, and interfacing between hardware and software.

8. I have published twenty-one research papers in professional journals and proceedings in the areas of user interface design, computer graphics, and the design of spatial, stereographic, and auditory displays. I also authored a book chapter on augmented reality displays in the book “Virtual Environments and Advanced Interface Design” (Oxford University Press, 1995). In addition, I created one of the first virtual spatial musical instruments called the MIDIBIRD that utilized the MIDI protocol, two six-dimensional spatial trackers, a music synthesizer, and a computer graphics workstation to create an advanced and novel musical instrument.

9. For the past 21 years, I have served as a consultant for Global Technica, Sunny Day Software, Stanley Associates, Techrizon, CDI Corporation, and the Barr

Group. In this capacity, I have provided advanced engineering services for many companies.

10. I consulted for the Boeing Company for over 16 years as a senior human factors engineer, user interface designer, and software architect for a wide range of advanced commercial and military programs. Many of the projects that I was involved with at Boeing involved advanced software development, user interface design, agent-based software, and modeling and simulations in the areas of missile defense, homeland security, battle command management, computer-aided design, networking and communications, air traffic control, location-based services, and Unmanned Aerial Vehicle (“UAV”) command and control. Additionally, I served as the lead system architect, developing advanced air traffic controller workstations and air traffic control analysis applications, toolsets, and trade study simulations for Boeing Air Traffic Management.

11. I was also the architect of the Boeing Human Agent Model. The Boeing Human Agent Model is an advanced model for simulating human sensory, cognitive, and motor performance as applied to the roles of air traffic controllers, pilots, and UAV operators. In another project, I served as the lead human factors engineer and user interface designer for Boeing’s primary vector and raster computer-aided drafting and editing system, which produces maintenance manuals, shop floor illustrations, and service bulletins for aircraft manufactured by the Boeing

Commercial Aircraft Company. Additional responsibilities during my time as a consultant include system engineering, requirements analysis, functional specification, use case development, user stories, application prototyping, modeling and simulation, object-oriented software architecture, graphical user interface analysis, and design, as well as software development using UML, C++, C#, and Java.

12. In 1995 and 1996, I was hired as the lead human factors engineer and user interface designer for the first two-way pager produced by AT&T. Prior to this technology, individuals could receive pages but had no way to respond using their pager. This new technology enabled users to use a small, handheld device to receive and send canned or custom text messages, access and update their address book, and view and update their personal calendar. This high-profile project involved designing the entire feature set, user interface/user interaction design and specification, as well as all graphical design standards.

13. From 1999–2001, I was the lead human factors engineer and user interface designer for a company called Eyematic Interfaces, where I was responsible for all user interface design and development activities associated with real-time mobile handheld 3D facial tracking, animation, avatar creation, and editing software for a product for Mattel. My work involved user interface design, human factors analysis, requirements gathering and analysis, and functional specifications.

14. I was the lead user interface designer for a company called ObjectSpeed, which developed a portable handheld telephone for use in homes and businesses that had many of the same capabilities that we take for granted in mobile cellular phones. This portable multifunction device supports voice, email, chat, video conferencing, internet radio, streaming media, Microsoft Outlook integration, photo taking and sharing, etc. The ObjectSpeed device was specifically designed and developed as a portable handheld device.

15. I was the lead UI designer for Ahaza, a company that builds IPv6 routers. I designed the user interfaces for configuring and controlling these advanced network hardware devices. My responsibilities included requirements analysis, functional specification, user interface design, user experience design, and human factors analysis.

16. I am the founder, inventor, user interface designer, and software architect of WhereWuz. WhereWuz is a company that develops advanced mobile software for GPS-enabled smartphones and handheld devices. WhereWuz allows users to record exactly where they have been and query this data in unique ways for subsequent retrieval based on time or location. WhereWuz was specifically designed and developed to run on small handheld devices.

17. I am the co-founder of a medical technology company called Healium. Healium developed advanced wearable and handheld user interface technology, enabling physicians to interact more effectively with electronic medical records.

18. I am the co-founder of StratoScientific, a medical technology company. StratoScientific is developing an innovative case for a smartphone that transforms a standard handheld device into a full-featured digital stethoscope, incorporating visualization and machine learning for telemedicine and automated diagnosis.

19. I designed and developed a large software project for Disney World, called xVR, which enabled operational employees to use a handheld device to view the current and historical status of all guests within multiple attractions and one of their restaurants. The application could run in a real-time/live mode, displaying data collected from sensors that show the location and status of all guests within the attraction. The application could also be run in a fast time/simulated mode. The application was developed on a laptop computer and was specifically designed to run on various devices, including laptops, PCs, smartphones, and tablets.

20. I have received several awards for my engineering work relating to interface design, computer graphics, and the design of spatial, stereographic, and auditory displays, including a \$10,000 scholarship from the I/ITSEC for advancing the field of interactive computer graphics for flight simulation and a Link Foundation award for furthering the field of flight simulation and virtual interface design. I have

also created computer graphic illustrations for several popular book covers and animations for a movie produced by MIRAMAR.

III. COMPENSATION

21. My work on this matter is being billed at my customary rate of \$650 an hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services. I have no pecuniary interest in the outcome of this proceeding. I understand I will be paid regardless of the outcome of any proceeding in which my work is used.

IV. LEGAL CONSIDERATION

22. I am not an attorney. My understanding of the law is based on information provided by counsel for Smith Interface Technologies, LLC.

A. Anticipation

23. I understand that a reference anticipates a claim if it discloses each and every element recited in the claim, arranged as in the claim, so as to enable a person of ordinary skill in the art (POSITA) to make and use the claimed invention without the need for undue experimentation in light of the general knowledge available in the art. I understand that a U.S. Patent document is presumed to have sufficient description to include sufficient detail for a person of ordinary skill in the art to make and use the subject matter that the document describes.

24. I also understand that a claim limitation may be present in an alleged prior art reference either expressly or inherently, but inherency may be established

only if such limitation is necessarily present in the reference. I understand that inherency cannot be established by probabilities or possibilities, , and the mere fact that a certain thing may result from a given set of circumstances is not sufficient to establish inherency.

B. Obviousness

25. I understand that a claimed invention is obvious and, therefore, not patentable if the subject matter claimed would have been obvious to a person of ordinary skill in the art at the time that the invention of the '413 Patent (a “POSITA”), which I have been asked to treat as August 5, 2011.

26. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a modification or combination was within the skill of a POSITA. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I also understand that to establish a finding of obviousness one must show that one of ordinary skill in the art would have had a motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

27. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in

which each element performs the same function it was known to perform, and the arrangement yields expected results, (2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was “obvious to try” because of a design need or market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify or combine prior art references to arrive at the claimed invention.

C. Claim Interpretation

28. I understand that a claim term is interpreted according to its ordinary and customary meaning as a POSITA would have understood the term in light of the surrounding claim language, other claims, the specification, and the patent’s prosecution history, which are referred to as intrinsic evidence. I also understand that prior art references cited in the patent’s prosecution history are considered intrinsic evidence. I further understand that evidence outside the patent and its prosecution history (e.g., dictionaries and technical articles) may inform the context in which a POSITA would have understood the claims of a patent. I understand this

ordinary and customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term.

D. Overview of Task & Basis for Opinions

29. I have been asked to review U.S. Patent No. 10,642,413 (referred to as “the ’413 Patent”). I have been asked to provide opinions related to certain issues from the perspective of a person of ordinary skill, having knowledge of the relevant art, as of August 5, 2011, and—except where otherwise noted—the opinions stated in this declaration are from that perspective. The qualifications and abilities of such a person are described in paragraph 32 below.

30. My opinions are based on my education, training, and experience as well as items that I reviewed to prepare my opinions, including the Petition, the documents listed in the Exhibit List included with the Petition, and any other items that I reference in my analysis below.

31. I understand that other issues may arise that require further explanation, and I will provide that explanation if appropriate. As a result, I respectfully reserve the right to update and supplement this declaration and the information and opinions provided herein.

E. Level of Skill in the Relevant Art

32. I understand that a patent must be analyzed from the perspective of a person of ordinary skill in the art (“POSITA”) as of the time of invention of the

patent, which is typically considered to be the patent's earliest filing date. While I do not necessarily agree with every aspect of Dr. Terveen's explanation of the level of ordinary skill in the art, I do not currently believe the aspects of that explanation I would change or clarify would impact the analysis set forth in this declaration. Nevertheless, for context, the following reflects the knowledge and capabilities a POSITA would have possessed at the time of the '413 Patent's priority date in August 2011.

33. It would have been well known to a POSITA that menu usability depends not only on visual layout, but also on context and input modality. For example, radial menus offered advantages for fixed-location tool selection, but they were generally considered unsuitable for hierarchical, cascading, or multi-state menu structures. *See* EX2009 at 2 (“Just replacing a rectangular context menu with a pie menu doesn’t magically make everything better, though. In fact, it introduces a new set of problems that need to be solved.”).

34. For purposes of this declaration, my opinions are therefore provided from the perspective of the POSITA described in paragraph 30 of Dr. Terveen's declaration (EX1003) as of August 5, 2011. This POSITA would have a Bachelor's degree in computer science, computer engineering, electrical engineering, human-computer interaction, or a comparable field and at least two years of professional experience in the design and development of graphical user interfaces (GUIs),

human-computer interfaces, or equivalents thereof. My education, technical expertise, and personal knowledge discussed in Section II show that I meet the qualifications of a POSITA.

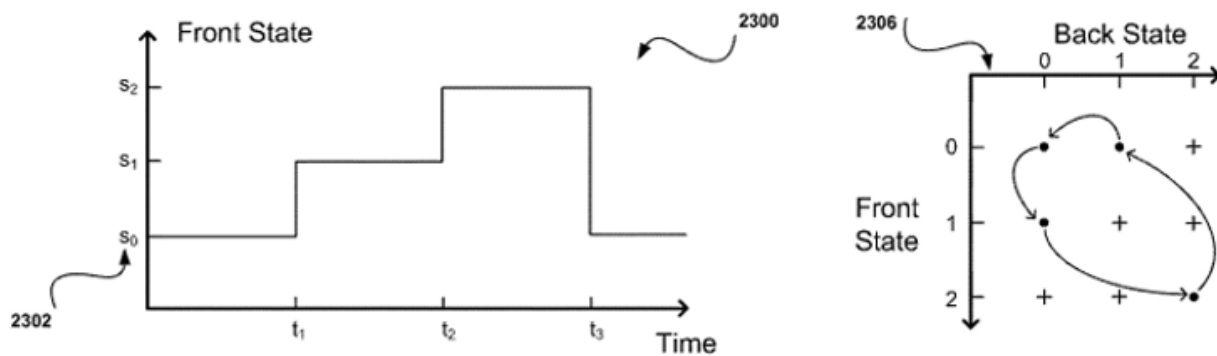
V. OVERVIEW OF THE '413 PATENT

35. The '413 Patent describes devices, methods and graphical user interfaces for manipulating user interface objects. *See generally* EX1001. At the time of the '413 Patent's August 2011 priority date, touch screen mobile devices were become increasingly capable of more complex tasks that required more control than possible with basic input method. Further, limited screen real estate meant that input and output techniques used for more complex tasks on desktop computers could not be used. The '413 Patent addresses this shortcoming by enabling users to operate these new powerful mobile devices and their feature-rich applications through implementation of multi-part gestures where users can take successive actions and receive feedback at each stage.

36. The '413 Patent describes these multi-part, multi-action gestures as “pattern-based gestures” that “may be recognized in the form of an ordered sequence” by, for example, “using a decision tree.” EX1001, 74:14-16; EX2008, ¶[00581]. In the context of the '413 Patent, a “phase space” is a conceptual space that captures the evolving states of user interaction over one or more dimensions. The patent defines a pattern-based gesture as “a path through a phase space of one

or more dimensions” (EX1001, 72:17–19), EX2008, ¶[00564], where each dimension can represent a property of the gesture—such as pressure level, contact location, contact duration, or the interaction surface (e.g., front or back of the device).

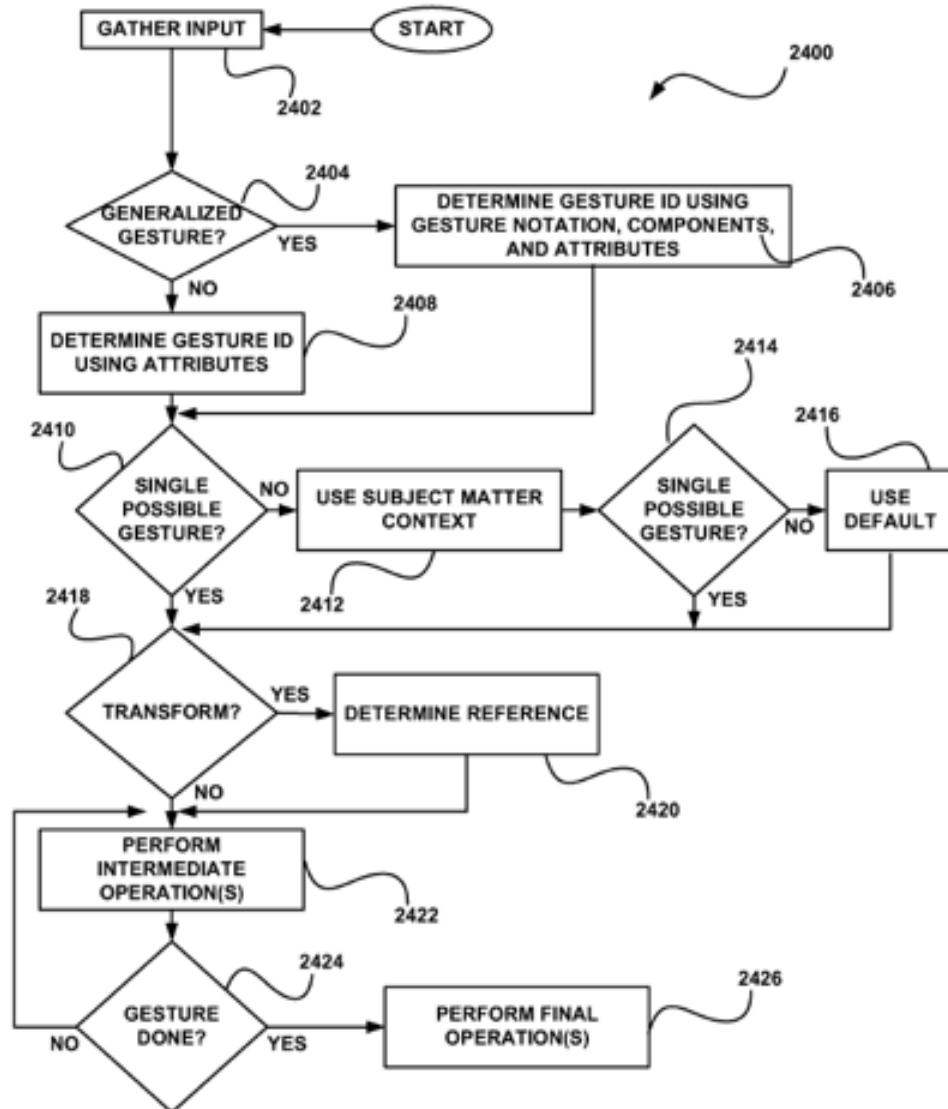
37. As further explained, a gesture may be “described by a phase space path which represents the time evolution of the touch state associated with a contact point” (EX1001, 72:31–34). Transitions between states may depend on changes in the contact point or the time spent on each leg of the gesture (EX1001, 72:37–64). In effect, gestures are not simply static motions but are modeled dynamically based on their behavioral trajectory through this space. This phase space abstraction allows for the recognition of complex, multi-stage gestures that may follow similar initial paths but diverge into distinct outcomes. Exemplary pattern-based gesture “pattern-based gesture 2302” is “described by a phase space path which represents the time evolution of the touch state associated with a contact point” on the touch screen. EX1001, 72:31-34; EX2008, ¶[00566].



EX1001, FIG. 23; EX2008, FIG. 28.

38. Each state, and transitions between states, may be defined by factors including “contact point changes” and “the time spent on each leg of the path through phase space.” EX1001, 72:37-64; EX2008, ¶¶[00567]-[00570]. Consistent with the “decision tree” aspect of these pattern-based gestures, multiple patterns may share the same starting (and intermediate) points. EX1001, 75:57-65; EX2008, ¶[00581].

39. To detect these pattern-based gestures, the ’918 Provisional, which the ’413 Patent incorporates by reference (EX1001, 1:17-19), discloses “gesture differentiation,” which is distinguished from simply detecting a “basic user interaction (i.e., button press, basic application interaction, etc.),” techniques. EX2008, ¶¶[00498], [00502]. The process begins by “determine[ing] whether there is any meaningful user input,” including applying a “threshold above which user interaction may rise before it may be deemed meaningful.” *Id.*, ¶¶[00501]-[00502]. Detecting the gesture may involve “gesture subject matter context,” including “objects displayed at or near the location of the gesture.” *Id.*, ¶[00512]. For example, the starting point may be the beginning of a touch contact at a particular location of a specific object on the touch screen. *Id.*;



EX2008, FIG. 24.

40. One example of a context-specific gesture starting at a particular location on the touch screen is a “slide gesture which begins near an edge of a display and then moves inward [that] cause[s] a menu or window to slide in from that edge of the display.” EX1001, 67:32-35. Moreover, “there may be different menus or windows associated with each edge of a display.” EX1001, 67:37-38. Such a menu may include menu items that function as UI control widgets (e.g., buttons, drop

down lists, or any other non-textual UI widget). EX2008, [0164], [0194]. User selection of a UI control widget may include establishing “one or more primary conditions to establish that the user intends for that particular multi-state widget to be selected, and a secondary condition, to establish the state of that widget.” EX2008, [0178]. “Once the selection establishment conditions have been satisfied, the state of the UI control widget may be set.” EX2008, [0182]. For example, “a user may make contact with . . . an interaction surface, with a contact point near a drop down menu [UI widget] . . . [and] may increase contact pressure until the drop down menu is highlighted.” EX2008, [0183]. Then, “[a]fter maintaining that contact pressure level for a few seconds, the list of menu items becomes visible, after which the user may select an item by varying the contact pressure.”

41. Again, as indicated by FIG. 24, multiple complete full patterns may share the same starting (and intermediate) points. Consistent with the multi-part nature, the device may perform “intermediate operations associated with the identified gesture.” *Id.*, ¶[00521]. “[E]ach execution of intermediate gesture operations may trigger a display update (i.e. rendering, etc.).” *Id.*, ¶[00524]. “The intermediate operations are continuously performed unless it is determined that the gesture has ended.” EX2008, [0525]. One purpose of these intermediate gestures is so “a user may receive feedback on their gesture before terminating the interaction.”

Id., ¶[00522]. “[C]ues to provide feedback to the user . . . may include 3D or pseudo-3D effects.” EX2008, [0538]-[0539].

42. One example of a pseudo-3D effect used to provide feedback to the user is “a 3D layered user interface implemented in a device having a single layer displays [*sic*] through the use of virtual display layers.” EX2008, [0793]. Specifically, “a 3D layered user interface may have a virtual foreground display layer and a virtual background display layer.” *Id.* “[V]irtual display layers may be given the appearance of depth through the use of 3D depth cues . . . [such as] a blur operation, such that layers located at successively greater depths may appear blurrier than those closer to the user.” EX2008, [0795]. Additionally, “[a] degree of translucency [e.g., blur] may be a function of a user input.” EX2008, [0848].

43. A “gesture may be ended by ending one or more user inputs” or “by performing a different gesture” at which time “the final operations associated with the identified gesture are performed.” *Id.*, ¶[00526]-[00528]. “For example, in one embodiment, a gesture which triggers the display of a pop-up menu and allows a user to cycle through menu items may have as a final operation the selection of the current menu item and removal of the pop-up menu.” *Id.*, ¶[00529].

A. Independent Claims 1 and 101

44. Independent claims 1 and 101 of the ’413 Patent are each directed to an “electronic device.” EX1001, 95:42-96:33, 112:37-113:29.

45. Claim 1 recites **[1a]** An electronic device, comprising: a display; a touch interface; one or more processors; memory; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for:

[1b] displaying, utilizing the display, a graphical user interface;

[1c] when the graphical user interface is displayed, detecting, utilizing the touch interface, a first gesture that begins in connection with a first edge of the display and moves inward;

[1.d] in response to the detection of the first gesture that begins in connection with the first edge and moves inward,

[1.d.i] displaying, utilizing the display, a first menu as sliding in and including one or more first menu items, and

[1.d.ii] blurring at least a portion of the graphical user interface such that a magnitude of the blurring of the at least portion of the graphical user interface progressively increases as a function of an increase in a magnitude of the first gesture being detected;

[1.e] when the first menu is displayed including the one or more first menu items:

[1.e.i] detecting, utilizing the touch interface, a first duration of contact on at least one of the one or more first menu items,

[1.e.ii] in response to the first duration of contact on the at least one of the one or more first menu items being detected to not surpass a threshold, performing a first operation, and

[1.e.iii] in response to the first duration of contact on the at least one of the one or more first menu items being detected to surpass the threshold, performing a second operation;

[1f] when the graphical user interface is displayed, detecting, utilizing the touch interface, a second gesture that begins in connection with a second edge of the display and moves inward;

[1g] in response to the detection of the second gesture that begins in connection with the second edge and moves inward, displaying, utilizing the display, a second menu including one or more second menu items;

[1.h] when the second menu is displayed including the one or more second menu items:

[1.h.i] detecting, utilizing the touch interface, a selection contact on at least one of the one or more second menu items, and

[1.h.ii] in response to the selection contact being detected on the at least one of the one or more second menu items, performing a third operation;

[1i] wherein the electronic device is configured such that the graphical user interface is displayed in at least one virtual display layer, and at least one of the first menu or the second menu is displayed in at least one other virtual display layer.

EX1001, 95:41-96:33

46. Claim 101 includes limitations similar to the Limitations [1a]-[1i] of claim 1 and clarifies that [101d] the one or more first menu items include a particular application icon. *Id.*, 112:37-113:29. Claim 101 also recites [101.e.iv] where the second operation is performed in response to the first duration of contact on the particular application icon being detected to surpass the threshold, and the first duration of contact being detected to include a slide gesture component after the first duration of contact on the particular application icon is detected to surpass the threshold. *Id.*

VI. OVERVIEW OF THE PRIOR ART

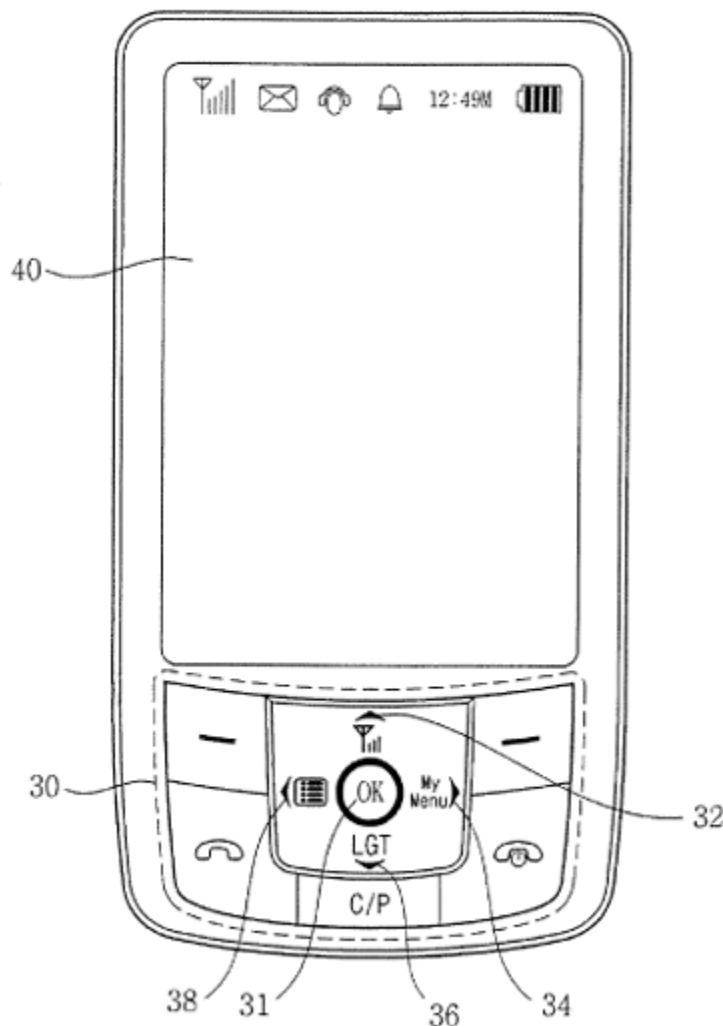
47. Dr. Terveen primarily relies on three cited references for his invalidity analysis. *See* EX1003, ¶56 (the Ahn-Chaudhri-Hinckley combination).

