

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

SMITH INTERFACE TECHNOLOGIES, LLC,

Patent Owner

Case IPR2024-01089
10,671,212

DECLARATION OF RAJEEV SURATI, Ph.D.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PROFESSIONAL BACKGROUND.....	1
III. COMPENSATION	10
IV. LEGAL CONSIDERATION.....	10
A. Obviousness.....	10
B. Claim Interpretation	11
V. Overview of Task & Basis for Opinions	12
VI. Level of Skill in the Relevant Art.....	13
VII. OVERVIEW OF THE '212 PATENT	14
A. Independent Claims 1, 16, and 84.....	15
VIII. OVERVIEW OF THE PRIOR ART	16
A. U.S. Patent Application Pub. No. 2006/0197753 (“Hotelling”).....	16
1. Figure 18: “an exemplary GUI for a main menu”	19
2. Figure 11: “an exemplary GUI for a media player”	20
B. U.S. Patent Application Pub. No. 2004/0051726 (“Martyn”)	23
IX. GROUNDS 1 AND 2 FAIL BECAUSE PETITIONER’S HOTELLING- MARTYN-CHO COMBINATION LACKS EVIDENTIARY SUPPORT	26

A.	The Petition mixes distinct embodiments from Hotelling without asserting a motivation to combine them	26
B.	Dr. Cockburn’s asserted motivations to combine Hotelling and Martyn are also deficient	33
1.	Dr. Cockburn’s first asserted motivation is a generalized assertion that lacks objective evidence	34
2.	Dr. Cockburn’s second asserted motivation manufactures a usability issue with Hotelling’s display that is unsupported by objective evidence	37
3.	Dr. Cockburn’s third asserted motivation conflates Hotelling’s music player mode with its main menu GUI	41
4.	Petitioner’s fourth asserted motivation ignores the difference between the user interface challenges of Hotelling and Martyn.....	44
X.	GROUND 1 AND 2 ALSO FAIL BECAUSE, EVEN ACCEPTING THE PROPOSED COMBINATION, THAT COMBINATION DOES NOT MEET ALL ELEMENTS OF THE CHALLENGED CLAIMS.....	46
A.	Limitations [1.c], [16.h], [84.d]: “[when] / [based on] a second duration of the touch . . . [is] / [being] detected . . . output feedback that is perceptible by touch”	46
1.	First Theory.....	47

2. Second Theory	50
XI. CONCLUSION.....	54

I. INTRODUCTION

1. My name is Dr. Rajeev Surati, Ph.D. I have been retained by Smith Interfaces Technologies, LLC as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”) regarding US Patent No. 10,671,212 (“the ’212 Patent”).

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, and researching 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 EX2002. The following provides a brief overview of some of my experience that is relevant to the matters set forth in this Declaration.

4. I hold a Doctor of Philosophy (Ph.D.) degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (“MIT”) (awarded in 1999) with a Grade Point Average of 5.0/5.0. I obtained a Master of Science degree in Electrical Engineering and Computer Science from MIT (awarded

in 1995) with a Grade Point Average of 5.0/5.0. I have a Bachelor of Science degree in Electrical Engineering from MIT (awarded 1992), graduating with a Grade Point Average of 4.9/5.0.

5. My Ph.D. thesis was entitled “Scalable Self-Calibrating Technology for Large Scale Displays.” My master’s thesis was entitled “Practical Partial Evaluation.” My undergraduate thesis, for which I received MIT EECS’s William A. Martin thesis prize for best undergraduate thesis, was entitled “A Parallelizing Compiler based on Partial Evaluation.” I was also awarded the Department of Energy’s highly-selective Computational Science Fellowship in 1995, which funded my Ph.D. studies.

6. In 1989, I began programming embedded systems and, in 1995, built one of the first publicly available Internet Coke Machines. During that time, I worked on various user interfaces. For example, one to augment the coke machine to dispense coke and another that became one of the early ecommerce mechanisms, using standards-based protocols for tracking user consumption, payments, and drink stock.

7. In 1997, while at MIT pursuing my Ph.D., I started a company called Flash Communications. At Flash, I invented an instant messaging platform focused on enterprise needs. I had the idea while observing the popularity of the Zephyr Instant Messaging Service at MIT beginning in 1988. Attracted by Flash’s

enterprise focus, Microsoft acquired Flash in 1998. With Microsoft, we built both a client and server product. The basic protocol we invented became the foundation for the well-known XMPP protocol that was widely used in the mid-2000s among instant messaging providers. I developed both the client and server products and dealt with many, if not all, of the issues one might face when implementing user interfaces for instant messaging systems. In fact, we helped to build Microsoft's famous MSN product, which adopted various user interface elements we created.

8. As a result of this experience, I became heavily involved with User Interface design on desktop machines, experience that I have continued to develop by building desktop, mobile, and web-based applications used by millions of people.

9. After graduating, I joined Microsoft (as was required by Microsoft in its acquisition of Flash Communications), where I worked on both client and server technologies related to instant messaging. My work spanned both the Microsoft Exchange Instant Messaging product that was released in 2000 and MSN Messenger.

10. From 2000 to 2007, I cofounded, led, and sold a business called photo.net to Namemedia, which is now part of GoDaddy.com. In early 2000, Photo.net was considered a top 1,000 website and it received large amounts of traffic. I built the site from its beginning as a single computer installed in a datacenter to a full rack of computers in that same data center. As a result, I became very familiar with the careful design and programming one must employ to build

and maintain systems that enable both engaging user interfaces and experiences at scale.

11. At photo.net, we prototyped many fundamental Internet community features, such as photo sharing, social networking, and memberships. The photo.net system was written on top of Oracle SQL Database and served up many millions of web pages each day (corresponding to millions of records) with high-volume inserts. The system also incorporated an ecommerce platform used to bill subscriptions, track users, etc. I spent seven years running the site, both writing new features driven by the Oracle SQL database and maintaining tables containing the many millions of records generated by the site. I also served as the database administrator, which provided me with extensive operational experience running such a system. I dealt with everything from database query speedup to the more mundane day-to-day issues, such as the maintenance of relational databases, including backups. In addition to the web interface, we also ran a mobile version of the product based on WAP, which I helped to develop.

12. After developing one of the first photo sharing sites, we further increased user engagement by designing an innovative photo rating interface. Our photo rating system became a huge draw for the site. The rating system increased engagement by four orders of magnitude, ultimately resulting in Sergey Brin, cofounder of Google, asking photo.net to provide imagery to Google for a screen

saver application. Joichi Ito, former head of the MIT Media Lab and a famous internet luminary called it “the best photo sharing site” in 2001.

13. I also innovated user interface technology at Nexaweb, where I was an investor and technical advisor. Nexaweb designed User Interfaces that worked well over the web. At Nexaweb, I spent a great deal of time solving underlying issues related to implementing a UI framework both in Java and Javascript. We also innovated a mechanism using server push prior to the AJAX becoming widely adopted.