A. U.S. Patent App. No. 2008/0207188 (“Ahn”)

48. U.S. Patent App. No. 2008/0207188 is titled “Method of displaying menu in a mobile communication terminal” and lists as its first inventor Hye Sang Ahn. I will refer to this reference as “Ahn.”

49. As depicted in FIG. 2 (inset below), Ahn discloses a mobile communication terminal that includes a display 40 along with a key input device 30 and a touch input device (e.g., element 10 in FIG. 1). EX1004, Abstract, ¶[0027].

FIG. 2

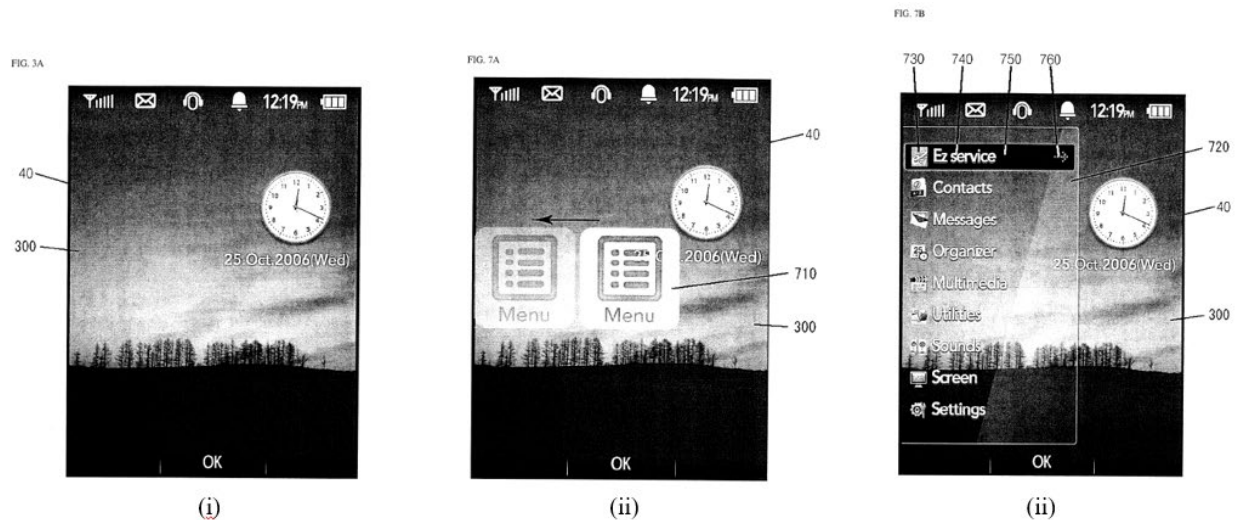


50. Ahn's proposed operation attempts to solve the problem where device characteristics are unavailable when a menu list is displayed. EX1004, ¶¶[0005]-[0006]. In conventional browsers, the standby screen disappears when a menu screen is displayed. Thus, Ahn proposes to provide menus that only take up a portion of the display so that the standby screen can still be displayed concurrently with the menu.

51. Against this backdrop, Ahn discloses a method of displaying menus in a mobile communication terminal, where each menu can correspond with a selection of one of a plurality of specific directions associated with the menus. EX1004, Abstract ("menu corresponding to the selected specific direction on the standby screen"). Ahn discloses three separate methods for displaying its menu via the use of (1) directional keys, (2) a simple touch and drag operation, and (3) via a start position operation. In each method, the menu is displayed after a corresponding input (e.g., button click or drag) is completed. That is, the display of the menu is a binary operation—appearing only when the input is satisfied. None of Ahn's menus are described as dynamically following a user's finger.

52. In a first embodiment, Ahn describes a method of displaying a menu via a selection of a specific direction on a direction key (the "Directional Key" embodiment). *See* EX1004, ¶¶[0030], [0034]-[0035].]. As shown in FIGs. 7A-7C, a menu (e.g., setting menu 720/820) can be accessed "when a leftward direction key

is selected.” *Id.*, ¶[0062]. Ahn describes “if the user selects the leftward direction key 38 . . . *[t]hereafter*, a basic setting menu list 720 . . . is displayed”. EX1004, ¶[0063]-[0064]¹; FIG. 2 (showing directional keys).



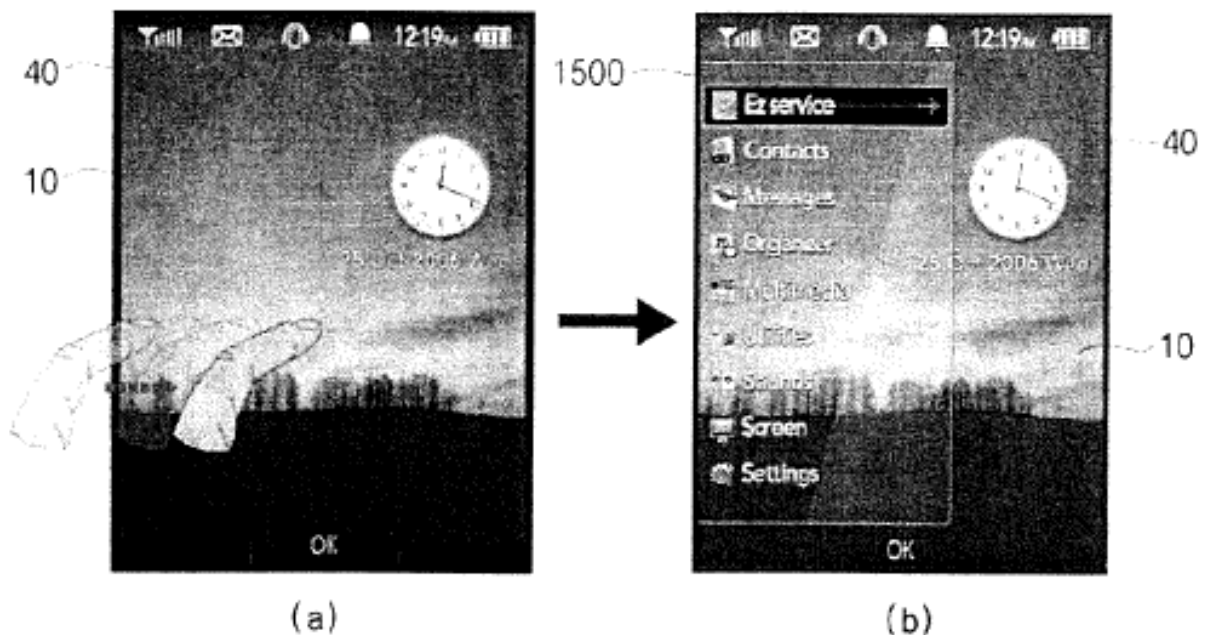
53. Ahn refers to an input of the its direction keys (e.g., 34, 38) as a “click.” EX1004, ¶¶[0045]-[047]. The direction keys are single input buttons that can only receive one type of input and a POSITA would have recognized a selection (*i.e.*, click) of a direction key as an input used to initiate a binary operation. Accordingly, if the direction key is pressed, then the menu is displayed “thereafter,” and if the key is *not* pressed, the menu is *not* displayed. EX1004, ¶¶[0051], [0060], [0064]. Accordingly, the opening of Ahn’s menu is performed in a binary manner rather than progressively.

¹ All emphasis added unless otherwise noted.

54. In another embodiment, Ahn describes a method of displaying a menu via “a touch and drag operation of a specific direction” (the “Touch and Drag” embodiment). *Id.*, ¶[0101], *see also* ¶¶[0102]-[0104]. The menus of Ahn’s “Touch and Drag” embodiments are identical to the menus of Ahn’s “Directional Key” embodiments, only differing with regard to the method of initiating their display. *Compare* EX1004, FIGs. 4B, 6B, 7B, *with* FIGs. 13B, 14B, 15B. The opening of the menu’s according to Ahn’s touch and drag operation is also performed in a binary manner, as explained below.

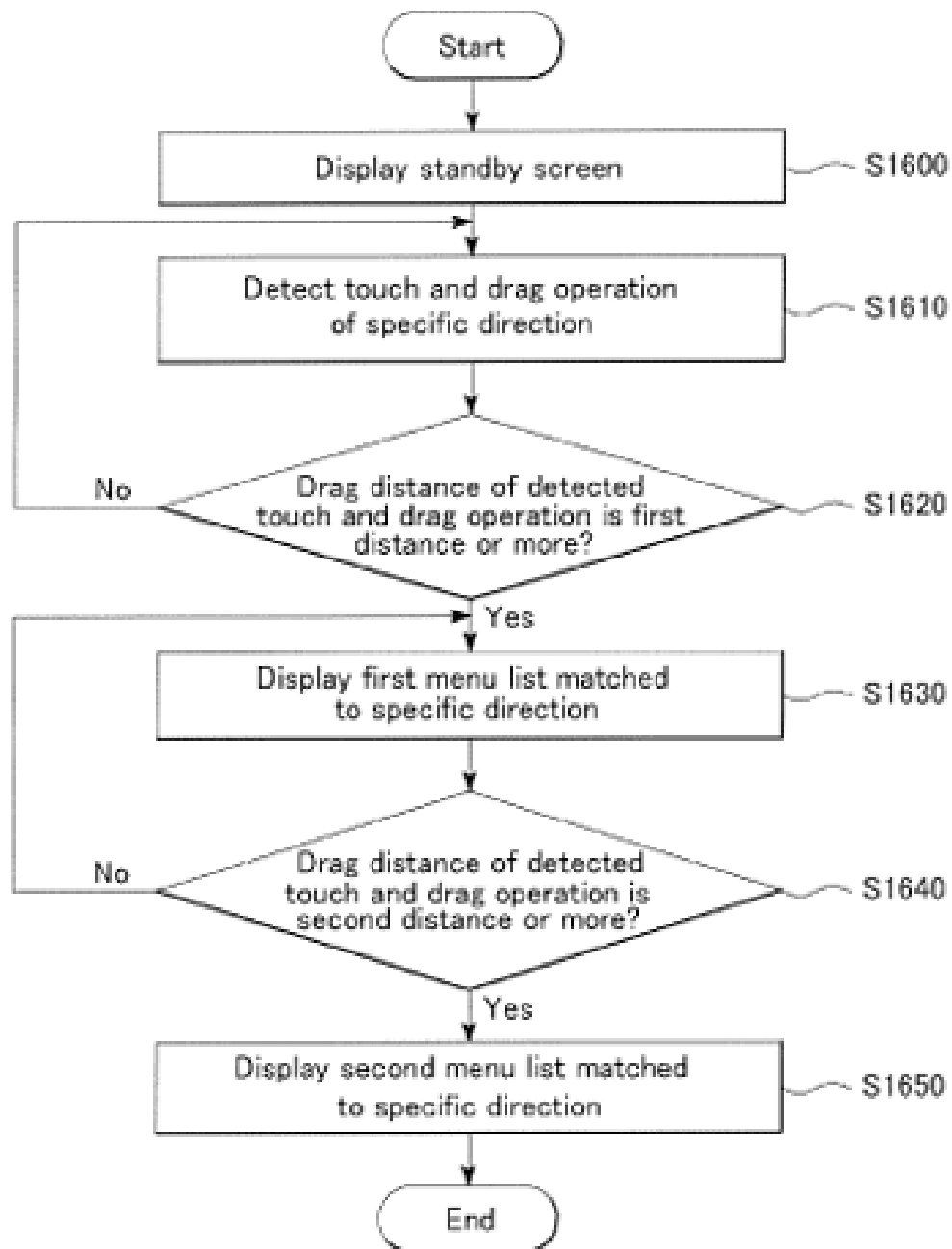
55. As shown in FIGs. 15A-15B of Ahn (inset below), “[a] controller 50 detects a touch and drag operation of a specific direction in the touch screen 10.” *Id.*, ¶[0107].

FIG. 15



Then “controller 50 determines whether a drag distance of the detected touch and drag operation is a first distance or more.” *Id.*, ¶[0109]. ***After*** the user performs the touch and drag gesture and the controller compares the drag distance to a threshold, the menu can then be displayed. *Id.*; *see also* FIG. 16 (detecting touch at step 1610 and displaying menu at step 1630)

FIG. 16

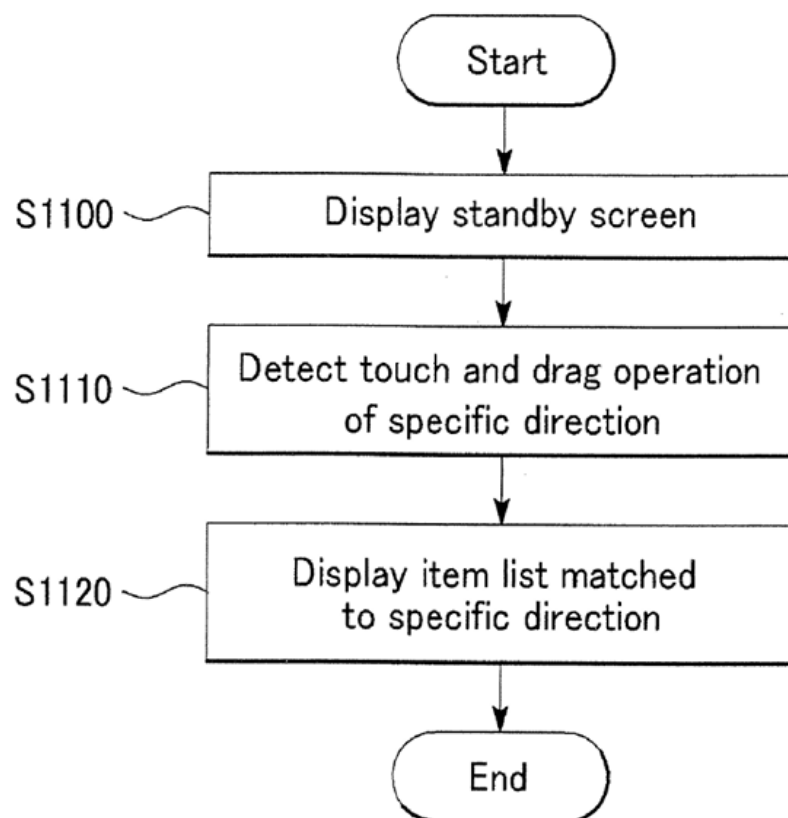


EX1004, FIG. 16. Ahn’s descriptions of its “Touch and Drag” embodiments do not include any disclosures that would lead a POSITA to understand the “Touch and Drag” menus to display any more progressively than the “Directional Key” menus.

See EX1004, ¶¶[0102]-[0103] (for example, “if the user . . . drags the touch in a left direction a menu list 1300 matched to a left direction is displayed in the display 40”). Therefore, a POSITA would recognize that, despite being initiated via a different type of user input, Ahn’s “Touch and Drag” menus are displayed via the same binary process by which its “Directional Key” menus are displayed.

56. This operation is also depicted in Ahn’s FIG. 11, which shows that the display of the item list is after the operation is detected (*i.e.*, completed).

FIG. 11



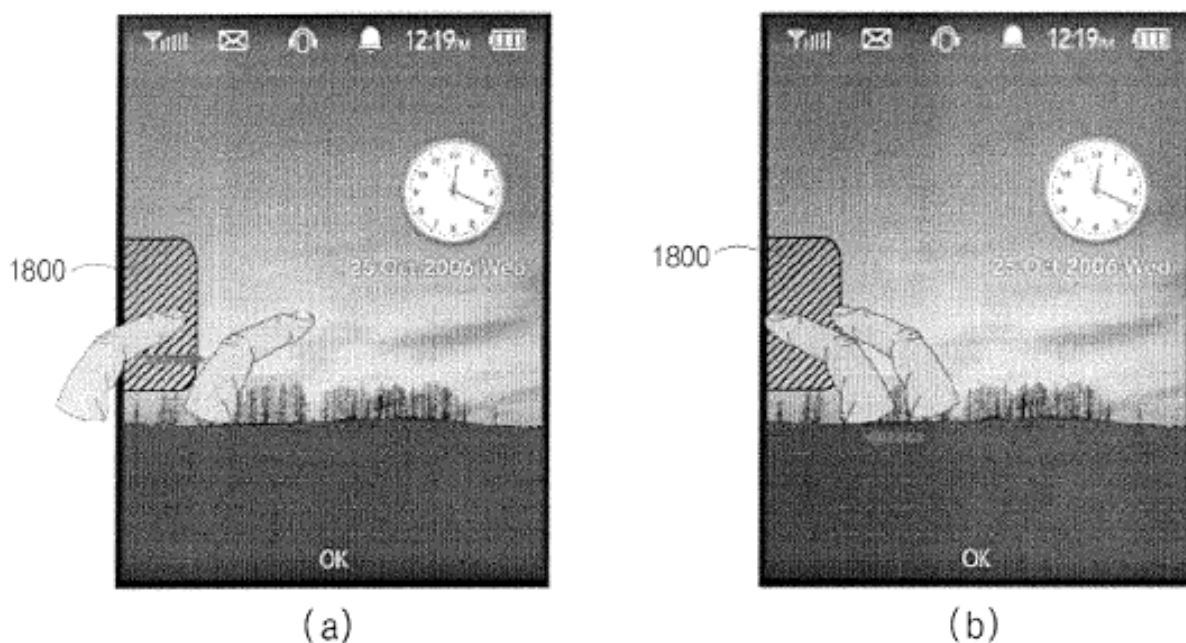
To the extent it is argued that Ahn’s FIG. 11 operates differently than FIG. 16 by displaying a menu before the “touch and drag operation” is completed, this is

inconsistent with Ahn. While paragraph [0099] states “[a]s the touch and drag operation is detected, the controller 50 controls to display an item list matched to the specific direction in the display 40 (S1120),” a POSITA would not understand the phrase “[a]s the touch and drag operation is detected” to mean the menu is displayed before the operation is complete. Ahn uses this phrase several other times to mean that the menu is displayed after the operation is completed. *See e.g.*, EX1004, ¶¶[0016], [0071], [0092], [0094]. For example, paragraphs [0094] of Ahn describes the Directional Key embodiment where “the user setting menu 420 is displayed in the right area of the standby screen 300 *as the user selects* the rightward direction key 34.” Of course, consistent with my paragraphs 52-53 above, Ahn’s device is configured such that menu 420 is displayed “after” the selection (EX1004, ¶[0050]-[0051]) even though it may appear fast enough for the user to think this occurs “as” the selection is happening. Similarly, Ahn’s FIG. 11 shows displaying the menu *after* the touch and drag operation is detected (*i.e.*, completed) even though the menu may be displayed “as” the user is still dragging their finger on the screen. *See* EX1004, ¶¶[0102]-[0103] (using conditional “if” clauses to describe the drag gestures like the clauses used to describe operation of FIG. 16 in Ahn’s paragraphs [0109]-[0110]).

57. In a further embodiment, Ahn states that this touch and drag operation can be used with “an identifier” such as “a specific icon for notifying a start position

of the touch and drag operation.” Ahn at ¶[0117]. This operation does not diverge from the binary operation described above and simply adds an icon so that a user is notified that a corresponding menu is available, as shown in FIGs. 18a-b (inset below). EX1004, ¶[0117] (describing that “[i]n the above-described implementations, if the touch and drag operation of a specific direction is performed . . . an identifier for notifying the user that an item list matched to the specific direction . . . can be displayed in the display 40.”)

FIG. 18



EX1004, FIG. 18a-b.

58. As shown in FIG. 18a, the user may perform the same touch and drag operation on the graphical shadow to display a menu 1500 as is done with the touch and drag operation depicted in FIG. 15. Ahn at ¶[0118]. Alternatively, as shown in

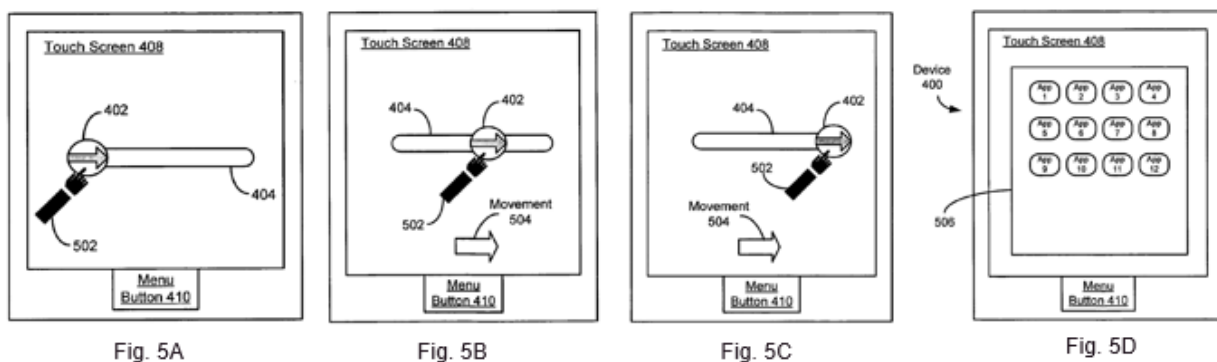
FIG. 18b, the user may press the graphic 1800 and, in response, “the menu list 1500 can be displayed as emerging from the left side of the display 40.” *Id.*, at ¶[0119]. In either scenario, the menu will begin to open when the gesture—either after a drag operation in FIG. 18a or a tap operation in FIG. 18b—is completed. At no point does Ahn describe that its menu (e.g., 1500) will emerge and be displayed in some in-between state where the menu is neither fully open or closed. EX1004, ¶¶[0118]-[0119]. Ahn’s use of the “as the user” language does not indicate that the menu is displayed for the same reasons as discussed in my paragraph 56. *See e.g.*, EX1004, ¶[0094]. For example, a POSITA would have understood that “as the user may press the graphic 1800 with a finger, the graphic 1800 gradually disappears in the display 40” (EX1004, ¶[0117]) indicates programming logic that triggers an animation after an input (e.g., tap) occurs and does not indicate that the animation relies on the user to progressively hold this “press” for the duration of the animation.

59. Thus, all of Ahn’s embodiments operate in a binary manner, opening the menu after completion of the gesture. Because Ahn does not explain that the addition of a start position identifier changes the menu display process, a POSITA would have understood that Ahn’s “Touch and Drag” menu embodiments that include start position identifier’s display menus via binary operations like the rest of Ahn’s “Touch and Drag” embodiments.