14. In 2004, I started Scalable Display Technologies based on the technology in my Ph.D. thesis and the related patent (U.S. Patent No. 6,456,339) entitled “Super-resolution Display.” As part of my Ph.D. thesis, I designed a special display that allows one to create ultra-high-resolution displays composed of multiple projectors tiled with a slight overlap. The system uses camera-based feedback to create an inverse map that drives the system such that a person can only see a continuous, seamless display. There is no visible bezel or overlap. During this project, I became familiar with how and why to drive content onto the display. Thus, I also became familiar with content encoding/decoding, user interfaces for driving large displays, multimedia content storage, high resolution imagery, networks, recording, GPUs, storage of content, etc.

15. At Scalable Display Technologies, I developed the core algorithms and implemented—with a team—the entire software system, which included the core user interface, machine vision and embedded components, and a multimedia playback system. Our first customer was Pixar. Today, the software is primarily used by the US military and commercial theme parks, such as Disney and Universal. Scalable is well-known in the industry for having some of the easiest and intuitive software to use.

16. In parallel with my work at Scalable, I lectured at MIT on many subjects, including the Android operating system. In 2008, I worked with a group of MIT students on the development of mobile applications for Android with Rich Miner, a cofounder of Android. Since many students created applications involving Google Maps and localization, I became familiar with these technologies. I also served as a lecturer and mentor at MIT Play Labs, which was an incubator for augmented reality (AR) and virtual reality (VR) software for mobile handsets and headset applications. Notably, in addition to mobile handsets, Android runs on the Oculus headset. As part of that program, I worked with several startup companies on mobile applications, including a startup that developed applications for Telegram, which is a cross-platform, cloud-based instant messaging system.

17. Since 2014, I have worked as an independent consultant for several companies including NEC, Hitachi, Hi Marley, and Estee Lauder.

18. In 2018, I became a senior partner at nCent Labs. In that role, I consulted on the development of an incentive-market-based platform for block chains and cryptocurrency. Another part of my work at nCent Labs focused on the development of SMS messaging applications for the nCent platform. In that part, I developed a full stack application around the protocol.

19. From 2019 to 2020, I served as a Technical Lead of the Skunkworks at Hydrow, which is a startup company that develops indoor rowing machines. In this role, I worked on special projects, including the development of a VR experience using Magic Leap and Oculus that immersed users in a team-based rowing crew on a scull using VR cinematography. I leveraged my experience with user interfaces and Human-Computer Interaction (“HCI”) to develop a system that gives people a sense of rowing on the water with others, while in reality, they were rowing on a rowing machine. We accomplished our goal through a clever use of cinematography. The result was quite compelling and increased user engagement with their exercise program.

20. In 2020, I started a company called Skyline Nav AI Inc. that develops technology that uses a visual location (i.e., the skyline) to geo-locate the place a picture was taken. At Skyline, I built and now maintain a full user experience centered around a workflow for people to snap a picture and use the image’s skyline and downloaded 3D mode to localize a person. I have written applications for both

Android and iOS that communicate with a backend server running the machine vision algorithms.

21. Over the past decade, I have served as a technical consultant and expert witness on patent litigation matters. In the course of this work, I have provided consulting services to a wide variety of technology companies, including BritishTelecom, Apple, IBM, Philips, Shopify, Zillow, Polaris Powered Technologies, Amazon, Salesforce, Hitachi, Slack, Harris Teeter, and others.

22. During the course of my career, I was the inventor of the subject matter for approximately 15 U.S. Patents. I have also received additional patents, including: U.S. Patent No. 8,817,111, entitled “System and method of calibrating a display system free of variation in system input resolution,” U.S. Patent No. 8,994,757, “System and method for providing improved display quality by display adjustment and image processing using optical feedback,” U.S. Patent No. 9,215,455, “System and method of calibrating a display system free of variation in system input resolution,” U.S. Patent No. 9,369,683, “System and method for calibrating a display system using manual and semi-manual techniques,” and U.S. Patent No. 9,497,447, “System and method for color and intensity calibrating of a display system for practical usage.”

23. I am also a co-inventor of patented technology related to instant messaging, including U.S. Patent No. 5,943,478, which describes associated

technology that I developed related to pop-up, two-way messaging over the Internet. While at Microsoft, I was an inventor on several patents, including: U.S. Patent No. 6,260,148 relating to methods and systems for message forwarding and property notifications using electronic subscriptions; and U.S. Patent Nos. 6,415,318 and 6,604,133 relating to inter-enterprise messaging systems using bridgehead servers. Aspects of these patents relate specifically to messaging and notification technology in telecommunications systems.

24. I serve on the advisory boards of several technology companies, including Paneve, which develops general purpose ASICs coupled with compiler technology; Nexaweb, which develops real-time web application frameworks using HTTPS; Antix Labs, which develops compiler technology for a universal gaming platform; and Permabit, which develops content-addressable storage.

25. In addition to my above experience, I have published papers on subjects relating to computing systems, computer network communications, databases, and other subjects within the realm of electrical and computer engineering. I have also received several awards for my contributions as an inventor and entrepreneur, including the Global Indus. Technovator Award 2009 and Laureate of 2009 Computer World Honors Program. My publication and award history are listed in more detail on my CV.

III. COMPENSATION

26. My work on this matter is being billed at my customary rate of \$700 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

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

A. Obviousness

28. 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 '212 Patent (a "POSITA"), which I have been asked to treat as August 5, 2011.

29. 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

IPR2024-01089 - 10 - Patent No. 10,671,212

motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

30. 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.

B. Claim Interpretation

31. 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.

V. Overview of Task & Basis for Opinions

32. I have been asked to review the '212 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 38 below.

33. More particularly, I have been informed that this proceeding is currently at a preliminary stage during which the Patent Trial and Appeal Board (the Board) will decide whether the Petition establishes a reasonable likelihood of at least one claim of the '212 Patent being invalid. I also understand that the Board may review the Petition under a stricter “compelling merits” standard that requires the Petition to make a stronger showing of invalidity; however, my analysis is not performed using this higher standard. At this preliminary stage, I have been asked to focus my analysis on the Petition’s proposed invalidity theories for the '212 Patent’s independent claims 1, 16, and 84.

34. 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 below analysis.

35. 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.

VI. Level of Skill in the Relevant Art

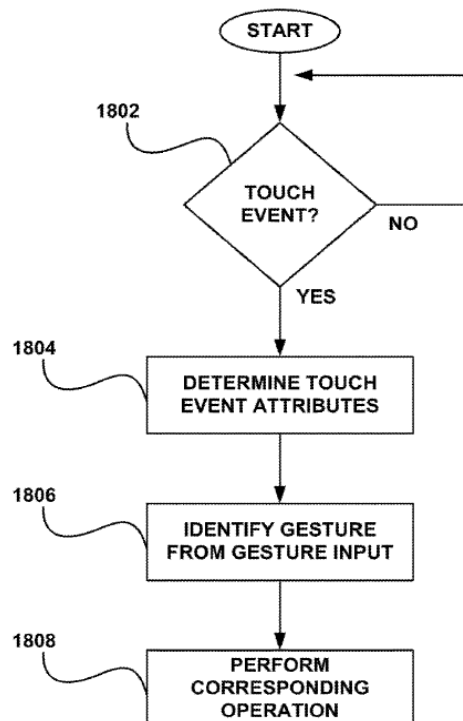
36. I understand that a patent must be analyzed from the perspective of a person of ordinary skill in the art (“POSITA”) as of the time of invention of the patent, which is typically considered to be the patent’s earliest filing date. While I do not necessarily agree with every aspect of Dr. Cockburn’s 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. For purposes of this declaration, my opinions are therefore provided from the perspective of the POSITA described in paragraph 34 of Dr. Cockburn’s declaration (EX1003), which is produced below:

a Bachelor of Science degree in computer science, computer engineering, or a comparable field and at least two years of professional experience working with graphical user interfaces and applications for

computers. Such experience could be obtained through research and study in a graduate program or through comparable exposure through industry employment, and additional years of experience could substitute for the advanced-level degree.

VII. OVERVIEW OF THE '212 PATENT

37. The '212 Patent relates to a “gesture-equipped touch screen system,” where “a signal is received in association with a touch interface” and “a user experience may be altered, utilizing the signal.” EX1001, Title, 15:9-10, 15:57-58. The patent discloses various gestures, patterns, or multi-touch inputs that can be used to improve the functionality of user interfaces for touch screen devices. See e.g., EX1001, 1:42-2:5, 2:6-31, 2:32-59, 2:60-3:15. Figure 18 depicts a high-level flowchart of the operation of one embodiment of the patent:



38. EX1001, FIG. 18. Possible “gesture inputs” used to control the choice of operation can include “touch event attributes (both initial and over time), system and/or application flags, switches, buttons, states, sensor data,” or other relevant control inputs available to the user. *Id.*, 57:19-22.

A. Independent Claims 1, 16, and 84

39. Independent claims 1, 16, and 84 are each directed to a multifunction response of “a mobile device including a touch screen” to “a touch.” “[When] / [based on]” “a first duration of a touch” “[is] / [being] detected as ceasing,” “perform[ing] an operation.” Limitations [1.b], [16.g], [84.c]. “[When] / [based on]” “a second duration of the touch, that is different than the first duration of the touch” “[is] / [being] detected” “after the first duration is detected without the ceasing”: (a) “output[ting] feedback that is perceptible by touch” and (b) “display[ing] at least one menu including a plurality of particular actions.” is displayed. Limitations [1.c], [1.d], [16.h], [16.i], [84.d], [84.e].

40. Independent claims 1, 16, and 84 further recite recognizing and acting upon additional touches in connection with “the at least one menu.” First, “[when] / [based on] a selection touch applied to the touch screen [is] / [being] detected in connection with at least one of the particular actions of the at least one menu,” “perform[ing] the at least one particular action.” Limitations [1.e], [16.j], [84.f]. Second, “[when] / [based on] a slide or swipe gesture applied to the touch screen [is]

/ [being] detected,” “chang[ing] at least one aspect of the display of the at least one menu.” Limitations [1.f], [16.k], [84.g].

VIII. OVERVIEW OF THE PRIOR ART

41. Dr. Cockburn’s analysis falters primarily with respect to two of the cited references, each of which are directed to distinct kinds of user interfaces. Each is discussed in more detail below.

A. U.S. Patent Application Pub. No. 2006/0197753 (“Hotelling”)

42. U.S. Patent Application Publication 2006/0197753 is entitled “Multi-functional hand-held device” and lists its first inventor as Steven P. Hotelling. *See* EX1004. I will refer to this reference as “Hotelling.”

43. Hotelling discloses a “multi-functional hand-held device capable of configuring user inputs based on how the device is to be used.” EX1004, Abstract, [0009]. The goal of the multi-functional hand-held device of Hotelling is to “integrate[] at least two [traditional] devices” (such as a PDA, cell phone, camera, music player, etc.) into a single device. EX1004 [0048]. In 2006, at the time of Hotelling’s invention, there was a growing interest in these sorts of multi-functional devices that would allow a user to merge each of their five or ten mobile devices into a single device that could easily be carried in their pocket. However, each of these five to ten (or more) devices featured different user interfaces, typically using

traditional physical buttons. This created a user interface challenge for developers who realized that a hundred-button multi-functional device was impractical.

44. Hotelling recognized this challenge. *See* EX1004, [0006-0007] (“traditionally separate hand-held electronic devices have begun to be combined...[i]t is a challenge to design a merged device with limited input devices without adversely affecting the dedicated inputs for each device.”). Further, “there is not enough space on the [merged] device to house all the necessary buttons and switches” to accommodate the “particular set of input mechanisms for providing inputs” used by each separate legacy device. *Id.* In alignment with user interface design fundamentals, Hotelling recognized that “it is preferable, not to overload the hand-held devices with a large number of input mechanisms as this tends to confuse the user.” *Id.*

45. Instead of overwhelming the user, Hotelling proposes “virtually incorporating the physical inputs for each functionality into the GUI,” which “adapt[s] to whatever device is selected.” EX1004, [0069]. Then, the device has “a GUI for each functionality.” *Id.* By replacing physical buttons with virtual buttons on a touch screen, Hotelling’s design has the added benefit that the “display size can be substantially increased.” EX1004, [0009]. While it was not the main thrust of Hotelling’s invention, it is significant that Hotelling’s virtual button design allowed for larger screens. Devices before Hotelling were heavily restricted by the amount

of space required for buttons, including dial pads, keyboards, scroll wheels, and more. Larger screens like the one disclosed in Hotelling substantially expand the design space available to user interface developers.

46. Hotelling also discloses the use of a “haptics mechanism” to “enhance the feel of actuating a GUI element” or to provide “alert[s]” to the user. EX1004, [0161-0163]. Hotelling teaches that “[t]his particular feature enhances the users experience and makes the virtual UI feel more like a physical UI.” EX1004, [0161]. Feel is important for a user interface. By designing a user interface that is similar to other interfaces with which users are familiar, a designer can improve ease-of-use. “For example, in response to touching a virtual button of the GUI, the haptics may provide force feedback in the form of vibration and the speaker may provide audio feedback in the form of a click.” EX1004, [0163]. This feedback tells the user that “they actually implemented an input” by “simulat[ing] the audio and tactile feel of a button or switch.” *Id.* Alternatively, the feedback “may be used to simulate a button click” through a “click” and “clock” output “when a hard touch is made.” *Id.*

47. Since Dr. Cockburn focuses specifically on two of the user interface designs described in Hotelling—one shown in Figure 18 and another in Figure 11—I will analyze them in more detail.