B. U.S. Patent App. No. 2007/0150842 (“Chaudhri”)

60. U.S. Pat. Pub. No. 2007/0150842 is titled “Unlocking a device by performing gestures on an unlock image” and lists Imran Chaudhri as its first inventor. I will refer to this reference as “Chaudhri.”

61. Chaudhri-842 discloses “[a] device with a touch sensitive display [that] may be unlocked via gestures performed on the touch-sensitive display.” EX1005, Abstract. The primary focus of Chaudhri-842 is a method for “unlocking [a] device” via a “predefined gesture,” where the method includes “moving [an] unlock image to a predefined location and/or moving the unlock image along a predefined path.” EX1005, Abstract. Chaudhri shows progress of such an unlocking gesture by the distance an indicator 402 (shown in FIGs. 5A-5C) moves toward an “end” of a channel 404. EX1005, ¶[0071] (The end of the channel 404 “serves as a predefined location to which the unlock image 402 is to be dragged.”).



EX1005, FIGs. 5A-5D.

62. As shown in FIGs. 5A-5D (above), this operation starts when a user “touch[es] a touch screen 408 of device 400 with her finger 502.” *Id.*, ¶[0074]. Specifically, “[t]he user touches the touch screen 408 at the location corresponding to the unlock image 402, which is located initially at the left end of the channel 404.” *Id.* When the user drags their finger “in the direction of movement 504 [...] [t]he unlock image 402 is dragged along the channel 404 as a result of the gesture.” *Id.*, ¶[0075]. When “the user has dragged the unlock image 402 to the right end of the channel 404 [...] [and] releases the unlock image 402 [...] the unlock action is complete. *Id.*, ¶[0076]. When the unlock action is complete, “the device unlocks and displays on the touch screen 408 user-interface objects associated with normal operation of device 400[,]” such as menu 506. *Id.*

C. U.S. Patent App. No. 2011/0209093 (“Hinckley”)

63. U.S. Pat. Pub. No. 2011/0209093 is titled “Radial menus with bezel gestures” and lists as its first inventor Kenneth P. Hinckley. I will refer to this reference as “Hinckley.”²

² While Apple’s Petition and Dr. Terveen’s Declaration (EX1003) both in their references to EX1006 incorrectly spell the name of the first inventor Kenneth P. Hinckley as “Hinkley”, we use the correct spelling here and intend our correctly

1. Bezel Gestures

64. Hinckley discloses a device including a bezel “used to extend functionality that is accessible through the use of so-called bezel gestures.” EX1006, Abstract. With these bezel gestures, the user can provide inputs to the device by interacting with bezel 303, rather than the touchscreen of display 308. EX1006, FIG. 3, ¶¶[0059]-[0069] (Use of Bezel as an Input Mechanism). Hinckley’s FIG. 3 shows an example of this where the device receives an input associated with the bezel, which may be “a variety of different types of recognizable inputs (e.g., taps, tap combinations, tap/hold combinations, slides, etc.).” EX1006, ¶[0066].

spelled references to “Hinckley” to map to the incorrectly spelled references to “Hinkley” in Apple’s Petition and Dr. Terveen’s Declaration.

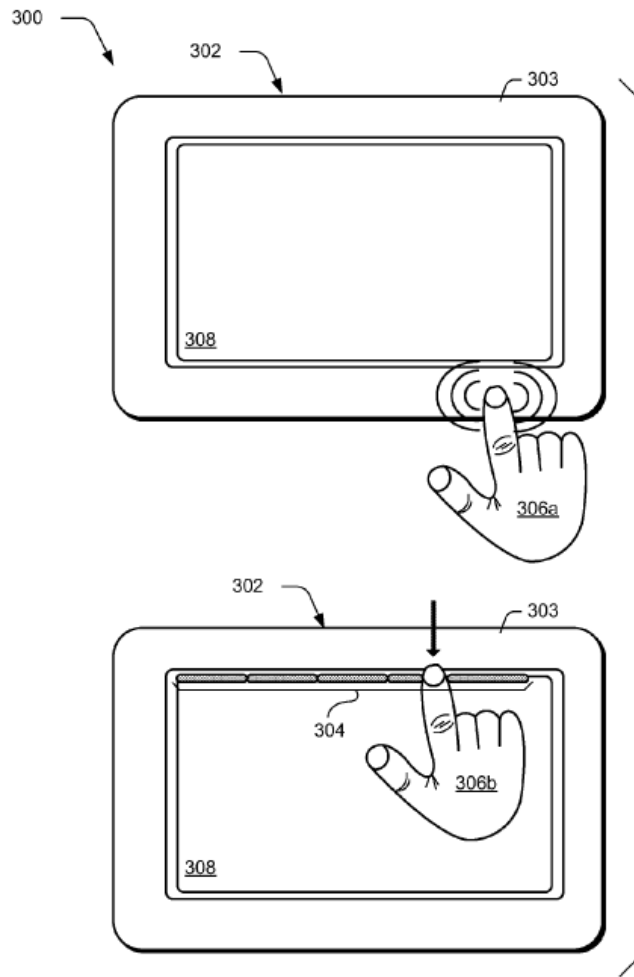


Fig. 3

EX1006, FIG. 3. It should be noted that the “variety of different types of recognizable inputs” as described in paragraph [0066] of Hinckley describes inputs on the bezel 303 and does not refer to using different types of inputs on menu icons (e.g., 1714). See EX1006, [0059] (header indicating paragraph [0066] is describing “Use of Bezel as an Input Mechanism”), FIG. 4 (“input associated with bezel”).

65. As shown in FIG. 6 (inset below), Hinckley describes specific embodiments in which “the bezel menu can be accessed through a bezel gesture in which a finger of user’s hand 606 touches the bezel and then moves across the bezel and onto the display device 608 [...] which allow[s] bezel menu 610 to be dropped down [...].” *Id.*, ¶[0073].

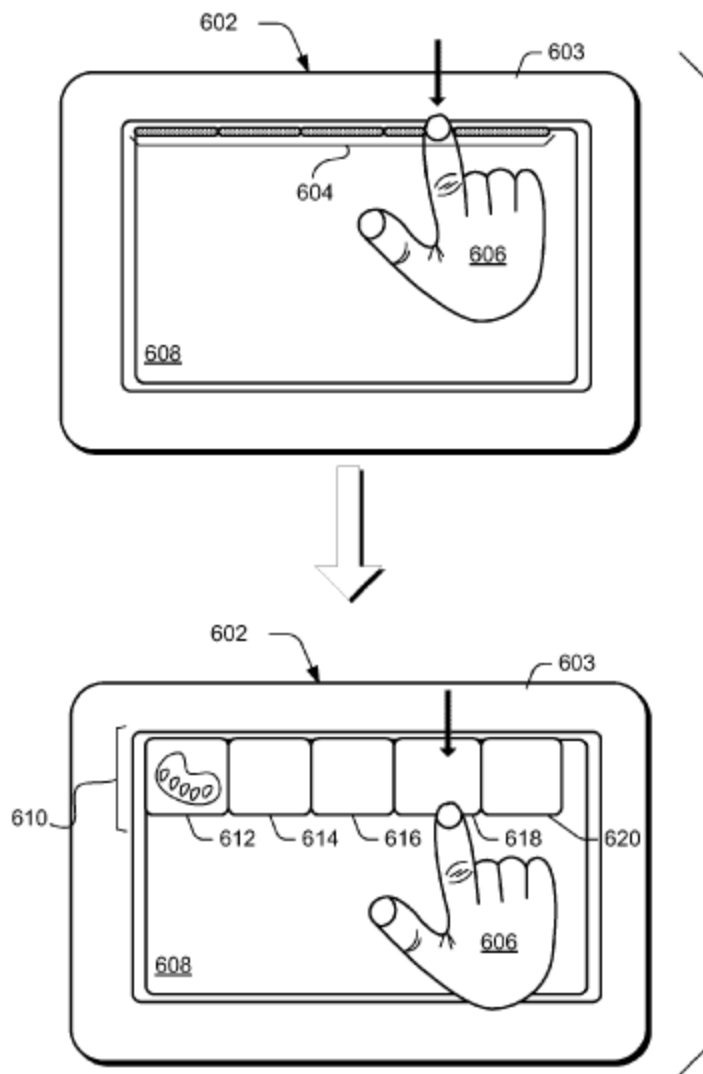


Fig. 6

EX1006, FIG. 6.

66. The bezel menu 610 includes selectable icons or slots “associated with ... different functionalit[ies] such as, for example, paint functionality, pen

functionality, note functionality, object creation, object editing, and the like.” *Id.*, ¶[0074]. These functionalities can be accessed via two different gestural modes—a novice mode and an expert mode. EX1006, ¶[0076].

67. The novice mode utilizes the bezel menu 610, shown in FIG. 6 above. In this mode, a gesture is used to reveal the bezel menu 610 and the user may then tap on a desired item associated with one of the icons or slots 612, 614, 616, 618, and 620 associated with the bezel menu. *Id.*, ¶[0077]. In this mode, the user can lift their finger after opening the bezel menu 610 and the bezel menu will remain open while the user decides which icon to select. *See id.*

68. The expert mode does not require use of the bezel menu and, instead, relies on the user to be familiar with the location of the icons in the bezel menu. EX1006, ¶¶[0078]-[0079] (“bezel menu may not necessarily be revealed at all in order to access functionality associated with an icon”). Instead of revealing the bezel menu and tapping on an icon, the user can perform a continuous finger-drag that crosses through the location of the slot or icon to access the function in a single transaction. *Id.*, ¶[0078]. “Hence, once users are familiar with a particular operation on the bezel menu, they can rapidly drag through it to create and position an object without ever having to be distracted by the opening of the visual menu itself.” *Id.*, ¶[0079].

2. *Radial Menu*

69. As shown in FIG. 17 (inset below), Hinckley further describes “radial menus [that] can be used in connection with menus such as bezel menus.” *Id.*, ¶[0105].

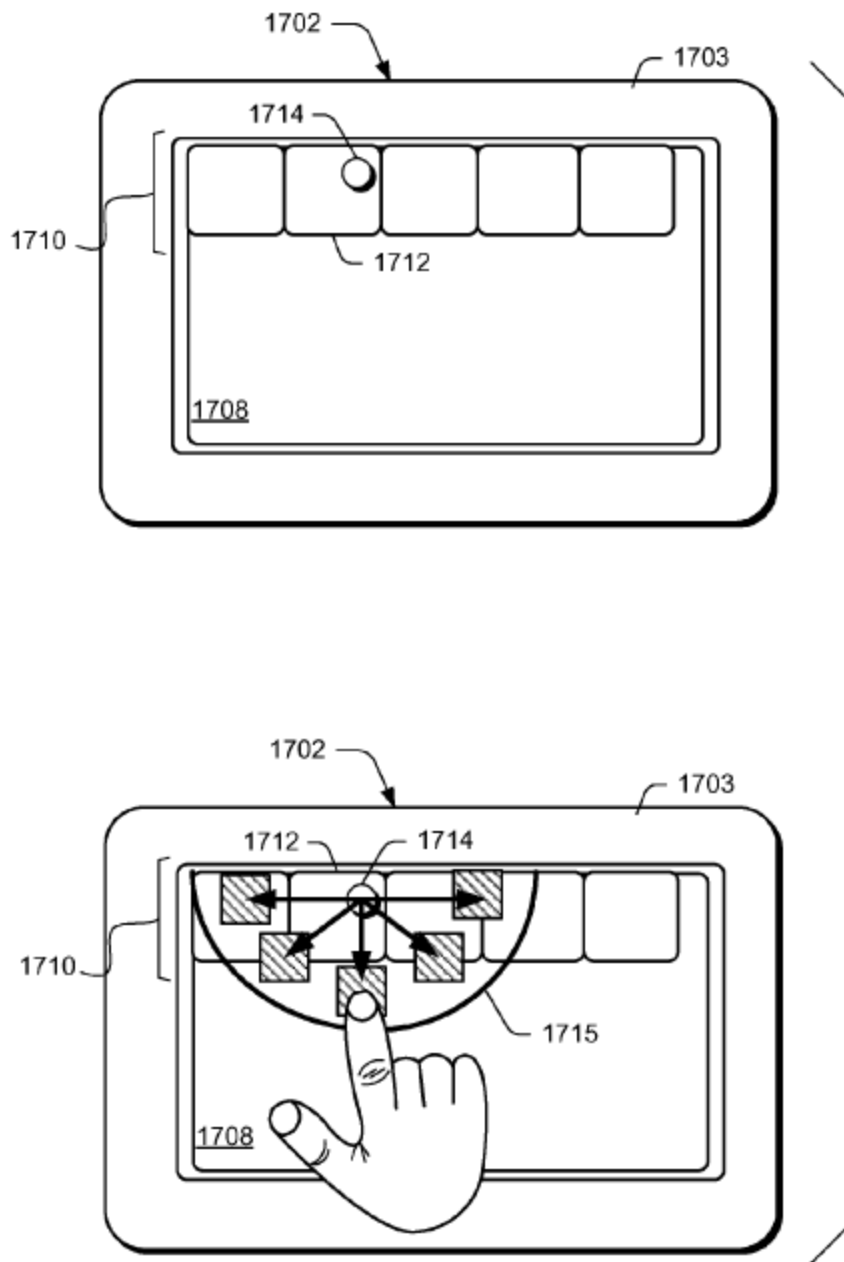


Fig. 17

70. As an example, Hinckley describes embodiments in which a bezel menu may include bezel menu icons or slots (e.g., icon or slot 1712) that further

include an additional radial menu icon object (e.g., radial menu icon 1714). *See id.*, ¶[0107]. Further, Hinckley describes that a radial menu 1715 can be accessed via gestures on radial menu icon 1714. *See id.* Specifically, gestures for accessing radial menu 1715 may include “hovering a pen over or near the radial menu icon 1714[,] [...] pull[ing] down the radial menu 1715[,] [...] [or] a tap-and-hold of the pen or finger on or near the radial menu icon 1714.” *Id.* When “radial menu 1715 is exposed, the user can access various functionalities or commands by touching down on or near radial menu icon 1714 and stroking in a particular direction.” *Id.*, ¶[0108]. Similar to the operation described above with reference to bezel menu 610, the menu 1710 along with icons 1712 and radial menu icons 1714 can be tapped when in the novice mode; however, in the expert mode “the radial menu might not be exposed at all.” EX1006, ¶[0110].

D. Radial Menus

71. Radial menus, also known as pie menus, trace back to user interface research in the 1980s, particularly at institutions like the University of Toronto and Xerox PARC, where human-computer interaction was being studied. Radial menus were used early on in CAD programs, well before they became popular in gaming interfaces. Such menus were used in CAD programs because they reduced mouse travel and could be more ergonomic and faster than traditional drop-downs or ribbons, especially when selecting between tools that would be controlled with the

mouse. In gaming, radial menus serve another purpose: quickly selecting weapons, actions, or dialogue options, using a mouse or external analog stick.

72. Radial menus were rarely implemented outside of certain contexts (e.g., video games, Computer aided drafting (CAD programs)) because they were not generally applicable to all functionalities or interfaces.³ EX2012, 4 (“Applications for pie menus are rare”); EX2009⁴, 1-2 (experimentation with pie menus has not led to implementation into any core browser features), 4-5 (warning pie menus don’t make all applications better and trying to implement pie menu for an application but finding no good solution). Possible applications for radial menus included

³ Marking menus, which were similar to radial menus, and included additional complexities were also used in certain contexts, like CAD programs, but are not referenced in this record and distinct from the radial menu described in Hinckley.

⁴ This exhibit is authored by Jonathan (“Jono”) Xia (formerly Jonathan DiCarlo) who worked at Mozilla Labs and Mozilla User Research, specifically in the context of “user-interface design.” EX2016, 1-2. Mozilla was a well known company in the human-computer interaction (HCI industry) and this exhibit would have been consulted by POSITAs under Dr. Terveen’s definition (EX1003, ¶30) as is illustrated with the hundreds of user comments on the article discussing implementation of User interface designs. EX2009, 5-37.

interactive interface like video games (e.g., The Sims or weapon wheels—where a weapon can be replaced with a different weapon using the controller analog stick), Computer Aided Drafting (CAD) programs, or specific web applications (e.g., Songza or in a mail application). EX2012, 4; *see also* EX1006, ¶[0106], FIG. 17 (disclosing radial menu for interactive drawing or image processing application). But even the use of radial menus in very well thought-out interfaces encountered issues. *See* EX2012, 4 (pie menu options “are relatively hard to hit and interfere with dismissing the menu”). Thus, a POSITA would have been well-aware that substituting an established menu for a radial one was not simple and actually quite unpredictable as radial menus were notoriously difficult to implement.

73. The difficulty implementing radial menus can be highlighted by their known disadvantages, which included that these menus were limited in the number of options they could display and would take up greater space as compared to conventional menus. EX2010⁵, 2 (full radial menu limited to eight choices and radial menus take up more screen space); EX2009, 4 (problems arise when there are

⁵ This exhibit is authored Ben Meadowcroft who has a Masters in Software Engineering” and 7 years’ experience in technical roles. EX2011, 1. This exhibit would have been consulted by POSITAs under Dr. Terveen’s definition (EX1003, ¶30).

too many items in a pie menu); EX2013 at 2 (“pie menus consume greater screen area and become polynomially larger than linear menus in both height and width with increased item size and number of items”). For these reasons, radial menus were typically limited to large screen applications, like video games, CAD programs, and web applications that were displayed on a television or desktop monitor with a separate controller (e.g., joystick or mouse). See EX2012 4 (“Pie menus mustn't be used where they cannot be placed around the mouse cursor.”). To illustrate, Jack Callahan 1988 study provides a comparison between linear menus and pie menus that depicts the significant additional space required to implement a pie menu.

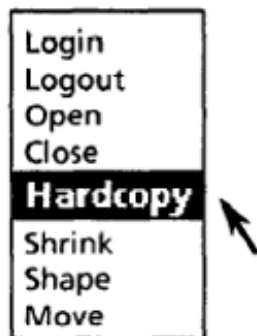


Figure 1: A typical linear menu

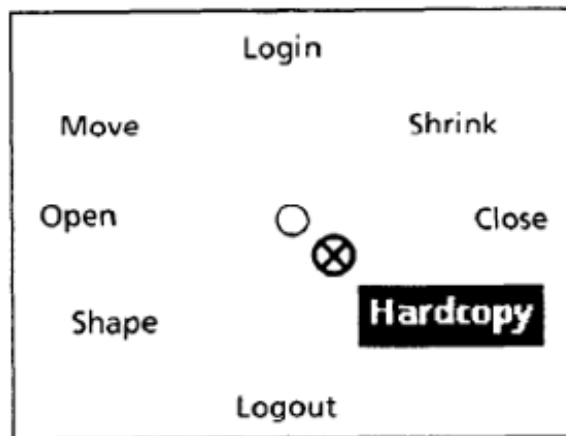


Figure 2: A crude pie menu

EX2013, 1. Thus, a POSITA would have understood that the implementation of radial menus would face more complications on smaller interfaces, like cellphones, where space was limited or in interfaces without external controls.

74. Additionally, and independent from the excess space requirement, the options that can be used with a radial menu will be limited to eight choices for a full

360-degree radial menu.⁶ EX2010, 2. While additional options past eight are technically possible, a POSITA would not have pursued such options as they would have eliminated the benefits of using the pie menu in the first place. *See* EX2010, 2 (pie menus **will not** scale well above eight choices); EX2009, 4 (when the wedges get too small, you lose the associated benefits).

75. Thus, while a POSITA would have understood that some aspects of radial menus seemed promising, these beneficial features often failed to carry over to the actual implementation. EX2013, 6; EX2009, 4-5. For example, radial menus were thought to be faster to select because of the low distance (D) to selection when the menu appears equidistantly around the cursor. EX2009, 2 (Fitts's Law: $\text{Time} = a + b \cdot \log_2(D/W + 1)$); EX2013, 5. But this theory rests on the assumption that the

⁶ While this restriction references a full, circular radial menu, the same restriction would apply to half-radial menus and quarter-radial menus for at least the same reasons. Half-radial menus (semi-circle) would be limited to five options and quarter-radial menus (quarter-circle) would be limited to three options. *See e.g.*, EX1006, FIG. 17, ¶[0109] (half-radial menu has five options and radial menu on associated with end icon (e.g., quarter-radial menu) would have fewer (e.g., three) options); EX2005, 90:9-92:2 (for radial menus associated with an end icon “half of the semi-circle wouldn't be possible to display” so “you might display fewer items”)

‘slice’ of the icon occupying the pie menu is large enough to select, and the diameter (W) of the menu is not constrained. As shown in the illustration below, Fitts law (1) does not account for the circumferential width as it assumes this will be large enough to easily select, and (2) that the radius of the menu “occupies the largest possible area.” EX2009, 2 (“pie slice area effectively extends all the way to the edge of the screen, making W very large indeed”).

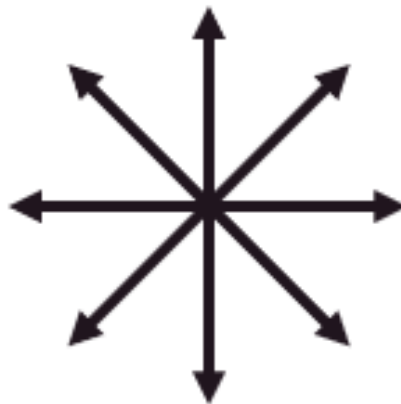


EX2009, 2.

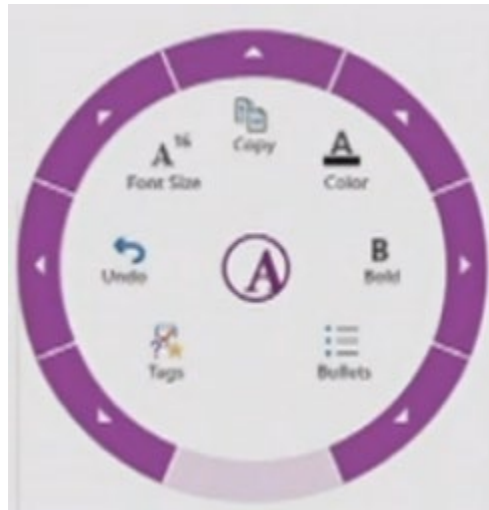
76. However, in practice, these advantages disappear when the radius (W) is constrained by space (e.g., in a cellphone) and when additional options are added, thereby decreasing the surface area of the selectable region. *See* EX2009, 2 (“bigger things are easier to hit than smaller things”). Further, the implementation of only portions of radial menus would serve to further aggravate these issues. *See e.g.*, EX2012, 4 (“Pie menus mustn't be used where they cannot be placed around the

mouse cursor” and “Don't use pie menus near the border where a part of the menu may be cut off.”)

77. Radial menus were also thought to benefit because the user would learn to associate a direction of the selection with the assigned functionality. Thus, as a user used became familiar with the menu, they could select an option without looking (also known as an expert mode). EX2012, 2; EX2009, 3; EX1006, ¶[0110]. However, this habitual benefit was known to be limited to only certain arrangements with fewer than eight options. *See* EX2009, 4 (when the wedges get too small, you lose the benefit of being able to select one without looking at it). To illustrate, typical layouts followed a “compass format” (i.e., N, NE, E, SE, S, SW, W, NW) that users would be familiar with (shown below), and, thus, the user would easily be able to conceptualize the directions without looking.