1. **Figure 18: “an exemplary GUI for a main menu”**

48. In one user interface design, Hotelling discloses “an exemplary main menu GUI 260 of a multi-functional device.” EX1004, [0085], FIG. 18 (inset below).

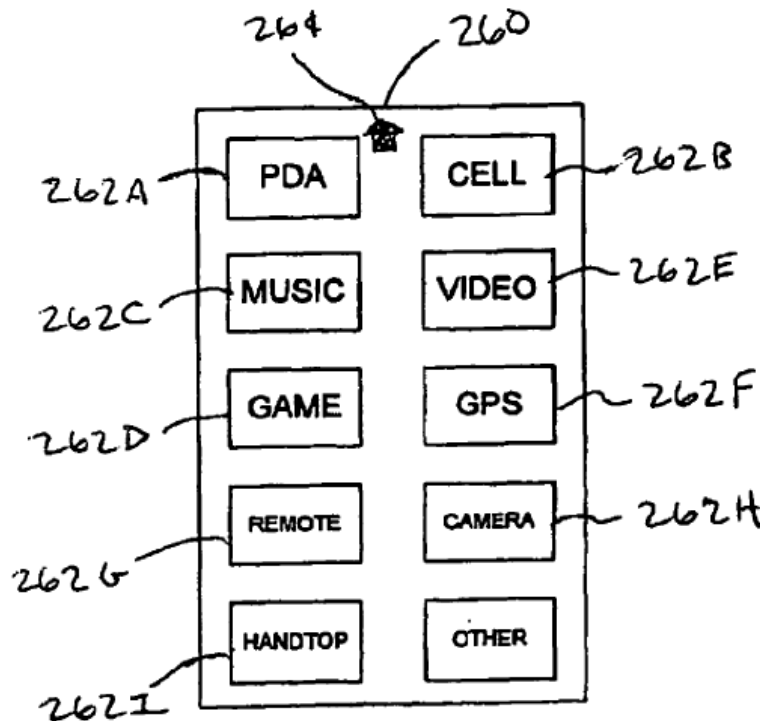


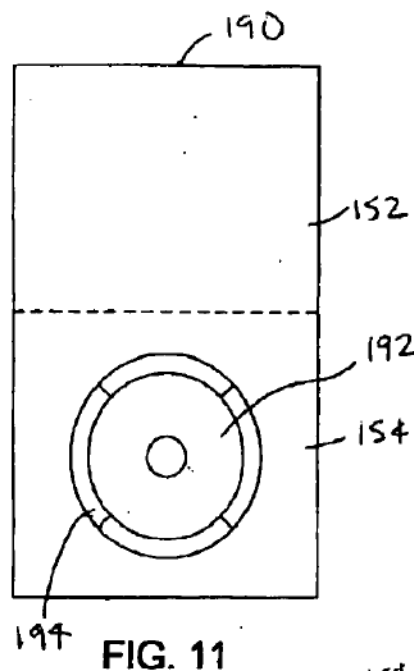
FIG. 18

EX1004, FIG. 18. When the user selects one of the various “virtual buttons” (e.g., “a music player button 262C”) shown in the Figure, “the main page for the selected functional[ity]” is shown on the display. EX1004, [0085]. If the user wishes to select a different functionality, “the user simply selects a soft home button...to return to the main menu page 260.” *Id.* This design ensures that, “[b]y keeping different

functionalities distinct, user confusion may be reduced.” EX1004, [0091]. Similar to its mimicking of physical buttons, Hotelling again leverages user background experience with this design. Before the advent of multi-functional, hand-held devices, users were accustomed to separate, distinct devices that performed each of the functionalities pictured in the figure above. By cabining each separate functionality behind a simple, easy-to-navigate main menu GUI, Hotelling recreates the user’s prior experience with separate devices. Reducing user confusion, which contributes to ease-of-use, is a critical goal of user interface design.

2. *Figure 11: “an exemplary GUI for a media player”*

49. In a second user interface design, Hotelling discloses “an exemplary GUI for a media player.” EX1004, [0022], [0077], FIG. 11 (inset below).



50. EX1004, FIG. 11. Specifically, Figure 11 depicts a “GUI 190 that is used in a music player mode.” EX1004, [0077]. Consistent with Hotelling’s goal of providing “a GUI for each functionality” (EX1004, [0069]), a POSITA would understand that Figures 18 and 11 disclose separate embodiments of different GUIs. This is apparent from the different names used by Hotelling, the different figures, and the different button layouts. The “music player mode” GUI is “divided into a standard region 152 and a control region 154.” *Id.* The “control region” further contains a “virtual scroll wheel 192 and five virtual buttons 194.” *Id.*

51. Hotelling also discloses “one example of a touch vocabulary associated with a music player” that could make use of the “music player mode” GUI depicted in Figure 11. EX1004, [0022], [0038], [0077], [0151], FIGS. 11, 27A-E. A “touch vocabulary” is a set of “behaviors and user feedback” that a device can display based on a set of input variables, including “the UI mode, the force of the touch (e.g., light or hard), the number of fingers used, whether or not there is any movement during the touch, the duration of the touch, and the touch location.” EX1004, [0145-0150], FIG. 26. As Hotelling recognized, touch vocabularies were a key part of the new design space created by touch screen devices. Multi-gesture inputs were usually unavailable on legacy devices, limiting input options.

52. User interfaces evolved with the introduction of touch vocabularies, beginning with designs like Hotelling and continuing through the more complex

designs in use today. Figure 27A (inset below) discloses an example of a touch vocabulary that would recreate the button functionality commonly found on portable music players using the “virtual buttons” shown in Figure 11. EX1004. Portable music players used physical buttons that were specific to music player features, such as “play,” “pause,” “next song,” etc. However, these functions are not useful to every portable device. Hotelling proposed that this problem, described above, be solved by the use of dynamic, virtual buttons.

#	UI Mode	Light Touch or Tactile Click	# Fingers used	Movement during touch or click	Duration	Touchdown location (virtual button?)	Proposed Behavior	Feedback
Basic list navigation								
List Nav.	light		1	none	any	anywhere	On touch-down, do nothing. On lift-off, highlight item(if any) under touch	highlight item touched
List Nav.	click		1	NONE	short	virtual button / list item	On touch-down, do nothing. On click-down highlight item under touch (if any). On un-click, execute the item clicked	animate screen sliding RIGHT to expose new screen
List Nav.	click		1	NONE	short	PLAY / PAUSE virtual button	On touch-down, do nothing. On click-down highlight Play/Pause icon. On un-click, PLAY/PAUSE	Change PLAY/PAUSE icon
List Nav.	click		1	NONE	LONG	virtual button	On touch-down, do nothing. On click-down highlight item under touch (if any). After delay, bring-up context sensitive menu.	Context sensitive menu pop-up
List Nav.	light or click		1	UP/DOWN	any	anywhere	On touch-down, do nothing. Scroll list UP/DOWN depending on UP/DOWN distance moved, use inertia & brakes, disregard LEFT/RIGHT motion during gesture. If highlighted item slides off visible screen, change highlighted item to one that is visible.	list scrolling UP/DOWN, first few letters enlarged
List Nav.	light or click		1	LEFT	any	anywhere	On touch-down, do nothing. On left motion, open highlighted folder or execute highlighted item (like legacy SELECT button)	animate screen sliding LEFT
List Nav.	light or click		1	RIGHT	any	anywhere	On touch-down, do nothing. On right motion, go back out a folder (like legacy MENU function)	animate screen sliding RIGHT

FIG. 27A

53. EX1004, FIG. 27A, *see also* [0038]. As shown in the figure, this “touch vocabulary” recreates the features that required physical buttons on “legacy” music players, such as “PLAY/PAUSE,” “SELECT,” or “MENU.” *See id.*

B. U.S. Patent Application Pub. No. 2004/0051726 (“Martyn”)

54. U.S. Patent Application Publication No. 2004/0051726 is titled “Computing device with improved user interface for applications” and lists as its inventor Mathieu Martyn. I will refer to this reference as “Martyn.”

55. Martyn discloses an “application summary window” for “small screen devices” like the mobile phones of the late 1990s and early 2000s. EX1005, [0009]. The summary window displays “a limited list of common functions and commonly accessed stored data” that “saves the user from navigating to the required application, opening it up, and then navigating within that application to enable the data of interest to be seen.” EX1005, [0009]-[0015]. Martyn also uses the descriptors “App Snapshot” and “snapshot view” to describe this summary window. EX1005, [0009]-[0010], [0015], [0017]-[0026], [0028]-[0031].

56. This “‘snap-shot’ view” addresses the need for “user interface[s] . . . to demonstrate or make explicit the changing internal status of [a] mobile telephone as navigation proceeds.” EX1005, [0008]. “For example, to select or initiate a function . . . a user has to understand (a) how to navigate to that function in order to select that function and (b) that the status of the telephone is such that the function can be selected or initiated.” *Id.*

57. To reach the summary window, a user starts by looking at a device in an “application launcher view.” EX1005, [0016]. “The Application Launcher view

is . . . presented as a standard scrolling list of application[s].” *Id.* “The list is vertical and only one application is presented per line.” Navigating this view is accomplished by “[s]tandard highlight functions.” *Id.* A “highlight” indicates which application the user currently has the choice of selecting. *Id.* A “user may take conventional navigation steps, such as scrolling, to move the highlight and using the available select function to navigate to the required highlighted application.” *Id.* “The innovative summary window functionality can be accessed as follows: should the highlight rest on the name of an application in the App Launcher for a certain amount of time (say, a 1.2 second timeout), the summary window (the ‘App Snapshot’) drops down from the highlight bar.” EX1005, [0017]. “Other selection processes could also be used (e.g., voice activation, softkey selection, etc) to access the App Snapshot.” *Id.* Additionally, “the App Snapshot may be ‘called’ . . . by using a right scroll function . . . a press and hold of the select function when an application name is highlighted; or using a right cursor key on a highlighted application name.” *Id.* This process is shown in Figures 1 and 2 below. In these figures the red highlight rests on the messages application and by leaving the

highlight over the messages application a blue summary window is accessed.



Figure 1

EX1005, Figure 1 (annotated).

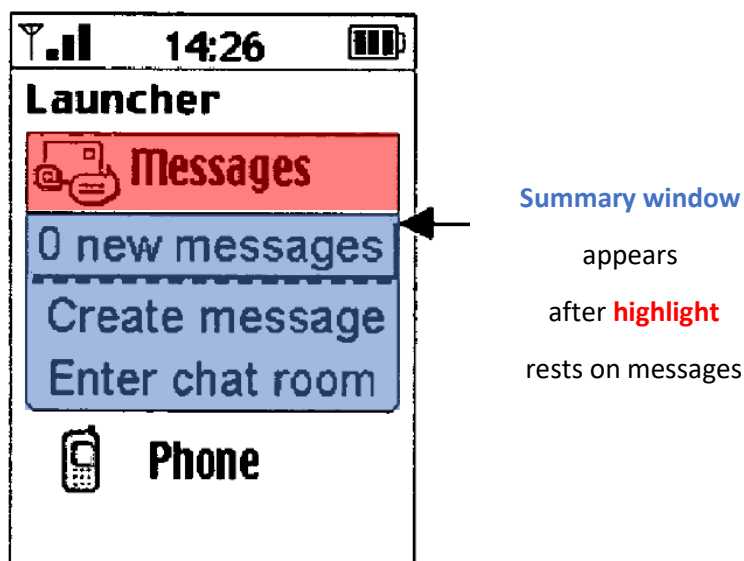


Figure 2

EX1005, Figure 2 (annotated).

58. Martyn also discloses ways to dismiss the summary window based on how it was called. “If a timeout is used to call the App Snapshot, scrolling the highlight back to the relevant application name” or “[s]crolling down through the App Snapshot” “removes the App Snapshot.” EX1005, [0021]. “If scroll functions (e.g. right scroll) call the App Snapshot the opposite scroll function removes the snapshot.” EX1005, [0022].

IX. GROUNDS 1 AND 2 FAIL BECAUSE PETITIONER’S HOTELLING-MARTYN-CHO COMBINATION LACKS EVIDENTIARY SUPPORT

59. Both Petitioner’s and Dr. Cockburn’s analysis of Hotelling and Martyn misunderstand these references. As a result, their proposed Hotelling-Martyn-Cho Combination is unfounded. For the reasons explained below, a POSITA would not have looked to a combination of these references to design a new user interface.

A. The Petition mixes distinct embodiments from Hotelling without asserting a motivation to combine them

60. Dr. Cockburn’s analysis (and, relatedly, the Petition) mixes and matches different embodiments from Hotelling without any explanation as to why a POSITA would do so. *See, e.g.*, EX1003, ¶¶100-137. Dr. Cockburn asserts that a POSITA would have understood Figure 18 of Hotelling to disclose “display indicia,” as claimed in claim 1 of the ’212 Patent. *E.g.*, EX1003, ¶¶106, 111 (“the GUI 260 of Hotelling’s mobile device displays a main page of ‘virtual buttons’ in the form of ‘icons’ (*indicia*)”) (emphasis in original). However, Dr. Cockburn then asserts that

the functionality associated with the “display indicia” of claim 1 is disclosed by the touch vocabulary of Hotelling’s music player mode. *See e.g.*, EX1003, ¶¶119, 122, 131. But, this is a different user interface design. Dr. Cockburn appears to confuse the “main menu GUI 260” with the “music player mode” GUI (FIG. 11) detailed in Hotelling. *Compare* EX1003, ¶111 *with* EX1003, ¶119.