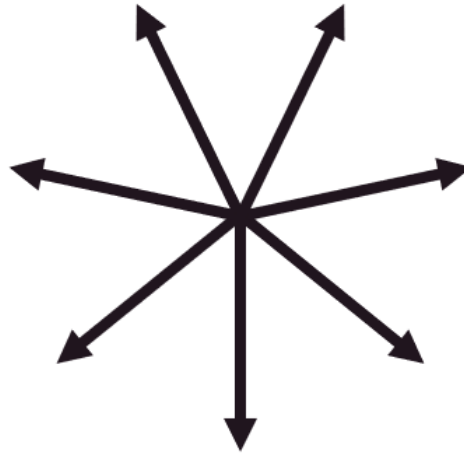
78. Moving away from this familiar structure would diminish the effectiveness of this habitual benefit, which is why a radial menu with only seven options would still use eight ‘slices’ and make one of them blank.



EX2014⁷, 1 (depicting radial menu having 7 options with 8 slices). This is because users are able to distinguish between different lateral directions (N, S, E, W or up, down, left, right) and a single option between each cardinal directions (NE, SE, SW, NW), but have difficulty if there are multiple options between each lateral directions. For example, when presented with a seven slice structure (produced in the image

⁷ This exhibit is authored by Kirill Grouchnikov, who is a “user interface engineer” with over a decade of experience in client-side development. EX2016, 2. Kirill Grouchnikov is a respected figure in the UI/UX space and this exhibit would have been consulted by POSITAs under Dr. Terveen’s definition (EX1003, ¶30).

below), the user would have trouble conceptualizing how to perform the gestures associated with the top four ‘slices’ without visual assistance.



79. The addition of additional options in the radial menu poses the same problems as well as further limiting the selection area. Thus, it is widely understood that radial menus of more than eight options should be avoided because they eliminate the benefits of these menus. EX2010, 2; EX2009, 4. The same holds true for the use of only portions of radial menus. Half radial menus are limited to five options and quarter radial menus are limited to three options. *See e.g.*, EX1006, ¶[0109].

VII. GROUND 1: DR. TERVEEN’S ANALYSIS FAILS TO SHOW OBVIOUSNESS OF CLAIMS 1-3, 7, 9, 10, AND 101 BY AHN, CHAUDHRI, AND HINCKLEY

80. Dr. Terveen’s analysis of Ahn, Chaudhri, and Hinckley fails to establish that Petitioner’s proposed combination would arrive at each of the claimed limitations.

A. Claims 1 and 101 are Not Obvious in view of Ahn, Chaudhri, and Hinckley

81. Independent claims 1 and 101 include similar limitations. *See* Section V.A. Dr. Terveen analyzes the limitations of claim 1 (Limitations [1a]-[1i]) and then incorporates this analysis for almost all the limitations of claim 101 (except for [101.d] and [101.e.iv]). EX1003, ¶¶151-168. The only additional analysis provided for claim 101 is with respect to Limitations [101.d] and [101.e.iv]. Therefore, while I only explicitly address the deficiencies of Dr. Terveen’s analysis of claim 1, these deficiencies must also apply to claim 101 with respect to all of the limitations of claim 101 except for [101.d] and [101.e.iv].

1. [1.e.i-iii], [101.e.i-iii]: detecting a first duration of contact on at least one of the one or more first menu items and in response [...]

82. Independent claims 1 and 101 recite one or more programs configured to be executed to perform certain functions on the device. Referring to Limitations [1.e] and [101.e], “when the first menu is displayed including the one or more first menu items” the programs “detecting, utilizing the touch interface, a first duration of contact on at least one of the one or more first menu items” and perform certain operations “in response to *the* first duration of contact on *the* at least one of the one or more first menu items” (emphasis added). Specifically, claim 1 recites that:

[1.e.ii] in response to the first duration of contact on the at least one of the one or more first menu items being detected to not surpass a threshold, performing a first operation, and

[1.e.iii] in response to the first duration of contact on the at least one of the one or more first menu items being detected to surpass the threshold, performing a second operation

Supra, §V.A (Limitation [1e.i-iii]); *see also* EX1001, 112:57-113:2 (same for independent claim 101).

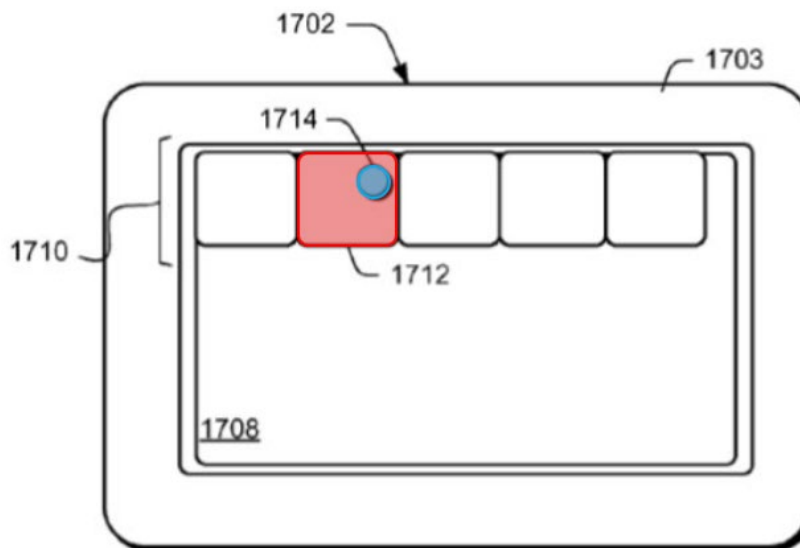
83. That is, Claims 1 and 101 require two scenarios that in response to a single contact (“*the* first duration of contact”) on menu item(s) (“*the* at least one ... first menu items”) “surpass[ing]” or “not surpass[ing]” a recited threshold. In the first scenario, the programs are executed to “perform[] a first operation” “in response to *the* first duration of contact on *the* at least one of the one or more first menu items being detected to *not surpass a threshold*” (emphasis added). EX1001, 96:4-7, 112:62-65. In the second scenario, the programs are executed to “perform[] a second operation” “in response to *the* first duration of [the same] contact on *the* [same] at least one of the one or more first menu items being detected *to surpass the threshold*” (emphasis added). EX1001, 96:8-11, 112:66-113:2.

84. These two Limitations collectively describe logical functions/steps to process instances of user input on a touch screen that begin from a common starting

point—the start of “a first duration of contact.” This is consistent with “decision-tree” and detection logic for detecting multi-part pattern-based gestures as described in connection with Figure 23 of the ’758 Patent and Figure 24 of the incorporated-by-reference ’918 Provisional as further discussed in Section V, above.

a) Dr. Terveen’s Mapping to Hinckley Does Not Perform Different Operations in Response to a Duration of Contact

85. Hinckley describes a bezel menu 1710 that includes “multiple selectable icons or slots 1712”, each of which “is associated with a different functionality.” EX1006, ¶[0106]. Hinckley also describes a separate icon—“a radial menu icon 1714 that gives a clue to the user that one or more radial menus [...] is associated with this particular icon or slot.” EX1006, ¶[0107]. Selecting this separate radial menu icon 1714 can give a user access to a radial menu that is not activated by interaction with the bezel menu icon 1712. *See* EX1006, ¶[0107] (“radial menus 1715 can be accessed through a tap-and-hold of the pen or finger on or near the radial menu icon 1714.”). Thus, as shown in the annotated version of Hinckley’s FIG. 17 (below), a POSITA would have understood that the bezel menu icon (1712 – shown in red) is a different object than the radial menu icon (1714 – shown in blue).



The selection of one object (e.g., icon 1712) will perform a different function than the selection of the other object (e.g., icon 1714). EX1006, ¶¶[0106]-[0107].

86. As discussed in Section VII.A.1, the plain language and structure of each claim mandates that: (1) each of the references to “*the* first duration of contact” must refer to the same contact and (2) each of the references to “*the* at least one [...] first menu items” must refer to the same menu item(s). In other words, a given contact must be capable of doing either of the two functions, depending on whether its duration surpasses a threshold. Dr. Terveen’s mapping, which is the same for claims 1 and 101, fails to meet this limitation. Instead, Dr. Terveen cites to portions of Hinckley taking *different* actions in response to two *different* contacts made on two *different* menu items. This does not meet the plain language of the claims.

87. Here, Dr. Terveen maps to *two different* contacts on two different icons. EX1003, ¶¶107-108 (“tapping[] *the icon 1712* would have triggered one

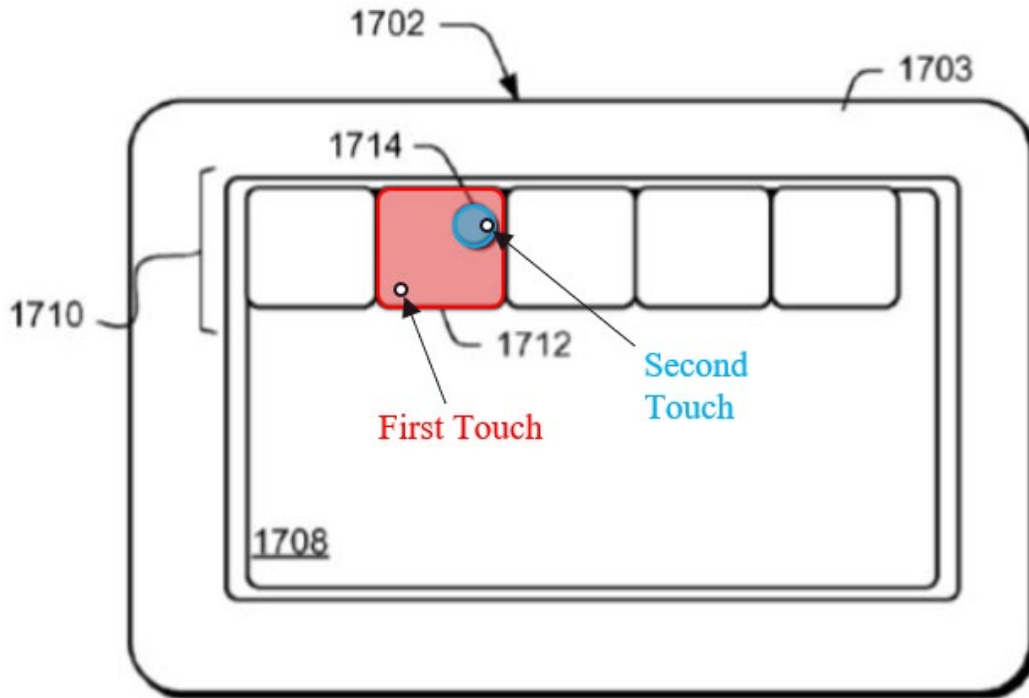
action, while a tap-and-hold gesture on the *radial menu icon 1714* portion of icon 1712 would have triggered a different action [...].” (bold and italic emphasis added)).

88. First, Dr. Terveen clearly maps to a user’s tap on the bezel menu icons (612-620 of FIG. 6; 1712 of FIG. 17 of Hinckley) as the contact for the claimed “first duration of contact.” EX1003, ¶¶105-108. In analyzing Limitation [1.e.i], Dr. Terveen quotes paragraph [0077] of Hinckley, which states “[t]he user may then tap on a desired item associated with one of the icons or slots 612, 614, 616, 618, and 620,’ which causes ‘the functionality associated with a particular icon or slot can be accessed.’” EX1003, ¶105 (citing EX1006, ¶[0077]) (emphasis added in EX1003). Dr. Terveen continues that, “[t]hrough its teaching of this tap [on icons 612, 614, 616, 618, and 620], Hinckley describes [...] *detecting, utilizing the touch interface, a first duration of contact on at least one of the one or more first menu items*, as in [1.e.i].” EX1003, ¶¶105-106 (emphasis added to “[t]hrough its teaching of this tap”). Dr. Terveen identifies the same contact—a tap on the bezel menu icon—for Hinckley’s FIG. 17, explaining “‘bezel menu 1710 includes multiple selectable icons or slots,’ and ‘[e]ach of the icons or slots is associated with a different functionality,’ [like in] FIG. 6.” EX1003, ¶¶105-107 (quoting EX1006, ¶[0106]). Thus, Dr. Terveen clearly maps “tapping the icon 1712” as being the contact associated with the claimed “first duration of contact.” EX1003, ¶¶107-108;

see also id., ¶¶105-106. Similarly, Dr. Terveen cites to the functions described in paragraphs [0077] and [0106] of Hinckley for the functionality associated with this contact. EX1003, ¶¶105-108. Accordingly, the contact identified for the claimed “first duration of contact” occurs on the bezel menu icon (e.g., 1712).

89. Second, Dr. Terveen then identifies a different (i.e., second) contact—“a tap-and-hold gesture on the radial menu icon 1714,” for triggering a different action than the action associated with the tap on the bezel icons (e.g., 1712, 612-620). *See* EX1003, ¶¶107-108. Thus, Dr. Terveen’s mapping identifies “tapping the icon” (a first contact) and then the separate action of “tapping and holding the radial menu icon portion of that same icon” (a second contact). EX1003, ¶¶108-114.

90. In Dr. Terveen’s mapping, the tap on the bezel menu icon 1712 (first menu item) would be a first contact on a first object, and the tap and hold on the radial menu icon 1714 (second menu item) would be a second contact on a second object. An example of Dr. Terveen’s mapping is illustrated in the below annotated FIG. 17 of Hinckley. *See* EX1003, ¶¶110-113 (“***tapping the icon*** would have triggered one action, while a ***tap-and-hold on the radial menu icon portion*** of that same icon would have triggered a different action” (bold and italic emphasis added)).



91. But the fact that a first contact on bezel menu icon 1712 results in a different action than a *second* contact on radial menu icon 1714 (i.e., a different contact on a different icon at a different point in time) fails to meet the claimed operation. In other words, Dr. Terveen’s mapping performs different actions based on where a contact occurs on the icon (e.g., the main portion of the icon or the radial menu icon portion of the icon). Two different inputs on two different icons (one input to icon 1712 and the other input to icon 1714) fail to meet the claims. This does not perform different actions in response to “the first duration of contact” surpassing or not surpassing a threshold, as the claim requires.

92. That Dr. Terveen interprets the radial menu icon 1714 as “a portion of that same [bezel menu] icon [1712]” is irrelevant. Regardless of whether radial

menu icon 1714 is a portion of icon 1712 or separate from icon 1712, Dr. Terveen's mapping clearly identifies two different points of contact on two *different portions* of icon 1712, and on two different portions of the screen. This does **not** perform operations "in response to the first duration of contact" being detected to surpass / not surpass a threshold, as claimed. To illustrate, "tapping the icon 1712" as identified by Dr. Terveen does not perform a different function if the user continues to maintain contact on the *same* portion of icon 1712 that was tapped. In other words, a tap on icon 1712 would not perform a different function than a tap-and-hold on the *same* portion of icon 1712. Only by selecting a *different* portion of the icon (e.g., 1714) does the combination offered by Dr. Terveen perform another function.

93. For at least this reason, the modification advanced by Dr. Terveen fails to arrive at Limitations [1.e.i]-[1.e.iii]. Dr. Terveen's mapping of claim 1 is incorporated into Limitations [101.e.i]-[101.e.iii] without further analysis or explanation. EX1003, ¶¶158-160.

b) The Institution Decision Does Not Follow the Theory Laid Out by Dr. Terveen

94. The Board states that "Petitioner contends that Hinckley discloses tapping the radial menu icon [1714] would have triggered one action different than an action triggered from tapping and holding the icon [1714]." D.I., 31 (citing Pet.

53–54, EX1003 ¶¶111–113).” But this does not follow the theory that Dr. Terveen advances.

95. Dr. Terveen states that “it would have at least been obvious to implement Ahn’s mobile communication terminal 100 such that *tapping the icon* would have triggered one action, while *a tap-and-hold on the radial menu icon portion of that same icon* would have triggered a different action.” EX1003 ¶111; see EX1003 ¶110 (“the icon 1712”). Thus, Dr. Terveen (1) identifies “tapping *the icon*” as triggering one action; (2) identifies “the icon” as being Hinckley’s “icon 1712”; and (3) distinguishes “the icon” from the “radial menu icon portion *of that same icon*.” EX1003 ¶¶110–111. Dr. Terveen clearly maps the “first operation”—which must be performed by the tapping in Petitioner’s theory—to “*performing a function associated with icon 1712*”⁸ that is “different” than opening the radial menu 1715 that results from tapping and holding the radial menu icon 1714).” EX1003, ¶112. This same mapping—tapping icon 1712—is clearly advanced throughout the

⁸ “performing a function associated with icon 1712” cannot occur from tapping on the radial menu icon 1714—which triggers a different action than “the [function] associated with tapping on the bezel menu [icon 1712].” Pet. 50–51. Thus, performing a function associated with icon 1712 must necessarily result from tapping icon 1712, not radial menu icon 1714 according to Dr. Terveen’s theory.

Dr. Terveen’s declaration. EX1003, ¶107 (stating that “tapping the icon 1712 would have triggered one action” and referencing “a tap-and-hold gesture on the radial menu icon 1714”), ¶110 (identifying the first menu item as “the icon 1712”); ¶107 (emphasizing the “*tapping*” is “on the bezel menu slot [1712].”) (emphasis in declaration) (quoting EX1006, ¶[0107]).

96. In fact, Dr. Terveen clarifies that he is not mapping to a tap of radial menu icon 1714. As shown in the table below, Dr. Terveen replaces the term “tapping” with “selecting” in the portion of Hinckley relied upon by the Board (D.I., 31-32) so as not to confuse the two.

EX1003, ¶107	EX1006, ¶[0107]
“ selecting ‘the radial menu icon [1714] triggers a default action which may . . . be <u>different</u> than the action associated with <u>tapping</u> on the bezel menu slot [1712]’” (quoting EX1006, ¶[0107])	“ tapping on the radial menu icon triggers a default action which may or may not be different than the action associated with tapping on the bezel menu slot.”

At no point does Dr. Terveen allege a tap on radial menu icon 1714 and, in fact, he goes out of his way to remove such a suggestion from the record. EX1003, ¶¶107, 114, 161.

**c) The Theory in the Institution Decision Misconstrues
Hinckley's Teachings**

97. Hinckley's description of "tapping on the radial menu icon [no identifier]" in paragraph [0107] describes an operation after the radial menu 1715 is open.⁹ EX1006, ¶[0107].

98. Hinckley's paragraph [0107] explains that "the radial menu 1715 can be accessed in any suitable way," and then provides three illustrative methods of access: hovering a pen, pulling down, and tap-and-hold. EX1006, ¶[0107]. After describing how the radial menu 1715 can be opened, Hinckley states that users can tap the icons within the radial menu to trigger specific actions. *Id.* A POSITA would have understood this passage to disclose that tapping on the radial menu icon occurs after radial menu 1715 has been accessed.

99. Stated another way, a POSITA would have appreciated that because Hinckley's icon 1712 accepted tap inputs, and because radial menu icon 1714 is small—smaller than the user's depicted finger—and surrounded by icon 1712, accepting taps on radial menu icon 1714 before radial menu 1715 is deployed would

⁹ Aside from opening radial menu 1715, Hinckley does not state that any action/function/command can occur by manipulating radial menu icon 1714 prior to accessing (*i.e.*, opening) radial menu 1715, whereas additional functionality is available after the radial menu is accessed.

have led to accidentally presses of icon 1712 when trying to press radial menu icon 1714.

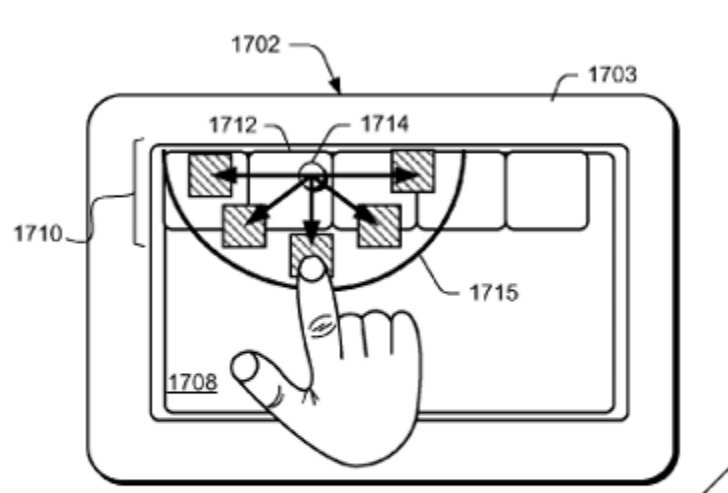


Fig. 17

EX1006, FIG. 17 (depicting icon 1714 as smaller than the depicted finger of a user). Thus, a POSITA would have understood Hinckley's disclosure of a press-and-hold input on radial menu icon 1714 to deploy radial menu 1715, and disclosure of a tap input on the radial menu icon only after accessing radial menu 1715, to logically mean that the different taps are only accepted on radial menu icon 1714 under different circumstances.

100. In contrast, a POSITA would understand that assigning functions to both a tap on icon 1712 and a tap on radial menu icon 1714, when the radial menu 1715 is not deployed, would lead to a relatively high error rate due to the small size and close proximity of the two objects. A POSITA, therefore, would have

understood from Hinckley's disclosure that a tap on radial menu icon 1714 would have no effect before the radial menu 1715 is deployed, and a touch-and-hold on the radial menu icon 1714 would have no effect once the radial menu 1715 has been deployed. This is further supported by Hinckley's description of tapping on its menu icons (e.g., 1712) to access their assigned functionalities (EX1006, ¶[0077]), but describes three *different* gestures that are *not tapping* (e.g., hovering, sliding, holding) on icon 1714 before radial menu 1715 is deployed (*Id.*, ¶[0107]). Further, this tapping distinction would not be an issue when radial menu 1715 is opened because the underlying icon 1712 is covered by the radial menu 1715 and would not be selectable (meaning the potential for error is no longer present when the radial menu is deployed). EX2005 80:8-89:22. Thus, a POSITA would understand the "tapping on the radial menu icon" in Hinckley's paragraph [0107] refers to a tap after radial menu 1715 is opened.

101. This operation involves two separate user contacts: first a tap-and-hold on icon 1714 to open the menu, and then a tap to select the cross-hatched radial menu icon. As such, this interaction does not satisfy the claim requirement for a single continuous contact. *See* Section VII.A.1. Therefore, Hinckley's teachings do not disclose the claimed gesture.

B. Dr. Terveen's Motivations for Modifying Ahn with Hinckley are Incorrect

102. Dr. Terveen provides two modifications based on Hinckley. Dr. Terveen's First Modification is that it would have been obvious "to implement Ahn's mobile communication terminal 100 such that Ahn's menus would include the multiple selectable icons or slots described by Hinkley and that at least one of the selectable icons or slots would be associated with a radial menu icon as described by Hinkley." EX1003, ¶¶110, 111, 114; *see id.*, at ¶63 ("result of this Ahn-Chaudhri-Hinkley combination is a mobile device that utilizes various gestures on a touchscreen and techniques for discerning between them to implement, among other user interface objects, bezel menus and radial menu"). Dr. Terveen's Second Modification is that it would have been obvious "to implement Ahn's mobile communication terminal 100 such that tapping the icon would have triggered one action, while tapping and holding the radial menu icon portion of that same icon would have triggered a different action." EX1003, ¶110.