61. Dr. Cockburn cites Figure 18 of Hotelling as disclosing the “display indicia” described in Limitations [1a], [1c], and [1d]¹, but turns to Figures 27A-E for the disclosure of the claimed functionality associated with this indicia in Limitations [1b-f]. *See* EX1003, ¶¶100-137. As explained in my overview of the prior art, these figures relate to two distinct user interface designs of Hotelling that describe separate user interfaces with separate purposes and functionality. *See supra*, §VIII.A. One is “an exemplary main menu GUI.” EX1004, [0085], FIG. 18. The other is “a touch vocabulary associated with *a music player*.” EX1004, [0151] (emphasis added), FIGS. 27A-E.

62. A POSITA would understand that the design considerations for a main menu are quite different from those for a music player. Main menus require a certain level of elegance and simplicity. On a main menu, a user must be able to easily find the application, sub menu, or data they would like to access. Easy navigation is prioritized over functionality because the main menu, by definition, contains

¹ Similar limitations at [16f/16h/16i] and [84b/84d/84e].

applications that execute each of a device's functions. A music player, on the other hand, is an end-of-the-line application. There is no sub-application to supplement the music player's functionality. As a result, its user interface must prioritize functionality. Thus, the music player interface must contain a button or gesture to capture each and every function a user could want to perform, within reason. To accommodate this functionality, the possibility of user confusion is deprioritized in an application GUI, such as the music player GUI depicted in Figure 11 of Hotelling. User confusion, however, is a top priority for main menu GUIs. *See* EX1004, [0091].

63. Without addressing this obvious design-decision conflict, Dr. Cockburn grafts the input functionality of Hotelling's music player onto its main menu design. The two modes are reproduced below to further illustrate the conflict between user interface design goals:

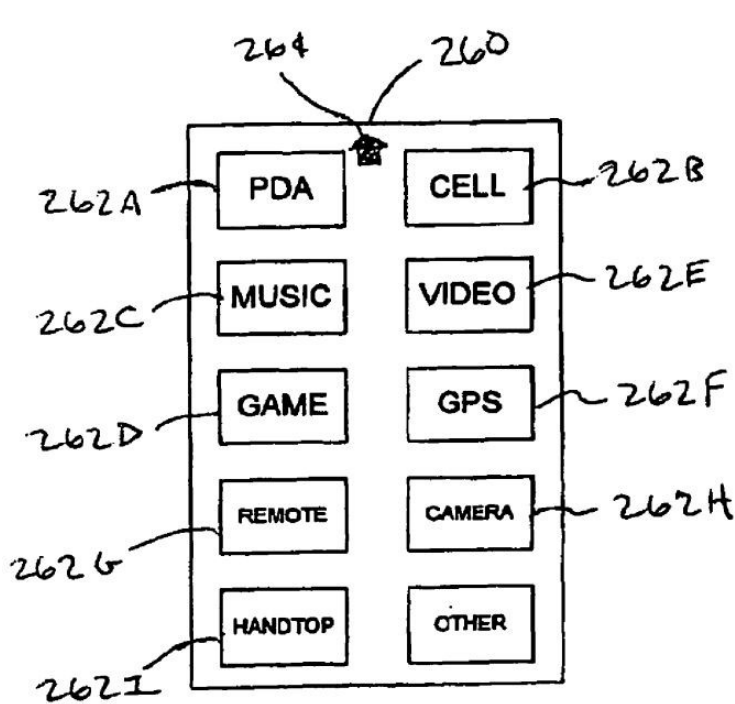


FIG. 18

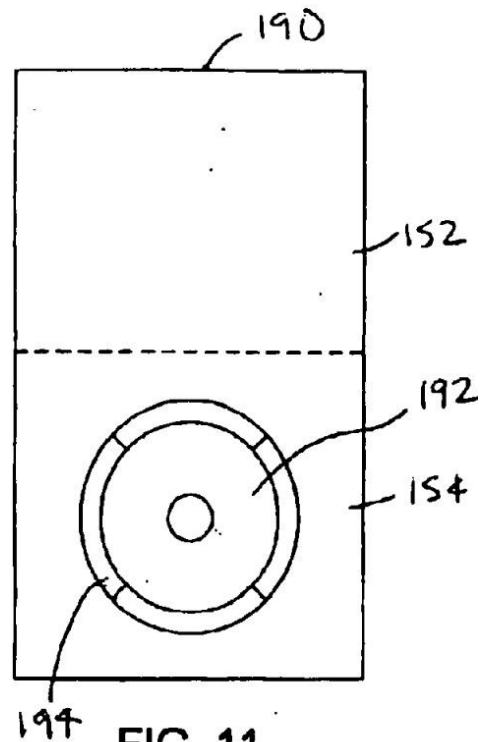


FIG. 11

EX1004, Figures 11, 18. As shown above, these embodiments are different GUIs with different buttons, different functions, and different purposes. A POSITA would understand that Figure 18 represents a main menu interface and Figure 11 represents an application interface. A POSITA would further understand that main menus prioritize simple navigation and applications prioritize complete functionality. Dr. Cockburn glosses over this distinction, making his analysis incomplete and his conclusions inaccurate as a result.

64. The relevant portions of Hotelling's specification cited by Dr. Cockburn in his mapping of Hotelling to the limitations of claim 1 are as follows:

65. ¶¶[0084]-[0085], ¶[0097], ¶¶[0172]-[0173] (describing the “main menu GUI,” shown in Figure 18, including “icons/buttons 262 for launching each of the various device functionalities”);

66. ¶[0151], Figures 27A-E (describing a “touch vocabulary associated with a music player” mode);

67. ¶[0145], ¶¶[0149]-[0150] (describing possible variables for a “touch vocabulary” for each device mode’s functionality).

68. As explained, Dr. Cockburn conflates these distinct user interface designs in his mapping of the limitations of claim 1 of the ’212 patent, treating them as if they are an interface and touch vocabulary for a single menu. *See e.g.*, EX1003, ¶105 (citing Figure 18 of Hotelling as disclosing the claimed “indicia”), ¶116 (citing Figures 27A-27B and ¶¶[0149]-[0150] of Hotelling as teaching “short” and “long” touches that allegedly correspond to a “first duration” and “second duration”), ¶¶122-131 (citing [0145], [0149]-[0151] as disclosing the functionality of the claimed “indicia” described in Limitations [1c] and [1d]). That analysis is deficient. A POSITA would understand these portions of Hotelling to describe two different solutions to two different user interface design challenges. One solution is directed to a main menu, the other to a music player. It would be obvious to a POSITA that the touch vocabulary shown in Figures 27A-E is for a music player interface because

that is exactly what Hotelling says. EX1004 [0038] (“FIGS. 27A-E are tables representing one example of a touch vocabulary associated with a music player”).

69. Dr. Cockburn therefore confuses the functionality Hotelling associates with an operational mode (e.g., touch vocabulary associated with a music player mode as shown in FIGS. 27A-E) with Hotelling’s “main menu GUI.” Further, he does not explain any rationale for combining them (or pointing to anything in Hotelling that would purportedly motivate a POSITA to do so). See EX1003, ¶106 (“In describing Figure 18, Hotelling states that the graphical user interface (GUI 260) provides a “main menu page” presenting a number of ‘icons/buttons 262 for launching each of the various device functionalities.’”); cf EX1003, ¶122 (“A POSITA would have recognized these ‘short’ and ‘long’ duration touches as *first duration* and *second duration* touches) (citing EX1004, [0145], [0149]-[0151], FIGS. 26, 27A-27B). While it is possible that a POSITA would have understood these features in the way Dr. Cockburn proposes, a POSITA also would have recognized that they come from two different interface designs from Hotelling and should not necessarily be combined without additional testing. Dr. Cockburn does not address this important point, which colors his entire analysis.

70. Further, Hotelling’s simple main menu GUI is inconsistent with the use of a complicated set of touch commands (such as the music player mode touch vocabulary) on its “main menu GUI.” As explained above, the design considerations

between a main menu user interface and an application interface are fundamentally different. The former aims for simplicity and ease-of-use, the latter for function. Hotelling's teaches are merely a manifestation of this concept. Towards that end, the only functionality that Hotelling describes for its main menu buttons is "launching each of the various device functionalities." EX1004, [0085]; *see also* [0084] (describing how, after an icon is selected, "[f]rom that point on, the multi-functional device operates like the selected device"). This is consistent with both Hotelling's teaching that its main menu should reduce user confusion and a POSITA's background knowledge of main menu interfaces in the art, generally. In further support of this teaching, the only section in Hotelling that discusses the "main menu GUI," as depicted in Figure 18, is entitled simply "Switching Between Devices." EX1004, 6.

71. A POSITA would have understood that Hotelling's main menu is designed for simplicity. This is consistent both with Hotelling's specification and well-known user interface design principles. Main menus are switchboards used to reach each separate application. Hotelling supports this well-known principle, stating that "it is preferable, not to overload the hand-held devices with a large number of input mechanisms" and, "[b]y keeping different functionalities distinct, user confusion may be reduced." EX1004, [0007], [0091]. Incorporating a set of complicated, multi-touch gestures into the main menu would do nothing but frustrate

this goal. Hotelling did not contemplate that its main menu would be used for more than that because, as explained above, Hotelling was aimed at the combination of multiple handheld devices that had previously been separate. A complicated main menu would have strayed too far from a user's past experience, thereby increasing confusion rather than reducing it. Neither Dr. Cockburn nor the Petition explain why a POSITA would ignore both fundamental user interface design principles and Hotelling's teachings of simple functions for its main menu GUI.

72. Accordingly, Dr. Cockburn's mapping of "indicia associated with an application" to one embodiment of Hotelling and the indicia's related functionality, such as a responding to a "first duration" and "second duration" touch associated with "the indicia" is unsupported in both Dr. Cockburn and Hotelling itself.

B. Dr. Cockburn's asserted motivations to combine Hotelling and Martyn are also deficient

73. A POSITA would not have found it obvious to combine the separate elements of Hotelling discussed above, much less combine them with the additional design of Martyn. User interface design for touch screens and user interface design for older, legacy devices are quite different. While the two share some design principles, the best methods for accomplishing those principles are different, which was appreciated in 2011 and has continued to be true in technology today. I will address each reason to combine Hotelling with Martyn provided by Dr. Cockburn in

turn. None of them are supported by Hotelling or the state of the art at the time of the '212 Patent.

1. Dr. Cockburn's first asserted motivation is a generalized assertion that lacks objective evidence

74. Dr. Cockburn's first reason to combine is a generalized assertion about the "ubiquitous" use of "contextual menus" in computer technology at the time of the '212 patent. *See* EX1003, ¶¶78-79. Dr. Cockburn asserts that, since "a POSITA would have known that '[a]ll . . . applications have menus'" and menus are a "central 'ingredient'" of a user interface, a POSITA "would have found it beneficial and obvious [to combine Martyn's contextual menu] in the context of Hotelling's 'main menu GUI'" *Id.* While it is true that menus have been used in digital devices for decades, that does not explain why a POSITA would be motivated to combine *this* menu with *this* interface. Broad statements about menus, generally, give no credit to the field of user interface design, in which how and why you use which menus when is studied, debated, and the subject of intense research. A POSITA would know this. Further, a POSITA would appreciate that these choices matter and their implications are not obvious.

75. In discussing this particular reason to combine the references, Dr. Cockburn fails to address that Hotelling and Martyn are different menus for different devices with different design constraints. Martyn is directed to problems with "small screen" devices. EX1005, [0009]. It was published at a time when touch screen

mobile devices were not common. Even though some touch screen devices existed, almost all mobile devices still used conventional screens with physical buttons. To understand this point, one need only look to Martyn's own disclosure.

76. Martyn does not make a single reference to touch screens or touch screen devices. Martyn does not contemplate gesture-based inputs, virtual buttons, or touch-based menu navigation, features that would all allow the user to navigate the device more efficiently, a key goal of Martyn. *See e.g., id.*, [0005] (“with conventional user interfaces, a user may need to scroll around and switch view many times to find the right data/functionality”). Quite the opposite, Martyn recommends against multi-gesture inputs, teaching that “shortcuts” that “involve[] memorizing various keyboard input sequences” to “get straight to a particular function . . . appeal only to a small number of expert users.” *Id.*, [0007].

77. Hotelling, however, describes a device that “preferably uses a touch screen as the primary input device.” EX1004, [0061]. Hotelling's use of a touch screen “substantially increase[s]” the “display size” over similar mobile devices, reducing the need to optimize screen organization. *See id.*, [0009]. The addition of virtual buttons, paired with a larger screen changes the user interface design challenge for a mobile device. Since it includes these features, the Hotelling device did not face the same user interface challenges as the device disclosed in Martyn. For example, unlike a user of the Martyn device, a user of the Hotelling device could

access application functionality in as few as two steps (i.e., launch the main menu and then launch the device mode where users could access core functionality associated with the device mode). *See e.g.*, EX1004, FIG. 11 (depicting virtual scroll wheel and five virtual buttons associated with a “music player mode.”).

78. A POSITA would not have been motivated to look to Martyn for its App Snapshot functionality, which addressed problems faced by older mobile devices. The App Snapshot of Martyn adds value to the interface of a device like Martyn’s, which has a small screen with room for only a few options at a time, fixed physical buttons that cannot adapt between applications with different functionality, and an overall difficult user interface challenge that requires distilling information as efficiently as possible. This is not equally true for devices with larger screens and variable button layouts. User interface design is a delicate balance of the functional with the simple and the intuitive. Martyn’s device differs in important respects from Hotelling’s device that fundamentally change this balance’s calculus.