103. Dr. Terveen provides two reasons to support its modifications: (1) "techniques described in Hinkley with respect to its bezel menus would have been equally beneficial in the context of Ahn's side menus" (EX1003, ¶64) and (2) "the implementation and operation of the bezel menus and radial menus[] would have been a simple substitution of one known element for another to obtain predictable results" (EX1003, ¶67). However, a POSITA would have understood that benefits

of radial menus in one application are not generally transferable to other applications and the substitution of a rectangular context menu (like Ahn's) with a radial menu (like Hinckley's) would have resulted in diminished functionality. EX2009, 4 (“Just *replacing a rectangular context menu with a pie menu* doesn't magically make everything better, though. In fact, it *introduces a new set of problems* that need to be solved.”); EX2012, 4 (“Applications for pie menus are rare”).

1. The ‘Techniques’ Identified by Dr. Terveen are Not Related to Hinckley’s Radial Menu

104. Although Dr. Terveen vaguely contends “the techniques described in Hinkley with respect to its bezel menus would have been equally beneficial” to Ahn, he identifies only two such techniques: gestural modes and bezel gestures. Neither of these techniques relate to the opening of the radial menu that Dr. Terveen's combination relies upon.

a) Gestural Modes

105. Dr. Terveen identifies Hinckley's “different gestural modes” (i.e., novice and expert modes) and asserts that implementing Hinckley's different user interaction modes in Ahn's device would have created a more intuitive, efficient, and pleasing experience for a user. EX1003, ¶64. However, Dr. Terveen's modification does not actually implement Hinckley's novice or expert mode. EX1003, ¶87-115. And doing so would have prevented the modification from meeting the claims at least because the combination of adding Hinckley's novice or

expert mode would not cause the menu to progressively follow a user's finger as described in my Section VII.D.1.b).

106. Hinckley's expert mode "allow[s] the user to more quickly perform actions on the user interface without waiting for the visual guides to appear" and Hinckley's novice mode provides the user visual guidance so that the user can become more familiar with the interface. EX1003, ¶64; EX2005, 96:10-97:13, 98:14-22. That is, the novice mode provides visual guidance showing the user how to perform the gestures so the user can memorize the gesture and associated functionality. Once the user is familiar, they may perform the desired gestures without receiving any visual guidance—resulting in the more "intuitive, efficient, and pleasing experience for a user." (EX1003, ¶64).

107. First, the novice and expert modes of displaying the bezel menu are irrelevant to the opening and operation of the radial menu. In fact, Hinckley only describes the gestural modes for a bezel menu that does not include a radial menu. *Compare* EX1006, ¶¶[0070]-[0080] (describing novice and expert modes for bezel menu without an associated radial menu). Thus, any benefits associated with the different gestural modes of a bezel menu would not result from Dr. Terveen's modification to include a radial menu (Dr. Terveen's First Modification) or operate the radial menu to perform different functions based on different gestures (Dr. Terveen's Second Modification).

108. The gestural modes associated with the radial menu are also unrelated to Dr. Terveen's modifications in Limitations [1.e.i-iii]. EX1003, ¶100-115. In the novice mode, the user would tap-and-hold icon 1714 to access radial menu 1715 and then remove their finger to assess the menu. EX1006, ¶¶[0080], [0110]. And, in the expert mode, the user would tap-and-hold-and-drag icon 1714 in the direction associated with the target function before the radial menu opens. EX1006, ¶[0110]. This functionality (and any associated benefits) would not result from a tap of icon 1712 as asserted by Dr. Terveen (Dr. Terveen's Second Modification) or a tap of icon 1714 as asserted by the Board. The benefit of the novice/expert mode is unrelated to the modification to tap on an icon to perform one action or tap-and-hold to perform another action. Both novice and expert use a tap-and-hold and don't rely on the tap of an icon.

109. Further, as explained in my Section VII.B.2 below, a POSITA would have understood that the benefits associated with the gestural modes of Hinckley's radial menu would have been lost when implemented to replace Ahn's menu.

110. Accordingly, the benefits of having novice and expert modes are not pertinent to explaining why a POSITA would have been motivated "to implement Ahn's mobile communication terminal 100 such that tapping the icon would have triggered one action, while tapping and holding the radial menu icon portion of that

same icon would have triggered a different action” as Dr. Terveen suggests in his paragraph 110.

b) Bezel Gestures Modes

111. Next, Dr. Terveen identifies Hinckley’s “bezel gestures” and asserts that implementing Hinkley’s specific gesture recognition techniques would have yielded these same general benefits in gesture distinction described by Hinkley. EX1003, ¶¶65-66 (citing EX1006, ¶¶[0001], [0040], [0046]). Dr. Terveen points to “Hinckley’s specific gesture recognition techniques” of distinguishing “gestures that start from the bezel and enter onto the screen” (i.e., bezel gestures) from other similar on-screen gestures. EX1003, ¶65 (citing EX1006, ¶[0046]). For example, Hinckley can distinguish a right swipe gesture starting on the bezel and entering into the screen from a right swipe gesture that starts on the screen so that the bezel gesture “do not interfere with” traditional gestures that occur only on the screen. EX1006, ¶[0046]. Hinckley’s paragraph [0046] also explains that the bezel gesture module 105 can still detect bezel gestures in embodiments where a touch on the bezel cannot be detected. Hinckley states that “different techniques can be utilized to sense a user’s input relative to the bezel 103” and that certain attributes are usable to differentiate the touch input [of a bezel gesture] from other touch inputs [from non-bezel gestures] recognized by the gesture modules 104, 105.” Such distinction between bezel and non-bezel gestures are what “yields the general benefit that gestures that start from

the bezel and enter onto the screen are, in general, distinguishable from other ostensibly similar gestures.” EX1006, ¶[0046].

112. However, these bezel gestures are distinct from the design of Hinckley’s menus and are not utilized at all with respect to Hinckley’s radial menu. EX2005, 72:7-12. Accordingly, the benefits associated with the functionality of bezel gestures or distinguishing between bezel and non-bezel gestures do not explain why a POSITA would have been motivated to “include the multiple selectable icons or slots described by Hinkley and that at least one of the selectable icons or slots would be associated with a radial menu icon as described by Hinkley” or “implement Ahn’s mobile communication terminal 100 such that tapping the icon would have triggered one action, while tapping and holding the radial menu icon portion of that same icon would have triggered a different action” (EX1003, ¶110).

2. A POSITA Would Have Understood that Hickley’s Radial Menu is Not a Simple Substitution for Ahn’s Linear Menus.

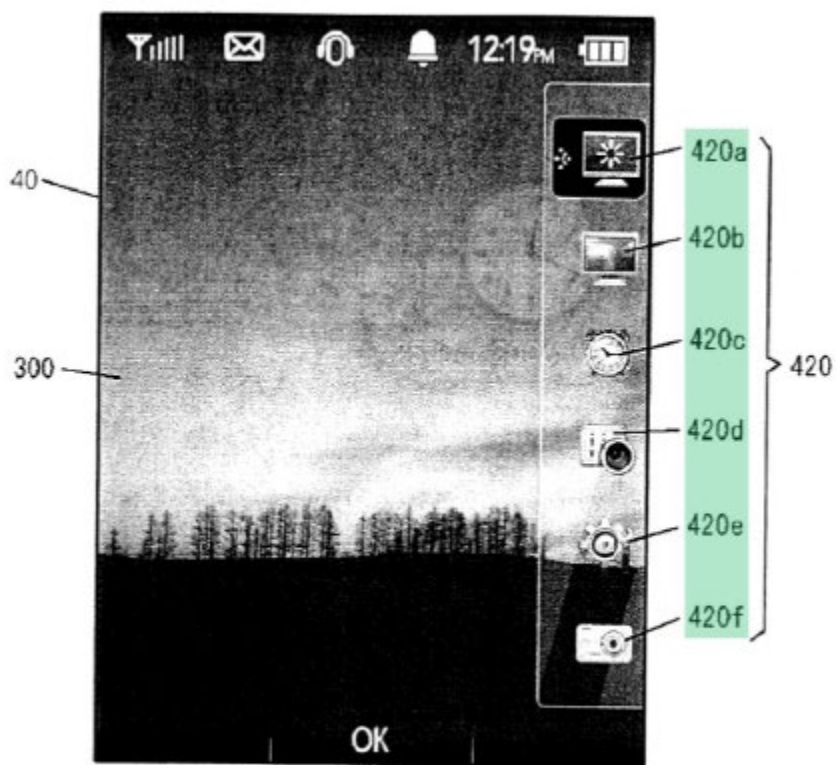
113. A POSITA would have understood that simply substituting Hinckley’s bezel and radial menu for Ahn’s menu would not yield predictable results as this substitution would yield further problems that are not addressed by the art or Dr. Terveen. EX2009, 4 (“Just replacing a rectangular context menu with a pie menu doesn’t magically make everything better, though. In fact, it ***introduces a new set of problems*** that need to be solved.”). In fact, a POSITA would have understood that

implementing a radial menu into Ahn's menus would have eliminated or severely diminished the functionality of Ahn.

a) A POSITA would not use Radial menus to perform the functionality described for Ahn's "first" menu

114. Claim 1 requires two functions to occur "in response to the detection of the first gesture:" (1) "displaying ... *a first menu* as sliding in and including one or more first menu items" and (2) "blurring at least a portion of the graphical user interface." See Section V.A ([1.d]). Dr. Terveen maps Ahn's right menu 1300/420 as corresponding the claimed first menu as this is the only menu in Ahn that 'blurs' the clock widget. See EX1003, ¶¶83-84, 100; EX1004, FIGs. 3A-3E, 6B, 7B-7C, 9A-9B, and 17 (depicted top, bottom, and left menus being opened without 'blurring' the clock). Claim 1 further requires "detecting, utilizing the touch interface, a first duration of contact on at least one of the one or more first menu items [*of the first menu*]" and performing two operations based on the duration of the contact in Limitations [1.e.ii-iii]. See Sections V.A and VII.A.1. Dr. Terveen identifies sub-menus 420a-420f of Ahn's right menu 1300/420 as the first menu items of the first menu.

FIG. 4B

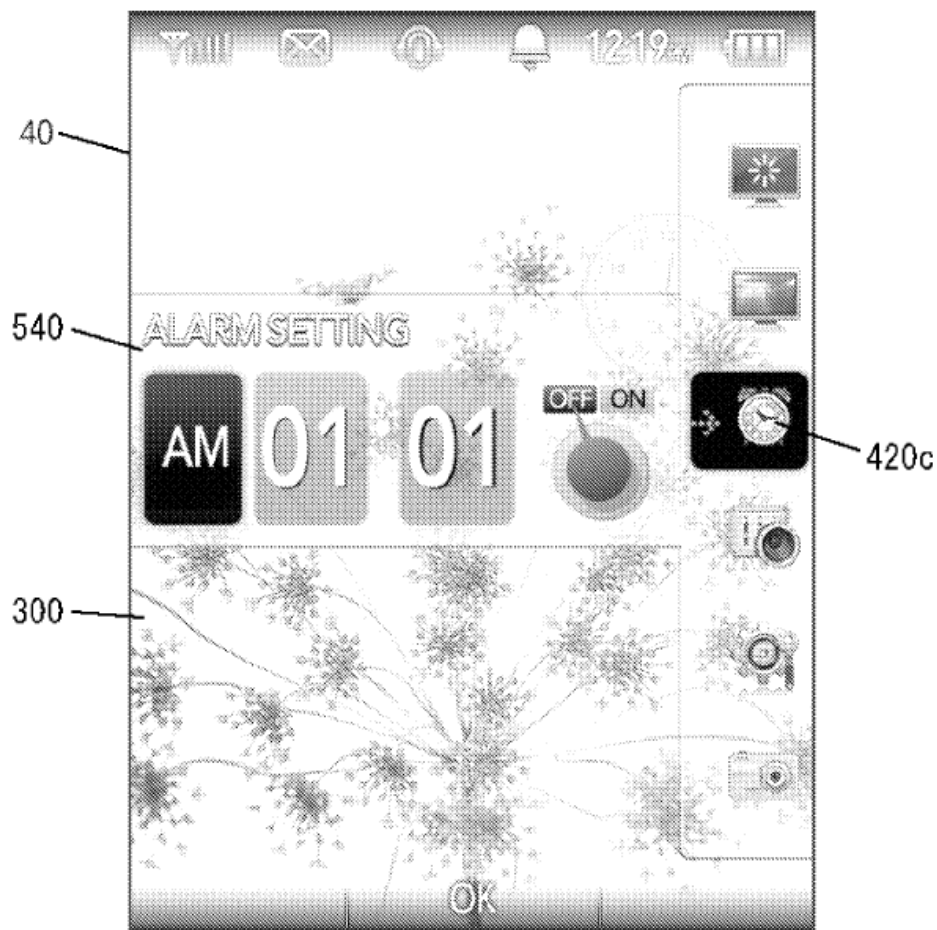


EX1004, FIG. 4B (highlighted)

EX1003, ¶100.

115. However, a POSITA would have understood that the functionality described for Ahn's sub-menus 420a-420f cannot be implemented by a radial menu. For example, Ahn's FIG. 5D illustrates that "when the user selects the alarm setting menu 420c, an alarm setting window 540 for setting an alarm is displayed in a transparent window" and the user can set an alarm using the alarm setting window 540. EX1004, ¶[0057].

FIG. 5D



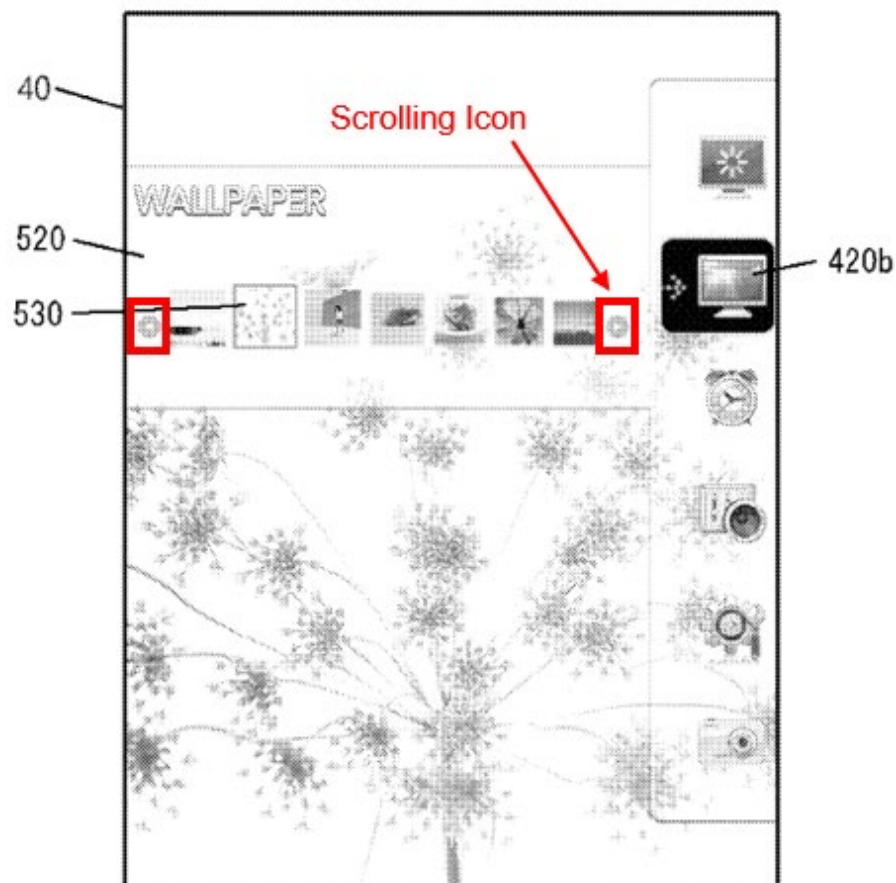
EX2017, 56¹⁰; EX1004, FIG. 5D. The alarm setting window two binary toggles (AM/PM and ON/OFF) and two numerical inputs (01-12 for hours of the day and

¹⁰ For improved clarity, the figure is from the counterpart European application (EP1962480) that is substantially identical to Ahn and claims priority to the same applications as Ahn. See EX2017, 78/79 (claiming priority to KR 2007-018175, KR 2007-033308, KR 2007-043087); EX1004, 1 (same).

01-59 for minutes). This functionality represented in this window cannot be reasonably implemented into a radial menu.

116. Likewise, Ahn's standby screen setting menu 420b opens up a standby screen setting window 520 for selecting a wallpaper that could not be reasonably implemented as a radial menu. EX1004, ¶[0054]. Referring to FIG. 5B (annotated and reproduced below), standby screen setting window 520 includes 7 icons that are displayed between two scrolling icons.

FIG. 5B



EX2017, 54 (annotated); *see* EX1004, FIG. 5B . This scrolling functionality is not something that would have been reasonably implemented into a radial menu. EX2005, 92:14-94:6 (explaining scrolling works for linear menus but not for radial menus). Further, a POSITA would have understood the number of options—at least 8 options —would be too numerous to be displayed in a *half or quarter* radial menu, as described by Hinkley, without introducing the usability problems. EX1006, ¶[0114], FIG. 17; EX2009, 4. Further, the slider bar functionality in brightness setting window 510 that is associated with sub-menu 420a (FIG. 5A) would not be easily implemented as a radial menu. EX1006, ¶[0114], FIG. 17.

117. A POSITA would have understood that radial menus would only be utilized in specific context and are not simply interchangeable with other types of menus. In fact, “replacing a rectangular context menu with a [radial] menu ... introduces a new set of problems that need to be solved.” EX2009, 4. Indeed, all of the sub-menus 420a-420f that Ahn explicitly describes (*see e.g.*, EX1004, FIGs. 5A-5D) would not have been implemented as including radial menus at least because they could not have been or would have resulted in severely diminished functionality

and user experience. Accordingly, the implementation and operation of the bezel menus and radial menus would not have been merely a simple substitution.¹¹

b) Radial menus would have limited the functionality of Ahn’s “second” menu

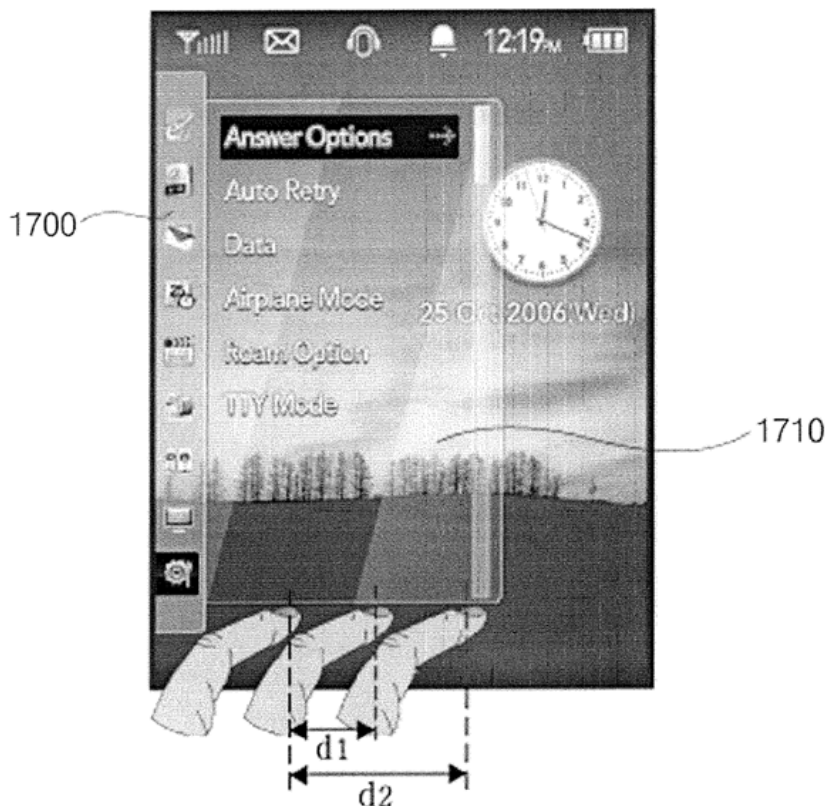
118. Recognizing that radial menus would not naturally implement the functionality described for Ahn’s right side menu 1300/420, Dr. Terveen attempts to draw a parallel between Ahn’s left side menu 1700 and sub-menu 1710 and Hinkley’s bezel menu 1710 and radial menu 1715. EX1003, ¶109. However, as explained later in my Section VII.D.1.a) below, Ahn’s left side menus 1700 and 1710 cannot meet the requirements for the claimed first menu and, therefore, are irrelevant to Dr. Terveen’s explanation of the operations of “the one or more first menu items” of the first menu. *See* EX1004, ¶[0111]-[0112] (menu list is displayed *after* drag distance of the touch operation exceeds a distance).

119. Nevertheless, a POSITA would have understood that using a radial menu “as ... an alternative” (EX1003, ¶109) to Ahn’s second menu list 1710 was

¹¹ Although Dr. Terveen’s modification is to “include the multiple selectable icons or slots [and associated radial menu icon] described by Hinkley” (EX1003, ¶110), a POSITA would have understood that implementation of any typical radial menu is not suitable for all interfaces and, particularly, that it would have limited the functionality available to cellular devices like those contemplated by Ahn.

inapplicable and would have decreased the functionality of Ahn's device. Second menu list 1710, depicted in Ahn's FIG. 17, is a result of the selection of a setting menu. EX1004, ¶[0082], FIGs. 9A-9B (depicting identical sub-menu as a result of the cursor selecting the settings item); FIG. 17 (illustrating setting icon of 1700 as being highlighted). The second menu list 1710 displays six options with one of those options indicating the presence of a further sub-menu. EX1003, ¶109 (arrow icon identifies the presence of a sub-menu).

FIG. 17



EX1004, FIG. 17.

120. A POSITA would not have implemented menu 1710 as a radial menu. First, the radial menu described in Hinckley lacks the functionality to cohesively

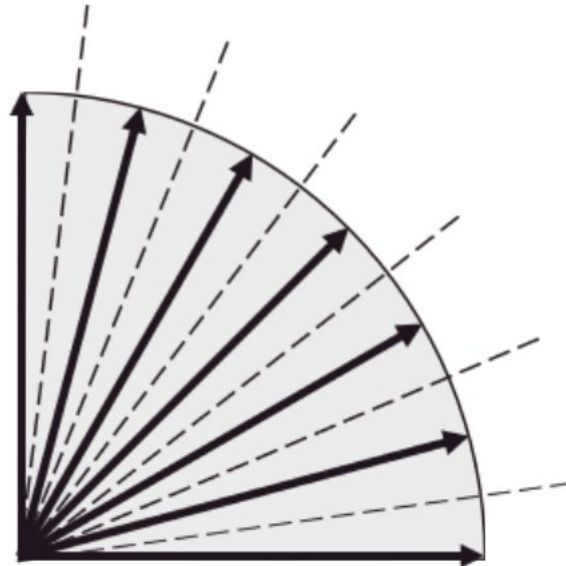
display further sub-menus as indicated by the arrow next to the “Answer Options” items. EX1003, ¶109. Further, a POSITA would have understood that the additional functions associated with the “Answer Options” sub-menu would not have been able to be inserted into the implemented radial menu as the six existing options were already too numerous for a partial radial menu like Hinckley’s. EX2010, 2; EX2009, 4. This cramped spacing would have been exacerbated given that Ahn’s text options would take up greater space and could not have easily been conveyed as pictorial icons, like Hinckley’s. EX2014¹², 2 (using text descriptions in radial menus is “a mess”); EX1006, ¶[0074], FIGs. 6, 17; EX1004, FIG. 17 (listing options like “Answer options,” “Auto Retry,” Roam Option,” and “TTY Mode” that would were not easily representable by icons). This implementation would be even more untenable given the position of the setting icon in menu 1700—depicted in the

¹² Although EX2014 was published in 2012, it accurately represents the opinions of a POSITA in the 2011 time frame. The space constraints of radial menus and was well known at this time and did not change between August 2011 and July 2012. *See* EX1006, ¶[0109]; EX2005, 93:1-18 (Dr. Terveen noting geographic constraints of radial menus). Further, the notion that text was messier than icons was also known. *See* EX1006, FIG. 17 (using icons for radial menu); EX1004, FIG. 17 (icons smaller than text).

bottom right corner (i.e., the end of the bezel)—which would necessitate a quarter-circle menu. EX1004, FIG. 17; EX2005, 90:19-92:11. Even Hinckley warned against such a configuration. EX1006, ¶[0109] (stating the semi-circle radial menu includes “five options” but that “[r]adial menus associated with slots or icons that appear at the ends of the bezel menu 1710 may have *fewer options due to spacing constraints*).

121. A POSITA would have known the six-plus options of Ahn’s second menu list 1710 would not have been suitably implemented into a radial menu without losing the benefits of Hinckley’s radial menu. For example, as explained in my Section VI.D, the use of six icons in a quarter circle would not allow a user to accurately select the icons based on muscle memory, as the angle of the swipe would be too fine to consistently reproduce. When using a typical 8 icon full radial menu, there is a 45 degree angle between each different swipe gesture associated with the icons; whereas the six icon quarter-radial menu would gestures separated by a mere 15 degrees. A user could not reproduce such refined gestures without the help of visual guidance, and, thus, Hickley’s expert mode would not be usable in this configuration. See EX2009, 4 (when the wedges get too small, you lose the benefit of being able to select one without looking at it.). To illustrate, if 15 degrees was the center of a radial icon, and there is only a 15 degree threshold for each icon (as shown in the example below), then the tolerance is only plus or minus 7.5 degrees.

If a user were off by more than 7.5 degrees (e.g., a swipe of 22.6 degrees), then it would trigger the incorrect icon.



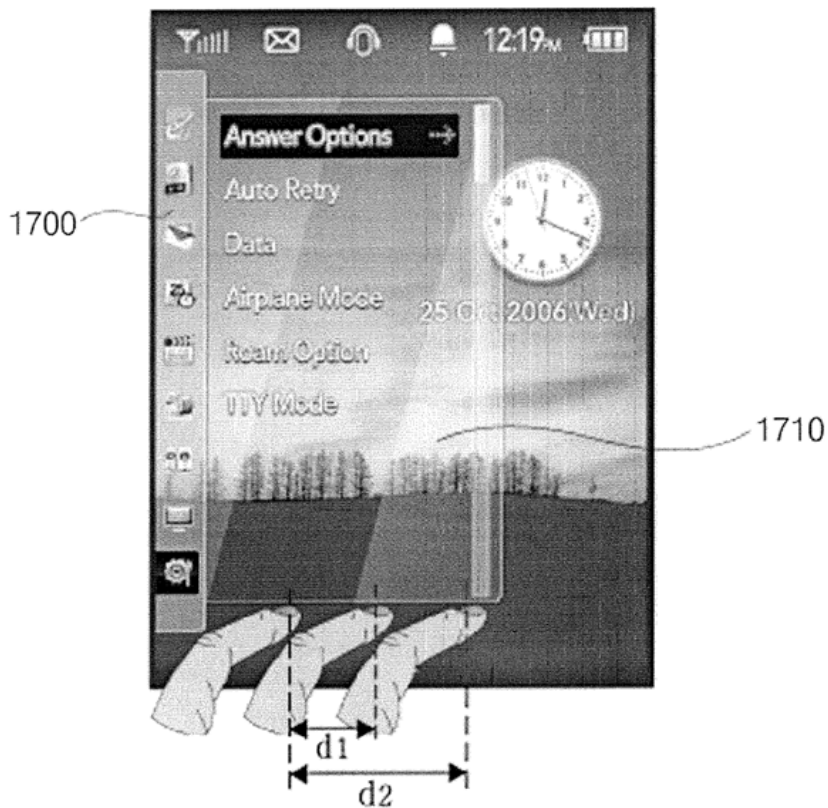
Further, to fit all of these options within the menu the size of the menu would be required to be very large, having long skinny ‘slices’ which increased the distance to select the icon, the time required to select the icon, and decreased the accuracy of the selection. *See* EX2009, 2 (increasing distance increases time)

122. Ahn does not describe the functionality of its remaining options in menu 1700 (shown as menu 820 in FIG. 9A). However, Dr. Terveen asserts that selecting some such items (e.g., “Contacts” and “Messages”) would have launched an application associated with that functionality. EX1003, ¶157. A POSITA would understand that a radial menu would be inappropriate for such operations. Unlike Hinckley’s icon 1712—which is associated with five separate functions/commands (EX1006, FIG. 17)—Ahn’s application-launching icons would be associated with

only a single command: launching an application. Because of its association with *multiple* commands, Hinckley's radial menu allows a user to change the function that occurs when selecting its icon 1712 so that one icon can perform each of these multiple commands. EX1006, ¶[0109]. Yet, in Dr. Terveen's theory, one icon (e.g., "Messages") would be associated with a *single* command (e.g., launching the Messages application). Thus, as explained in Section VII.B.2.c), where there is only one function (i.e., opening the application) that is associated with an icon—as opposed to the plurality of functions associated with Hinckley's icon 1712—a POSITA would not have included a radial menu as there are no auxiliary functionalities to be accessed.

123. Further, including an icon "associated with a radial menu icon as described by Hinkley" (EX1003, ¶110) would have prevented a user from accessing the functionality of the menu when one of the sub-menus is open. As depicted in Ahn's FIG. 17, the icons of the parent menu 1700 remain displayed when the sub-menu 1710 is shown indicating that they remain selectable when the user is viewing the sub-menu.

FIG. 17



EX1004, FIG. 17.

124. Thus, a user can switch between sub-menus by selecting another of the eight icons displayed in menu 1700. However, when including the slots and associated radial menus of Hinckley, the radial menu would prevent the user from interacting with the majority of the icons in menu 1700. EX2005, 77:2-8, 80:8-22; EX1016, FIG. 17 (depicting 80% of the icons being obscured). Because the radial menu (e.g., 1715 of Hinckley's FIG. 17) overlies the icons, the user cannot select the covered icons without first performing some action to remove the radial menu; whereas Ahn's interface allows a user to quickly select between options while

viewing a sub-menu. EX2005, 77:2-8 (“not possible to access the underlying icons”). Dr. Terveen’s modification limits this functionality.

125. A POSITA would have understood that radial menus would only be utilized in specific context and are not simply interchangeable with other types of menus. As explained above, Dr. Terveen’s modification would significantly diminish Ahn’s menu interface and would not achieve any of the purported motivations that Dr. Terveen identifies. Indeed, all of the sub-menus 1700 that Ahn explicitly describes and the menus 820 identified by Dr. Terveen would not have been implemented as including radial menus. Accordingly, the implementation and operation of the bezel menus and radial menus would not have been merely a simple substitution.

c) Dr. Terveen’s analogizing of Ahn and Hinckley’s menu is inappropriate

126. As explained above, the radial menus would not be suitable for any of Ahn’s menus. This incompatibility stems from Dr. Terveen’s false analogy between Ahn’s menu and Hinckley’s bezel menu. EX1003, ¶109. A POSITA would have understood that all menus are not compatible or interchangeable and that menus with similar appearances can vary drastically according to their underlying functionality. Specifically, a POSITA would understand Ahn’s list menus were distinct from Hickley’s bezel menu despite sharing some similarities (e.g., both had icons arranged linearly).

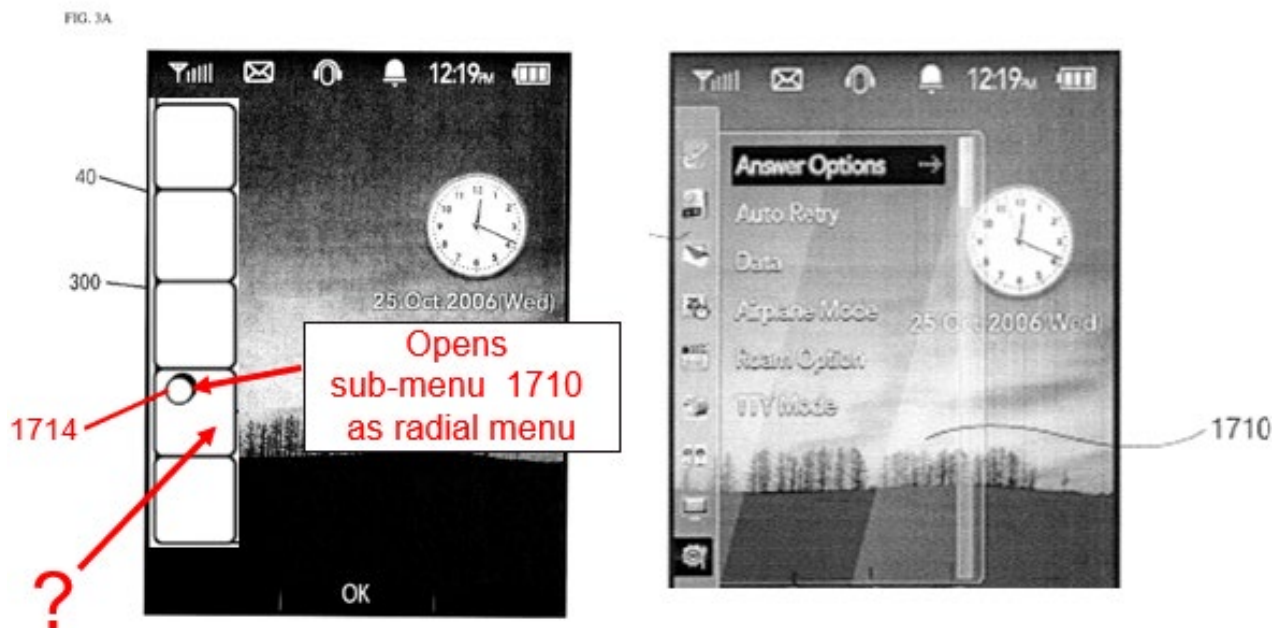
127. Ahn's menus operate as a cascading type menu, with the parent options functioning to reveal sub-options or sub-menus. In other words, selection of Ahn's menus only performed a single function—revealing other menus. EX1004, FIGs. 5A-5D, 7B-7C, 9A-9C. To illustrate, each menu option 420a-420f of Ahn's right menu 1300/420 reveals a window when it is selected (EX1004, ¶¶[0052]-[0057]), the only described menu option (e.g. a setting menu) of Ahn's left menu 1700/820 reveals a sub-menu 870 when it is selected (EX1004, ¶¶[0079]-[0082]), and Dr. Terveen's alleged operation of the menu options of Ahn's left menu 1700/820 launch an application when it is selected (EX1003, ¶157). Thus, Ahn's menu items perform a single occurrence function (e.g., to open or reveal other options).

128. On the other hand, Hinckley's bezel menu includes a single icon 1712 that can perform several different continuously interactive functions, such as “such as, for example, paint functionality, pen functionality, note functionality, object creation, object editing, and the like.” EX1006, ¶[0106]. Selecting icon 1712 accesses only a portion of the programmed functionality (e.g., triggering one of five functions). EX1006, ¶[0077] (“The user may then tap on a desired item associated with one of the icons or slots 612, 614, 616, 618, and 620. Through this gesture, the functionality associated with a particular icon or slot can be accessed. For example, tapping on a particular icon or slot may cause an object to be created on the canvas associated with display device 608.”). There is, therefore, a need to be able to access

the remaining untriggered functionalities, which can be done by selecting a separate radial menu icon 1714 that reveals a radial menu 1715. The radial menu 1715 enables a user to change which of the functions the icon 1712 is associated with. EX1006, ¶[0109]; EX2005, 82:2-84:4. For example, a user can select icon 1712 to use a paint function on display 1708. EX1006, ¶[0106]. Once finished with the paint function/command, the user can then open radial menu 1715 and select a different functionality (e.g., pen, eraser, editor) to assign to icon 1712 that the user can subsequently use to interact with the display 1708. Thus, Hinckley's bezel menu has a single icon 1712 that can be selected to perform multiple interactive functions. EX1006, ¶¶[0106], [0109].

129. Dr. Terveen ignores these differences in functionality between Ahn and Hinckley's menus and contends they can simply be substituted for one another; but a POSITA would understand that this is not the case. EX2009, 4-5 EX2012, 4 ("Applications for pie menus are rare"). Ahn's single icon is assigned a single function (e.g., opening a sub-menu) and, thus, there are no untriggered functionality that needs to be accessed. Petitioner's modification is to add a radial menu icon to the icons of the menu 1700. Thus, the radial menu icon would be used to open and select this sub-menu with a tap-and-hold-and-swipe. It is unclear how Dr. Terveen's combination would be implemented. For example, as shown in the annotation of Dr. Terveen's combination below, he contends tapping-and-holding on radial menu

icon 1714 opens the radial menu 1715 (EX1003, ¶115) that would correspond to Ahn's second menu list 1710 (EX1003, ¶109) and that tapping icon 1712 would perform a function associated with icon 1712 that is "different" than opening the radial menu (EX1003, ¶112).



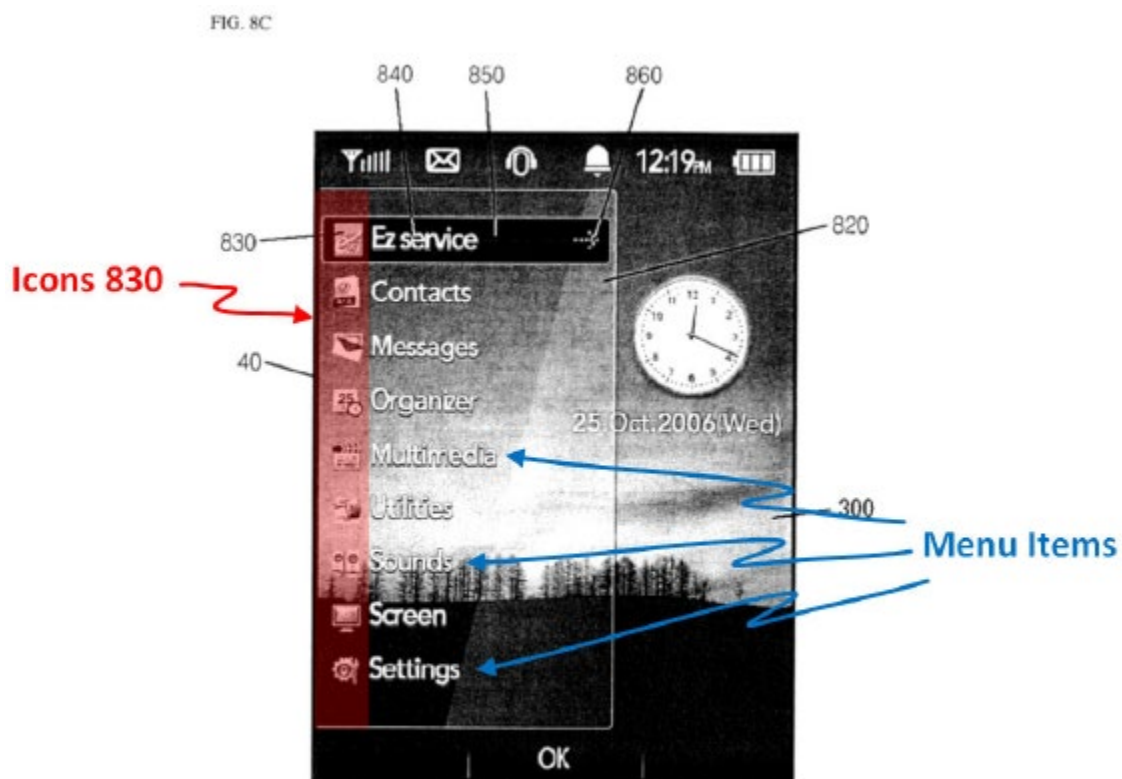
130. But the only function associated with “the icon” is opening the sub-menu, which is now deployed as a radial menu in Dr. Terveen’s modification. Thus, there is no ‘different’ action to be performed by including “the multiple selectable icons or slots described by Hinkley and that at least one of the selectable icons or slots would be associated with a radial menu icon,” as Dr. Terveen asserts (EX1003, ¶¶110, 111, 114), and the proposed combination would not meet Limitations [1.e.i-iii].

131. Further, a POSITA would understand that the functionality associated with Hinckley's icons is different from the sub-menu's associated with Ahn's menus. Ahn's menu icons (e.g., 420a-f, 830) are not-particularly-interactive, operating to open up sub-menus or windows (e.g., 510, 520, 540, 870). EX1004, FIGs. 5A-5D, 9A-9B, ¶¶[0051]-[0057], [0081]-[0085]. This is a common functionality associated with a standby screen, where icons are usually used to open menus or launch applications. EX1004, ¶[0005] (Ahn's menus are designed for use when "a specific button is selected on a standby screen"). On the other hand, Hinckley's icons are used to implement interactive functions (e.g., interact with the display by painting, drawing with a pen, creating or editing interactive objects) and would occur in an application that is separate from a standby/background screen. For example, Hinckley describes paint, pen, note, and object editing functions for icon 1712 that would be used in a drawing or image processing application, not a standby screen. EX1006, ¶[0106]. Although radial menus were applicable for heavily interactive applications, such as Hinckley's drawing or image processing application, a POSITA understood that radial menus would not have been used in less-interactive interfaces like Ahn's standby screen. EX2012, 4. Dr. Terveen merely assumes that substituting Hinckley's radial menu for Ahn's contextual menu would provide the same benefits, but a POSITA understood that is not the case. citing EX2009, 4.

C. Dr. Terveen's Mapping of Claim 101 is Internally Inconsistent

132. Claim 101 recites “displaying [...] a first menu as sliding in and including one or more first menu items including a particular application icon” ([101d]) and “displaying, utilizing the display, a second menu including one or more second menu items” ([101g]). EX1001, 95:57-65, 96:12-20.

133. In mapping to the claimed first menu and the particular application icon, Dr. Terveen identifies Ahn's basic setting menu list 820 and icon 830, as illustrated in the annotated FIG. 8C of Ahn (inset below). EX1003, ¶¶155-156.

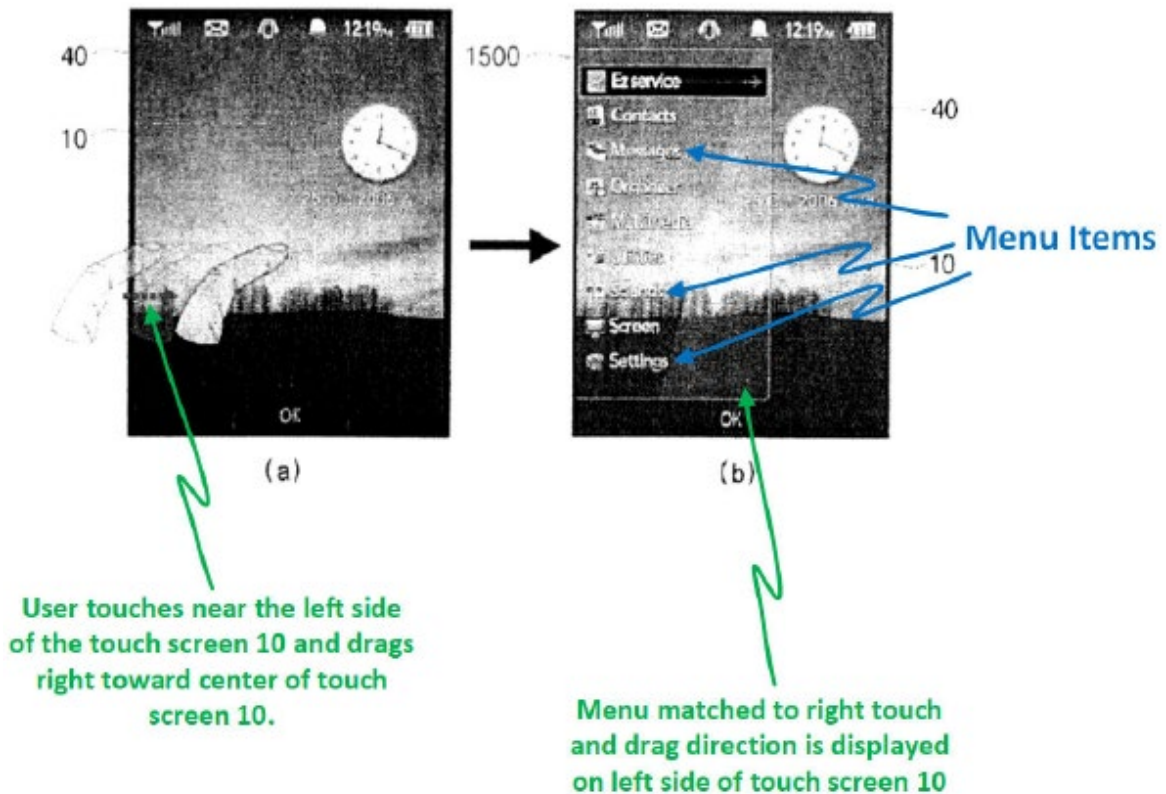


EX1004, FIG. 8C

Dr. Terveen explains that some of these menus (e.g., “Contacts” and “Messages”) “would have launched an application associated with that functionality.” EX1003, ¶157. Thus, Dr. Terveen identifies the Contacts or the Messages icon as corresponding to the claimed particular application icon.

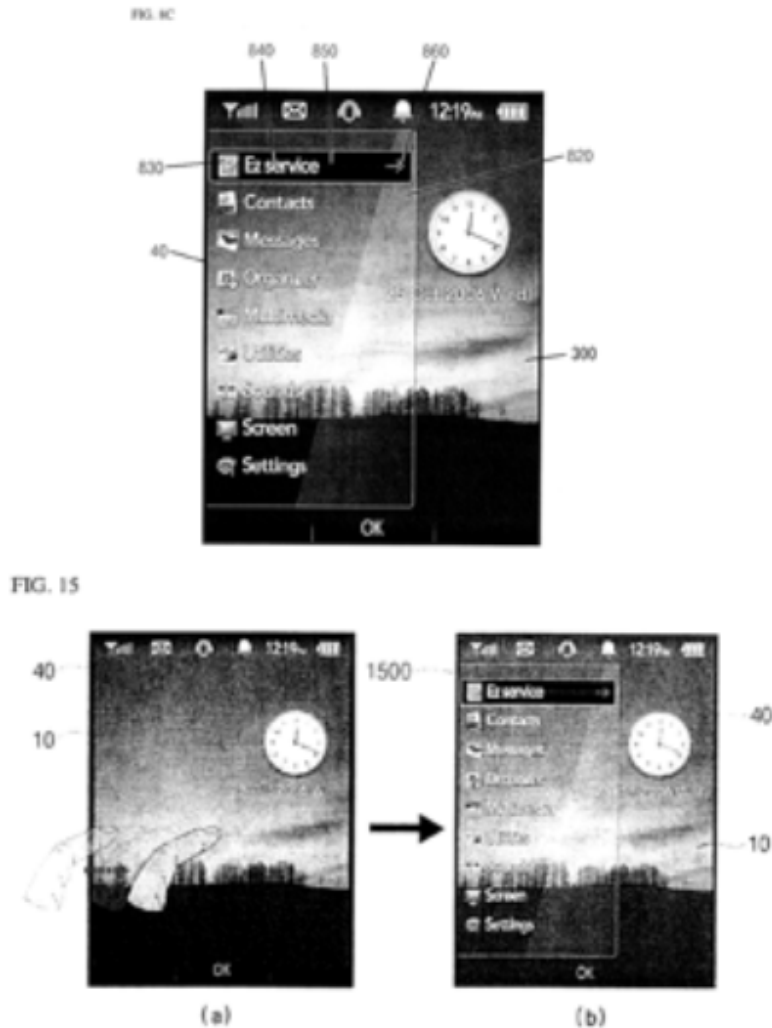
134. In mapping to the claimed second menu, Dr. Terveen incorporates his discussion of Limitation [1.g]. EX1003, ¶165 (“Element [101.g] [is obvious] for at least the same reasons described above with respect to Element [1.g].”). For Limitation [1.g], Dr. Terveen identifies Ahn’s menu list 1500 as corresponding to the claimed second menu, as illustrated in the annotated FIG. 15 of Ahn (inset below). EX1003, ¶¶121-122.

FIG. 15



EX1004, FIGS. 15a and 15b

135. As explained in Section VI.A, basic setting menu list 820 corresponds to the left menu in Ahn's directional key embodiment, and menu list 1500 corresponds to the same left menu in Ahn's touch and drag embodiment. EX1004, ¶¶[0073]-[0076], [0103]. Thus, the first and second menus (menu list 820 and 1500) identified by Dr. Terveen are the *same* menus. Compare EX1004, FIG. 8C (Menu 820 listing Ez service, Contacts, Messages, Organizer, Multimedia, Utilities, Sounds, Screen, and Settings options); with FIG. 15 (Menu 1500 listing same options).



Thus, Dr. Terveen's mapping to the same menu as corresponding to both the first and second menu is erroneous.

136. Even if Dr. Terveen interprets the menus as different because they are from different embodiments of Ahn, this interpretation would only introduce further deficiencies into Dr. Terveen's mapping. Both the claimed first and second menus are displayed in response to the detection of first and second gestures ([101d], [101g]), respectively, that are detected utilizing the *touch interface* ([101c], [101f]).

Thus, even if menus 820 and 1500 are determined to be different because they are from different embodiments, the menu 820 associated with the directional key operation would fail to be displayed in response to the detection of a gesture via the touch interface. *Supra*, ; EX1004, ¶¶[0073]-[0076] (“If the user selects the leftward direction key 38 of four direction keys provided in the key input device 30 [...]”).

1. The Institution Decision Does Not Follow the Theory Laid Out by Dr. Terveen and Contradicts Dr. Terveen’s Other Theories

137. As explained above, Dr. Terveen maps to the same menu—Ahn’s left menu—for the first and second menus. The Board contends otherwise because pages 59-60 of “the Petition further identifies Paragraphs 102 through 104 of Ahn as describing the claimed second menu.” D.I., 40. The Board notes “Paragraph 102 in particular discusses only Figures 13a and 13b” and, therefore, Petitioner must contend that “the menu list shown in Figure 13b [right menu], as well as the menu list shown in Figure 14b [bottom menu], teach[es] the claimed second menu recited in limitation [1.g]” that is incorporated into Limitation [101g]. *Id.*

138. The Board’s theory is based on a typographic error found in Dr. Terveen’s declaration and the Petition that misattributes the quoted text from Ahn’s paragraph 99 to paragraph 102. EX2005, 89:12-18 (confirming this citation is a typographic error); *Compare* EX1003, ¶122 with EX1004, ¶¶[0099], [0102].

122. “As the touch and drag operation is detected, the controller 50 controls to display an item list matched to the specific direction in the display 40 (S1120).”

EX1004, [0099]. “The item list comprises, for example, a menu list or a file list.”

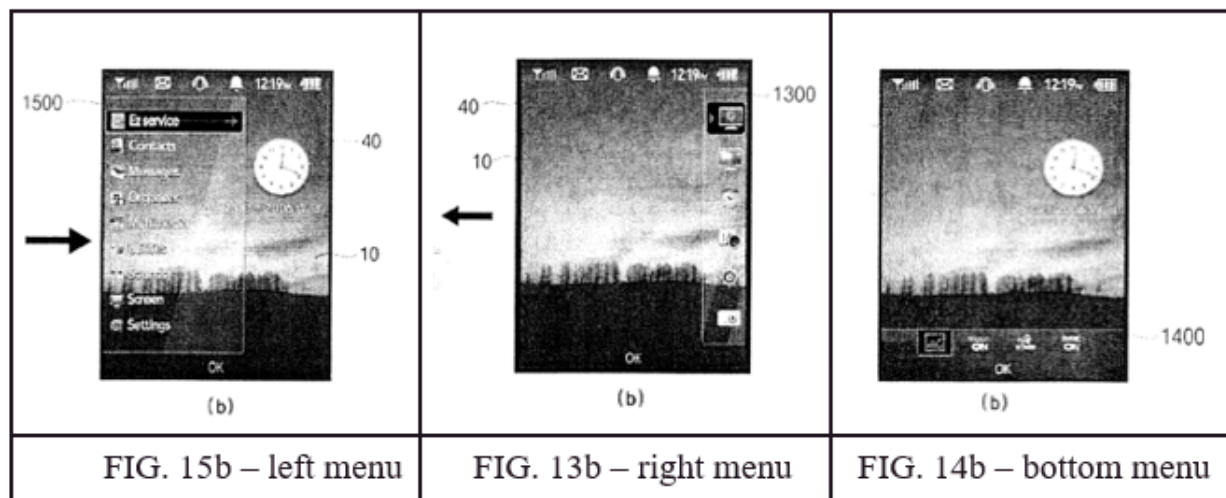
EX1004, [0102]. In one example illustrated in FIGS. 15a and 15b, “if the user

[0099] As the touch and drag operation is detected, the controller 50 controls to display an item list matched to the specific direction in the display 40 (S1120). The item list comprises, for example, a menu list or a file list. The memory 20 stores matching information between the specific direction and the item list. The memory 20 can store at least one matching information. For example, a menu list A comprising menu a, menu b, menu c, and menu d may be matched to a right direction. Further, for example, a menu list B comprising menu e, menu f, menu g, menu h, and menu i may be matched to a left direction.

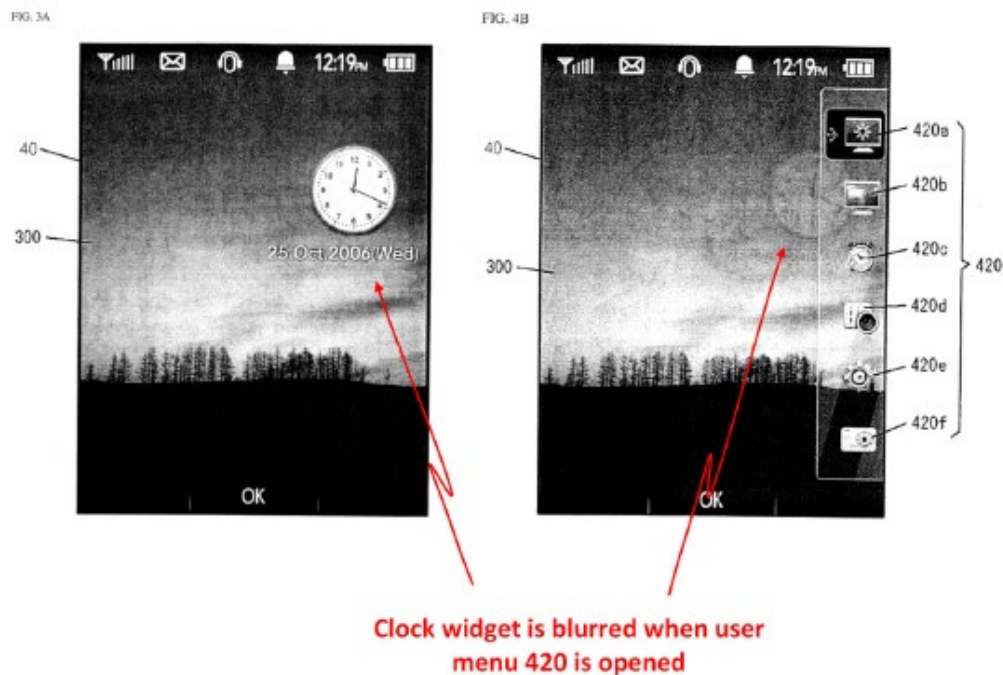
139. While Dr. Terveen also cites to paragraphs 103-104, these citations address a menu “matched to a right direction” (i.e., a right swipe) and, in response, displaying a menu “at the left side of the display (i.e., Ahn’s left menu), which corresponds to “menu list 1500” shown in “FIG. 15(b).” EX1003, ¶122.

122. “As the touch and drag operation is detected, the controller 50 controls to display an item list matched to the specific direction in the display 40 (S1120).” EX1004, [0099]. “The item list comprises, for example, a menu list or a file list.” EX1004, [0102]. In one example illustrated in FIGS. 15a and 15b, “if the user matched to a right direction is displayed in the display 40,” EX1004, [0103]. “A position in which the item list is displayed in the display 40 can be preset.” EX1004, [0104]. “For example, referring to FIGS. 13 to 15, . . . a menu list matched to a right direction is set to be displayed at the left side of the display 40 (FIG. 15(b)).” EX1004, [0104]. Thus, a POSITA would have understood and found obvious that

Thus, neither the Petition nor Dr. Terveen maps to the menu list shown in Figure 13b (right menu) or the menu list shown in Figure 14b (bottom menu) for any of the Limitations associated with the second menu. See EX1003, ¶¶121-130 (repeatedly identifying Ahn’s *left* menus 820/1500 as the second menu).



140. Further, the Board’s theory creates inconsistencies in Dr. Terveen’s other mappings. For example, if Dr. Terveen contends that Ahn’s left menu list 820/1500 teaches the claimed first menu recited in limitation [1.d], then Dr. Terveen’s theory fails as the associated gestures (“first gesture”) would not blur the clock widget that Dr. Terveen maps to the graphical user interface, illustrated below:



EX1004, FIGS. 3A and 4B

EX1003, ¶93.

D. Dr. Terveen’s Mapping to the Blurring Limitation of Claim 1 is Deficient

141. Claim 1 recites “blurring at least a portion of the graphical user interface such that a magnitude of the blurring of the at least portion of the graphical user interface progressively increases as a function of an increase in a magnitude of

the first gesture being detected.” EX1001, 95:57-65. Dr. Terveen advances two theories when mapping to Limitation [1.d.ii], each of which fails to meet the Limitation.

1. Ahn Theory

142. Dr. Terveen asserts that Ahn’s menu moves with the user’s finger as the user drags their finger across the display. *See* EX1003, ¶92. Dr. Terveen must show this assertion to be true in order for its mapping of Ahn to meet element [1.d.ii]. Unless Ahn’s menu follows the user’s finger across the display, the overlap of Ahn’s menu onto the standby screen does not “increase[the blurring] as a function of an increase in magnitude of the first gesture[,]” as required by element [1.d.ii].

a) Ahn’s Menus are Not Progressive

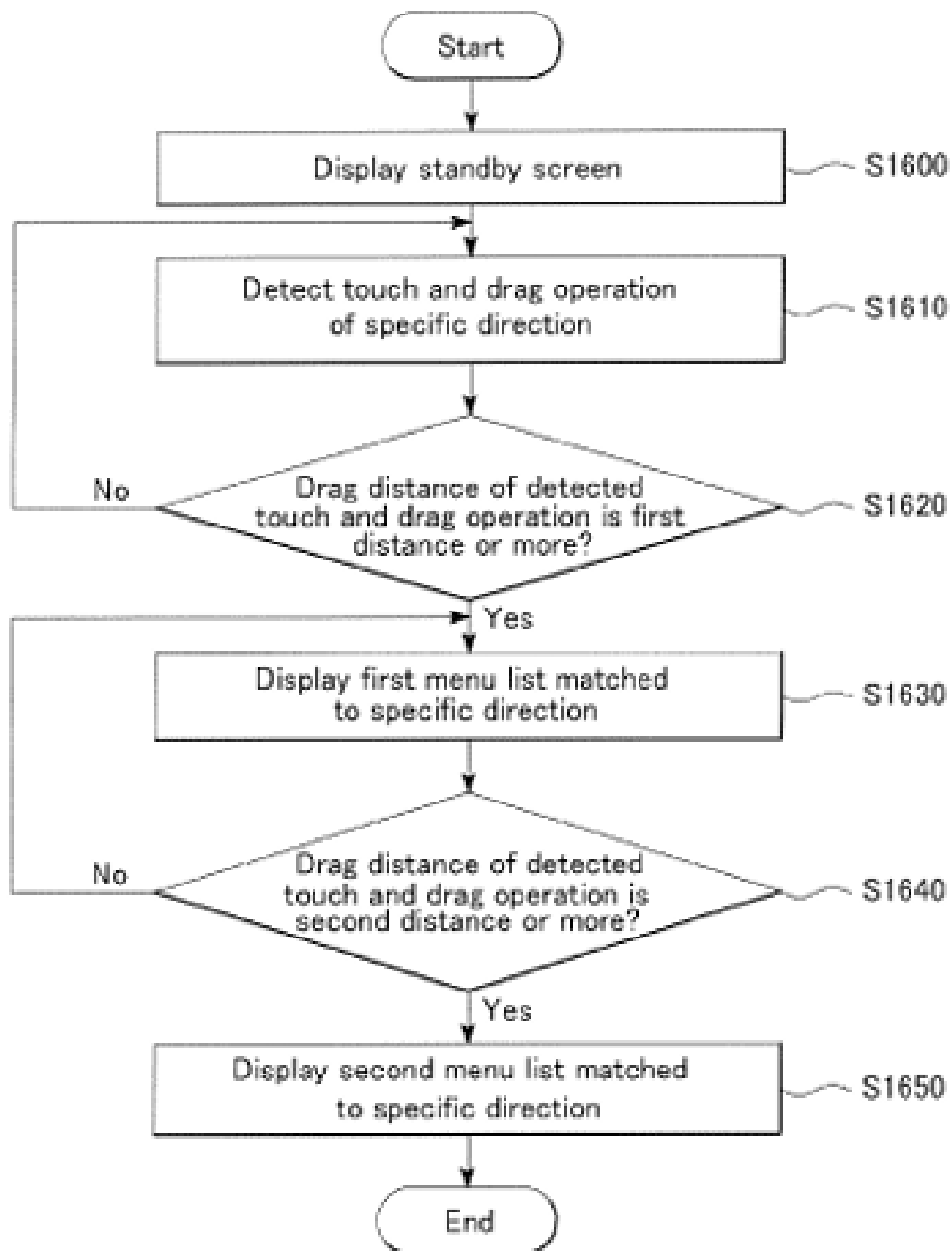
143. To establish Ahn’s menu progressively slides with the user’s finger, Dr. Terveen relies on one sentence in Ahn’s paragraph [0118]: “as the user drags the graphic shadow 1800, the menu list 1500 can be displayed along the dragged graphic 1800.” EX1003, ¶86. The crux of Dr. Terveen’s theory is that “drags the graphic shadow” means that the showed is depicted as *moving with* the user’s finger when it performs a touch and drag operation. But this is inconsistent with Ahn’s use of the term “drag.”

144. Ahn unambiguously uses “drag” to describe the input—*not* any responsive movement of graphic 1800. In Ahn, “drag” “indicates movement of an

external object on the touch screen 10.” EX1004, ¶[0098]. In other words, “drag” means an external object (e.g., finger, stylus, etc.) is acting on the screen or part of the screen, not that a graphic is moving. For example, where Ahn states that “if the user *touches a specific position* of the touch screen 10 and *drags the touch* in a left direction a menu list 1300 . . . is displayed[,]” (EX1004, ¶[0102]) the user is dragging a finger across the touch screen. Similarly, Ahn states that “if the user *drags the touch screen 10* by a distance longer than or equal to the distance d1 and shorter than the distance d2, the first menu list 1700 is displayed.” EX1004, ¶[0116]. Of course, the user is not moving the touch screen 10 (which is depicted in Ahn’s FIG. 1) but is merely performing a *drag gesture that starts on the touch screen*, in response to which the first menu list is displayed. See EX1004, ¶[0111] (describing touch operation as “dragged”). Likewise, when “the user drags the graphic shadow 1800” (EX1004, ¶[0118]), the user is merely performing a drag gesture that *starts* on the graphic shadow, in response to which the menu list 1500 is displayed.

145. Ahn’s description of its other operations confirm this interpretation. As explained in Section VI.A, a POSITA would understand Ahn’s menu operations are binary and would open after a corresponding input is received. Ahn’s “Touch and Drag” embodiments are described as binary—not progressive—and only displays its menu *after* determining that a drag distance exceeds a first distance as opposed to displaying the menu during the gesture.

FIG. 16

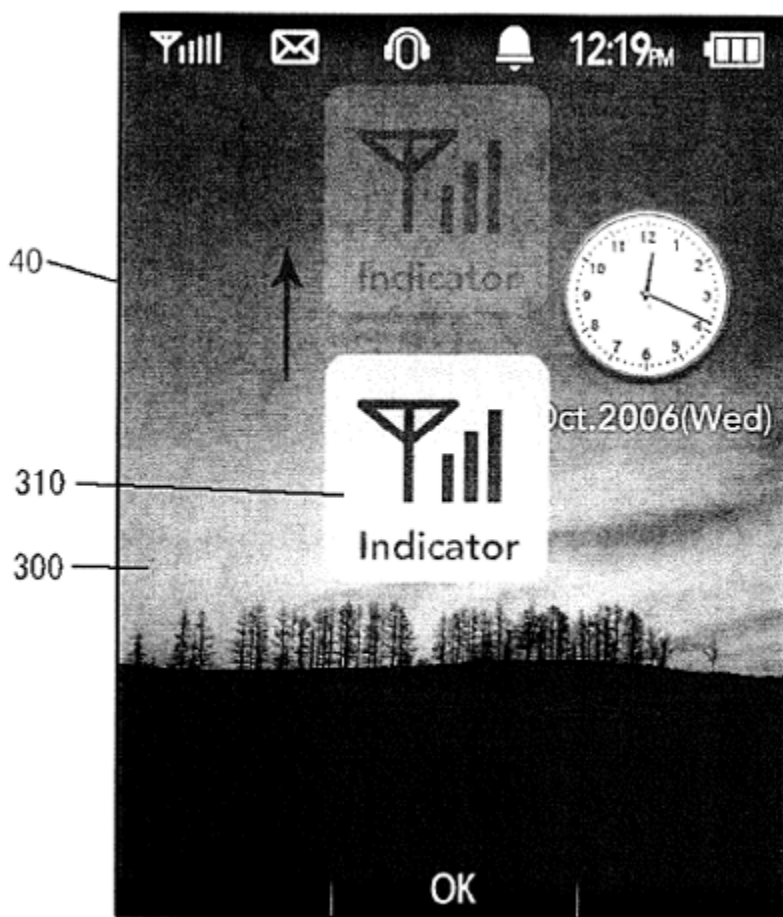


EX1004, FIG. 16 (step S1630 only occurs after S1620 is met); EX1004, ¶[0109]
 (“controller 50 determines whether a drag distance of the detected touch and drag operation is a first distance or more (S1620). *If a drag distance of the detected touch*

and drag operation is a first distance or more, the controller 50 controls to display a first menu list.”). Similarly, as explained in my Section VI.A, the operation of FIGs. 11 and 13-15 discloses a touch and drag operation where the menu is opened after the gesture is completed. *See e.g.*, EX1004, ¶¶[0102] (“if the user touches ... and drags the touch in a left direction”), [0116] (“if the touch and drag operation of the user *is completed*”)

146. Ahn’s disclosure to add “an identifier” for “notifying a start position of the touch and drag operation” (EX1004, ¶[0117]) does not change the described binary menu opening operation to a progressive one. In other words, paragraph [0117] of Ahn merely describes adding a start position identifier to the binary “Touch and Drag” menu embodiments already described elsewhere in Ahn’s disclosure and does not describe differences in the process through which the menus are displayed. *See D.I.*, 26-27 (finding Figures 18a-18b and Figures 15a-15b the same embodiments). In fact, Ahn illustrates the user’s finger moving along the display while the graphic 1800 remains stationary. EX1004, FIG. 18a. If Ahn wanted to display the graphic 1800 as moving with the finger, it knew how to display such a movement. In fact, Ahn’s Figures 3B, 4A, 6A, 7A, 8B depict an object as moving in response to an input. *See e.g.*, EX1004, ¶[0037] (“an icon 310 ... disappears in the standby screen 300 while moving in an upward direction after being displayed around the center of the standby screen 300”).

FIG. 3B



EX1004, FIG. 3B. Ahn does not describe graphic shadow as “moving” or depict the shadow as moving. There is nothing to establish that Ahn’s paragraph [0118] intends to use the term “drag” inconsistently with the rest of the specification or operate the menu in a manner contrary to the described touch and drag operation.

b) Hinckley confirms Ahn’s non-progressive operation of Dr. Terveen’s combination

147. When implementing the first menu, Dr. Terveen relies on the alleged “parallel between the first menu list 1700 and second menu list 1710 described by Ahn’s FIG. 17 and the bezel menu 1710 and radial menu 1715 described by

Hinkley's FIG. 17." EX1003, ¶109. Like Ahn, Hinkley's menus are non-progressive when sliding in¹³ and are displayed only after an input is recognized. For example, when utilizing the "different gestural modes ... to access functionality associated with the bezel menu" (EX1003, ¶64 (quoting EX1006, ¶¶[0076])), Hinkley's menu would not follow the user's finger as it slides in from the edge. In Hinkley's expert mode, the menu would never be displayed. EX1006, ¶¶[0079], [0110]); EX1003, ¶64 (without visual guides); EX2005, 96:10-97:13, 98:14-99:10. And in Hinkley's novice mode, the menu is only displayed "when the timer expires" and the user's finger is still in the region occupied by the bezel menu. EX1006, ¶¶[0080]¹⁴. Thus, like Ahn, Hinkley's menu is not displayed until after the corresponding input (e.g., distance of slide or timer) is detected.

¹³ Dr. Terveen's citation to Hinkley's menu sliding "off of the screen" is irrelevant to show why a POSITA would have displayed a menu "as sliding in," which Hinkley teaches as non-progressive to achieve the advantages of its 'gestural modes.'

¹⁴ Hinkley's statement that the menu "tracks with the user's finger" is distinct from its description of an object "following the user's finger" indicating that tracking (*i.e.*, monitoring) is distinct from moving the object with (*i.e.*, following) a finger. EX1006, ¶¶[0080])

148. Accordingly, implementing Hinkley's bezel menu in the "different gestural modes" necessitates the menu to operate contrary to that alleged in Dr. Terveen's modification. EX1003, ¶64 ("implementing Hinkley's different user interaction modes"). To be able to implement these different gestural modes, Hinkley's bezel menu must not initially move with the user's finger as the system must first determine (via a "timer") whether the user intends to operate in the novice or exert mode. But this functionality cannot coexist with Dr. Terveen's reliance of Ahn's menu sliding out as a user drag's graphical shadow 1800 for Limitation [1.d]. Although the Board found that that "dragging a finger inwards from an edge of a display screen to 'slide in' a menu onto the display screen was within the level of ordinary skill in the art," this does not mean that a POSITA would have implemented such functionality. As explained above, a POSITA would not have arrived at an operation to have Ahn's menu follow a user's finger, as this is contrary to the operations in both Ahn and Hinckley and would eliminate the functionality of the gestural modes that Dr. Terveen relies on for its motivation

149. Further, 'sliding' the menu (e.g., "off of the screen" as in Hinckley) as the Board alleges does not indicate that this is performed as a function of an increase in a magnitude of the first gesture. For example, this sliding could be shown as an animation, such that a menu slides out after the touch and drag operation is completed. This functionality would not meet the claimed language as the menu

would not “progressively increase[] as a function of an increase in a magnitude of the first gesture being detected.” For example, in Hinckley’s novice mode, the bezel menu would slide in after the timer expires and would not slide as a function of an increase in a magnitude of a gesture. EX1006, ¶[0080]. Nor does this menu follow the user’s finger when it slides out. EX1006, ¶[0080] (menu “tracks” (i.e., detects) user’s finger which is distinct from an object “following the user’s finger”); EX1003, ¶88 (Dr. Terveen acknowledging an object, not a menu, follows finger “in other contexts”).

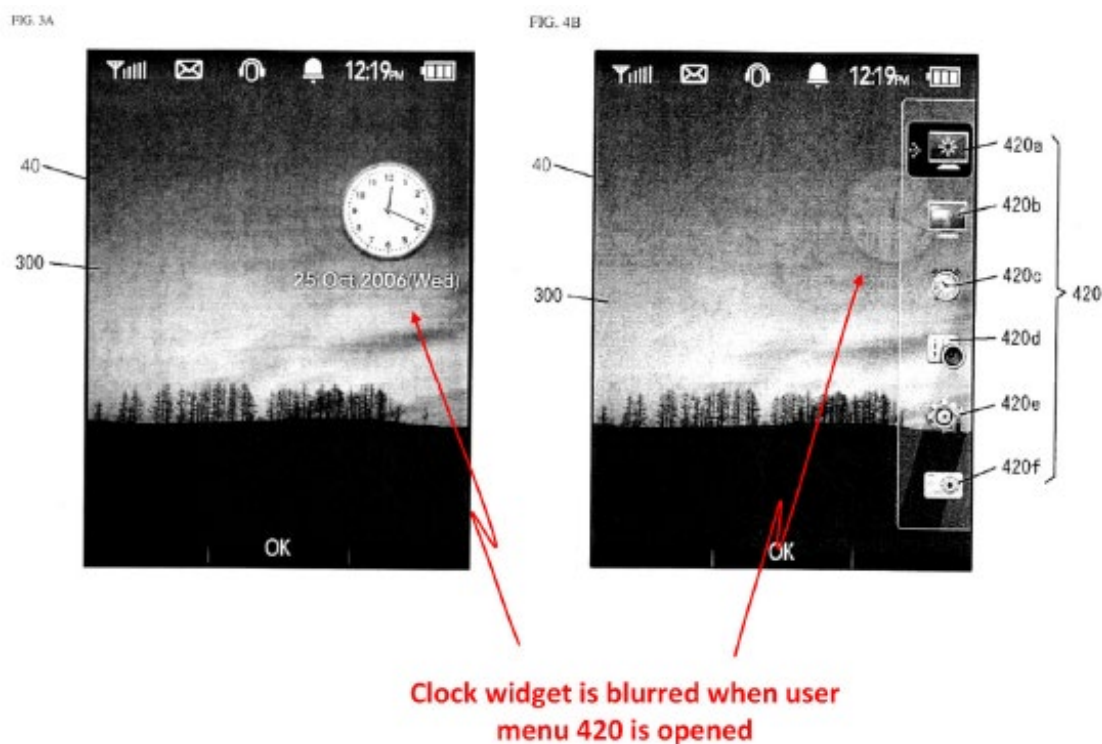
2. *Ahn-Chaudhri Theory*

150. Dr. Terveen’s modification of Ahn with Chaudhri is premised on the combination providing a user with an indication of or feedback on the progress of Ahn’s gesture. EX1003, ¶57 (“Ahn-Chaudhri combination is a mobile device that changes an optical intensity of one or more user interface objects as a function of progress towards satisfaction of a user’s gesture input”). Specifically, Dr. Terveen asserts that a POSITA viewing Chaudhri-842 would have found it obvious to vary the optical intensity of the clock widget as a visual indication of a user’s progress in completing Ahn’s “touch and drag” gesture. EX1003, ¶98; *see id.*, ¶94 (“obvious to apply Chaudhri-842’s user interface state transition techniques— particularly the use of transitioning optical intensity to provide visual feedback regarding the progress

of a transition between user interface states as a result of detecting a user's input of a predefined gesture"). But this is logically incorrect.

a) Modification does not show progress

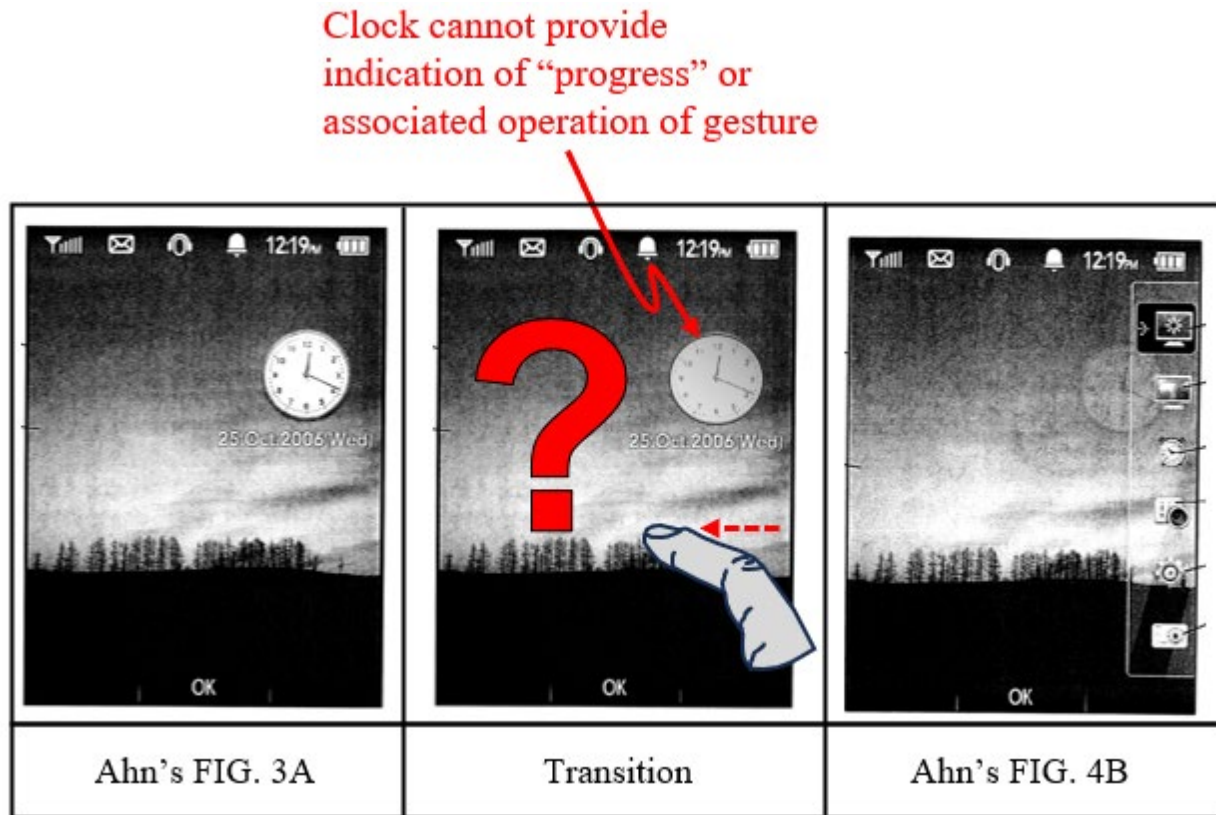
151. Dr. Terveen's alleged "blurring" of Ahn's clock widget during Ahn's "touch and drag operation" would not provide the user with any indication of progress of opening the corresponding menu list. In Dr. Terveen's modification, the "clock widget is blurred when the user menu 420 is opened," as illustrated by Ahn's FIGs. 3A and 4B.



EX1004, FIGS. 3A and 4B

EX1003, ¶¶93-94.

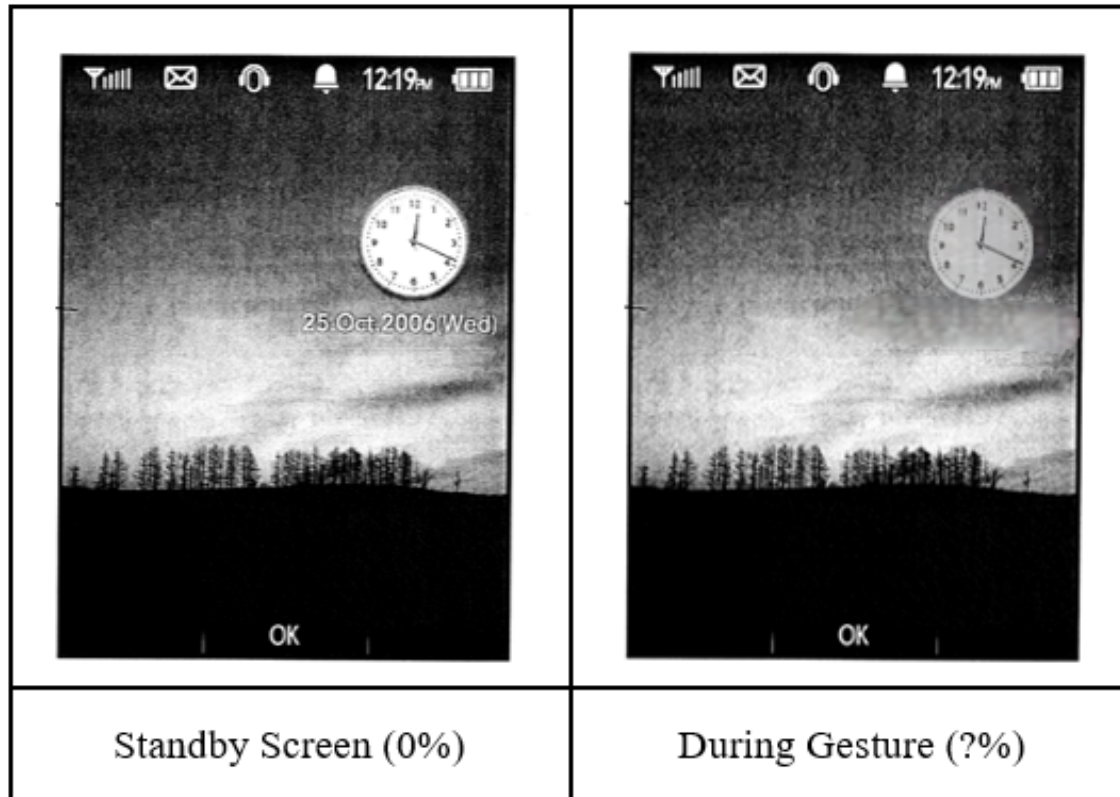
152. Dr. Terveen’s alleged modification of Ahn is illustrated by the below progression which transitions from a Standby Screen (corresponding to Ahn’s FIG. 3A) to a Menu Screen (corresponding to Ahn’s FIG. 4B) in response to a touch and drag gesture as described in Ahn. *see* EX1003, ¶98 (“obvious to vary ... the optical intensity of the standby screen or at least portions thereof (e.g., the blurring of the clock widget, as shown in FIGS. 4B, 10, and 13b) as a visual indication of a user’s progress in completing Ahn’s “touch and drag” gesture).



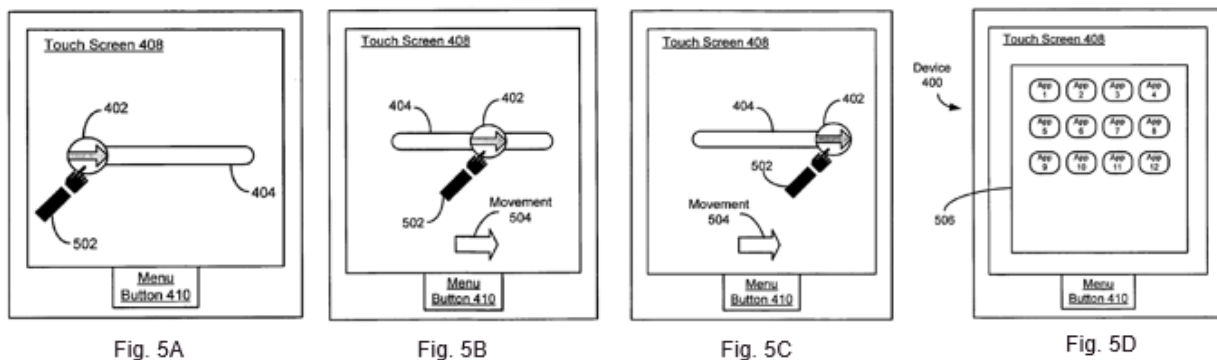
153. As is apparent in this sequence of figures (and Ahn’s FIG. 13B), the clock widget remains visible throughout the transition. EX1004, FIG. 4B (depicting clock widget). This optical transition of the clock widget, from left to right in the

images above, does not suggest to a user *when* or *if* the menu 420 will appear as in Ahn's FIG. 4B. Nor is this end-state (display of the menu) otherwise related to the alteration of the clock widget in any manner that is perceivable by a user. For example, there is no frame of reference to discern any progress, relative to its required distance, of the gesture based on the changing appearance of the clock widget.

154. As such, the change in optical intensity *during* the touch and drag operation would not provide any indication of (1) how much of the gesture is completed, or (2) how much of the gesture is remaining. *Id*; *contra* EX1005, ¶¶[0088]-[0090] (“If the user input condition includes a predefined gesture, then the indication of progress of the gesture may be defined in terms of how much of the gesture is completed and ***how much of the gesture is remaining***”); EX1003, ¶97 (quoting same). In other words, a user cannot tell from the alleged “blurring” how much more “blurring” or swiping is needed before the menu will appear. This can be illustrated in the below figure:



Unless a user can accurately determine the amount of progress for Ahn's touch and drag gesture based on the variation in the clock widget in the above image, then it cannot indicate progress. This is inconsistent with Chaudhri's ability to easily convey the progress of the gesture to a user. *See* EX1005, FIGs. 5A-5D.



155. A POSITA would understand that this change in appearance merely indicates that a gesture is happening, but the user *already* knows that much because the user *is performing* that gesture. As in Ahn alone, the user would only know the gesture is complete once the menu list 1300/420 appears, ***upon completion of the gesture***. Accordingly, a POSITA would have understood that Dr. Terveen's purported modification would not provide any indication of a user's progress in completing Ahn's "touch and drag" gesture because, unlike Chaudhri-842, Dr. Terveen's proposed UI shows no indication of progress toward a clearly visible end goal. Thus, Dr. Terveen's purported 'progress' motivation would not have led a POSITA to make the asserted modification of Ahn.

156. Further, as explained in my Section VII.D.1, the alleged variation in optical intensity would not have helped differentiate between the user-interface object (e.g., the menu list 1300) and the background (e.g., the standby screen and/or clock widget) because the menu is not shown until after the gesture is completed. Because the menu is not displayed during the gesture, there can be no differentiation until the menu appears ***after the gesture is completed***. Thus, differentiation cannot provide a logical motivation for why a POSITA would vary the optical intensity of the clock during Ahn's "touch and drag" gesture.

b) Modification operates contrary to Chaudhri

157. Dr. Terveen contends it obvious to decrease the optical intensity of the standby screen or portions thereof (e.g., clock widget), which he identifies as Ahn's background. EX1003, ¶98. Thus, Dr. Terveen's modification decreases the optical intensity of Ahn's background. Chaudhri-842 teaches the opposite of Dr. Terveen's modification.

158. Chaudhri-842 does not vary the optical intensity of a background. To the contrary, Chaudhri-842 discloses adjusting an optical intensity of user interface (UI) objects: "The optical intensity of a user-interface object, as used herein, is the object's degree of visual materialization." EX1005, [0085]. And Chaudhri-842 repeatedly makes clear that user interface objects are distinct from a background. *See, e.g.,* EX1005 [0085] ("optical intensity may be the visual differentiation *between* the user-interface object and *the background*"). Thus, Chaudhri-842 teaches varying the optical intensity of foreground UI elements.

159. Unlike Chaudhri-842's foreground UI elements, Ahn's clock widget is part of the background, as conceded by Dr. Terveen. *See* EX1003, ¶93 ("that a clock widget is part of the background standby screen 300"), 98 ("the background (e.g., the standby screen and/or clock widget)."). But Chaudhri-842 does not suggest adjusting the optical intensity (or otherwise altering) a background. And neither the Petition nor Dr. Terveen explain this gap in logic.

160. Further, the relied upon portion of Chaudhri-842 explains that “progress is indicated (604) towards satisfaction of the condition by transitioning an optical intensity of one or more user interface objects *associated with the second user-interface state.*” EX1005, [0081]-[0082], FIG. 6 (reproduced and annotated below).

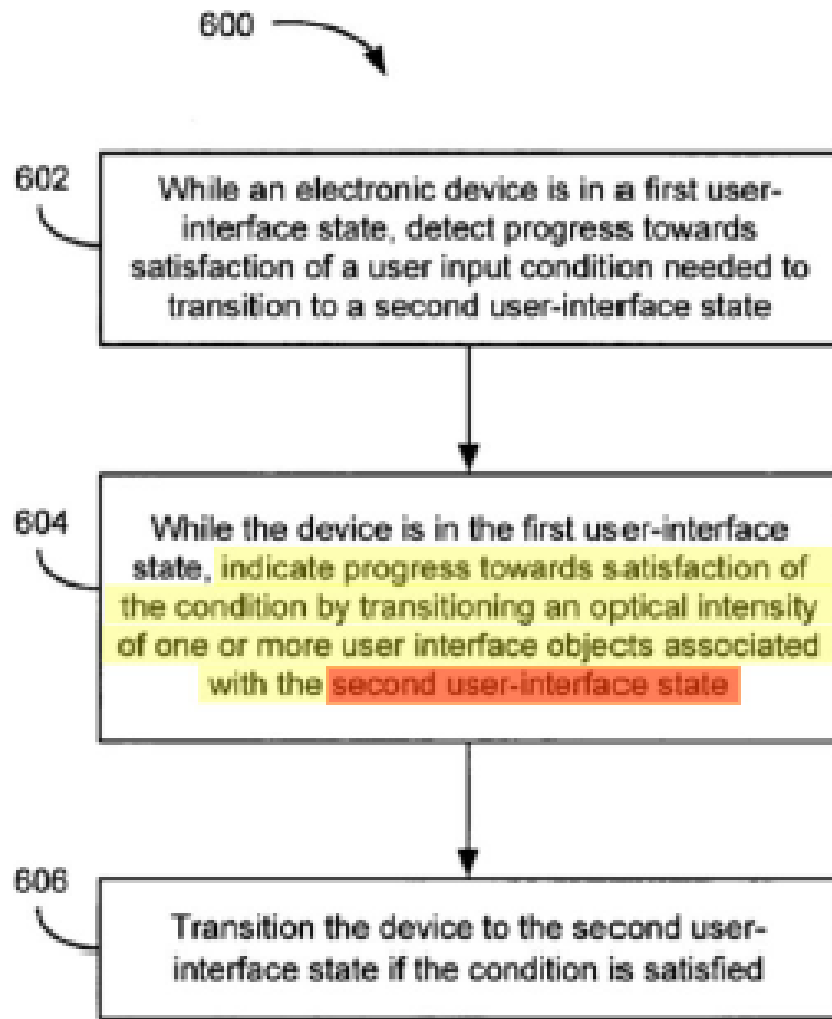


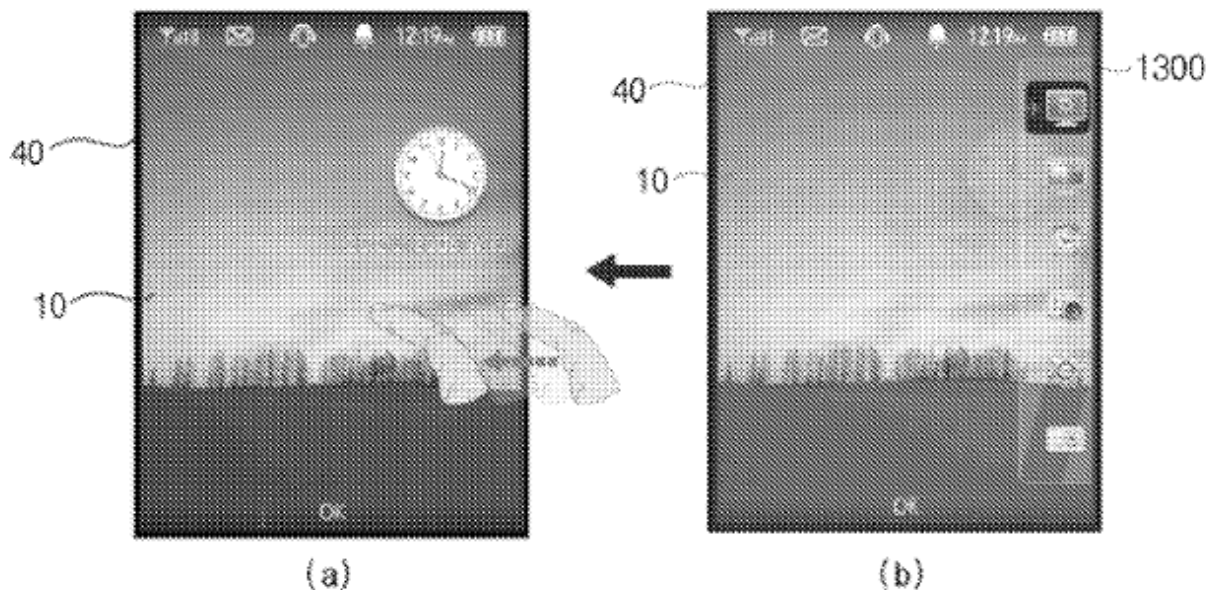
Figure 6

EX1005, FIG. 6

161. In contrast, Dr. Terveen's Ahn-Chaudhri combination relies on *reducing* the optical intensity of Ahn's background (the alleged *first* user interface state). Specifically, Dr. Terveen identifies the standby screen of Ahn (FIG. 13A – below left) as corresponding to the first user-interface state and “state where a menu

list 1300 is displayed” (FIG. 13B – below right) as the second user-interface state.
EX1003, ¶98.

FIG. 13



EX1004, FIG. 13.

162. In Dr. Terveen’s mapping to Ahn, the clock widget is displayed in (e.g., associated with) the *first* user-interface state. But Chaudhri-842 does not teach adjusting the optical intensity of an object associated with the *first* user-interface state. Thus, Dr. Terveen’s modification to change the appearance of Ahn’s clock widget (of the *first* user-interface state) is inconsistent with Chaudhri-842’s teaching to “transition[] an optical intensity of one or more user interface objects associated with the *second user-interface state*.” EX1005, ¶82. As such, here too, Dr. Terveen’s alleged modification does the opposite of what Chaudhri teaches, and a

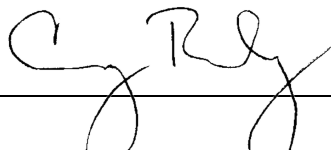
POSITA would not have made such a conflicting modification—especially not without some articulable reason, which Dr. Terveen does not provide.

VIII. CONCLUSION

163. All statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and these statements were made with the knowledge that willful false statement and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,

Dated: April 23, 2025



Craig Rosenberg, Ph.D.