79. Dr. Cockburn’s generic statement that menus were used in mobile devices is true. However, this does not mean that all menus are created equally, for the same purposes, and are useful in the same contexts. A POSITA, understanding the difference between Hotelling and menus designed for older devices, would not have looked to Martyn to solve the design challenges faced by this new, touch screen design.

2. *Dr. Cockburn's second asserted motivation manufactures a usability issue with Hotelling's display that is unsupported by objective evidence*

80. Dr. Cockburn's second reason to combine manufactures a usability issue with Hotelling's display that is unsupported by Hotelling and overstated as an existing problem in the art. Then, Dr. Cockburn proposes to solve it with Martyn, which features all of the substantially worse usability issues faced by older devices lacking touch screens. The cell phones and other mobile devices of Martyn's era had to manage slower processing devices with tiny screens and a limited set of physical buttons for navigation. As a result of this design challenge, user interface designers were forced to sacrifice elegance for functionality in a very limited design space (i.e., few inputs, limited interface options, slower processors, etc.).

81. Dr. Cockburn asserts that "the Hotelling-Martyn combination would provide a user experience that 'is far faster and easier than conventional navigation approaches.'" EX1003, ¶80 (citing EX1005, [0015]). Dr. Cockburn implies that the combination of Hotelling and Martyn would be far faster and easier than the navigation approaches considered "conventional" for touch screen devices in 2011. This is not, however, what the cited text from Martyn is referring to. As explained, these are two different devices from different eras. "Conventional" designs at the time of Martyn had to manage tiny screens with a limited set of physical buttons to navigate that screen. To say that Martyn's menu is faster and easier than the

conventional approaches of 2001 (when Martyn was invented) is not the same as saying it is equally faster and easier than the conventional approaches known in 2011. In fact, user interface design was evolving so rapidly at the time that a POSITA would doubt this to be true without further experimentation. Certainly, it would be far from predictable.

82. A POSITA, Dr. Cockburn asserts, “would have known, usability in this context . . . was a key differentiator amongst competing mobile devices.” EX1003, ¶82. Dr. Cockburn then asserts that “[t]he mobile device with a superior interface from a usability standpoint would be more attractive to consumers.” *Id.* But, Dr. Cockburn again ignores the fundamental differences in user interface design between a touch screen phone and a 2000s-era cell phone with no touch screen, a scroll wheel, and menus that display four options at a time. *Compare* EX1004 [0061] (“the hand-held device preferably uses a touch screen”) *with* EX1005, FIG. 1 (depicting a non-touch screen device with limited visible options).

83. The usability issues faced by Martyn and any usability issues faced by Hotelling are fundamentally different because of the evolution of user interface technology. Dr. Cockburn’s assertion that the same solution (Martyn’s App Snapshot menu) would resolve the problems of both devices without additional experimentation is unfounded. A POSITA would have understood Martyn’s App Snapshot menu as a solution for older mobile devices containing physical buttons,

keys, and scroll wheels. Martyn describes the use of a physical scroll wheel and a “right cursor key” to launch the App Snapshot. *See* EX1005, [0018]. A POSITA would have understood that this was slower, less efficient, and more frustrating for a user than Hotelling’s touch-screen display.

84. A POSITA would not have been motivated to look to Martyn for its App Snapshot functionality because it was an older design directed to legacy devices. Again, as explained above, menus (and user interfaces, generally) are designed for the contexts in which they are used. When designing a new menu, POSITAs look to similar contexts, or environments for ideas that utilize the tools and manage the constraints of *that* environment. This is especially true with respect to usability, a chief goal of user interface design. The device of Martyn, by virtue of being a device with physical buttons and a tiny screen, has substantially worse usability problems than the device of Hotelling. A POSITA would not look to Martyn for user interface solutions in Hotelling’s environment.

85. In the late 2000s, when mobile device technology made it economical for touch screens to become widespread on mobile devices, user interface design entered a sort of renaissance. Touch screens opened the door to a whole new world of gestures, screen organization, and navigation possibilities that revolutionized the world of mobile devices. Both the usability problems and solutions changed radically with the advent of devices featuring full-screen touch screens and the

variable button schemes allowed by virtual buttons. At the same time, the commercial pressure to combine what had previously been many separate devices (as described in Hotelling) into a single device added complexity beyond what was common in mobile devices in 2001. Both the new challenges and new resources made user interface design with devices like the one described in Hotelling a new, different design challenge. It was not obvious that the solutions of legacy devices would be equally desirable for this new generation of mobile devices.

86. Those differences mean that Dr. Cockburn’s failure to explain *why* or *how* its proposed modifications to Hotelling would have resulted in any improvement to “usability” renders its conclusions completely unsupported. *See* EX1003, ¶¶80-82; *see also* Pet., 15. For example, Dr. Cockburn mentions “fewer user inputs,” but fails to compare user inputs in Hotelling versus the purported Hotelling-Martyn combination. *See id.* Nor does Dr. Cockburn offer any comparison in speed of performing similar functions in Hotelling versus the purported Hotelling-Martyn combination. *See id.* Dr. Cockburn’s failure to actually compare the two is fatal to his second motivation. For example, Martyn’s interface required a user to highlight—with a limited input (scroll wheel)—a desired application and then wait for a timeout. EX1005, [0017] (“The innovative summary window functionality can be accessed as follows: should the highlight rest on the name of an application in the App Launcher for a certain amount of time (*say a 1.2*

second timeout), the summary window (the ‘App Snapshot’) drops down from the highlight bar.” (emphasis added)). While such a pop up menu may have been desirable when stuck with Martyn’s comparatively rudimentary input, the desirability of such a menu is not a foregone conclusion with a touch screen. For example, Hotelling’s touch screen would have allowed a user to simply open an app and choose a function more easily and much more quickly than with a long press. Dr. Cockburn therefore fails to support his conclusory assertion of any improvement in usability.

87. Given the paradigm shift in user interfaces, a POSITA would not have looked to a reference such as Martyn with an outdated interface for the solutions of the future. Dr. Cockburn’s second reason to combine Hotelling and Martyn is contrary to the way in which a POSITA would understand these references. It is based on a flawed comparison of two wildly different usability puzzles and a purported solution that is unsupported by a POSITA’s knowledge.

3. *Dr. Cockburn’s third asserted motivation conflates Hotelling’s music player mode with its main menu GUI*

88. Dr. Cockburn’s third reason to combine Hotelling and Martyn again confuses Hotelling’s “touch vocabulary for a music player mode” with its proposed functionality of its “main menu page GUI.” Dr. Cockburn asserts that “a POSITA would have been prompted to pursue a combination with Martyn based on the teachings and motivations evident from Hotelling itself.” EX1003, ¶83. However,

Dr. Cockburn then cites to Hotelling's teaching of a "context sensitive menu pop-up" in Hotelling's "music player mode" embodiment as support for this assertion. *Id.* (citing EX1004, [0151], Figures 27A-27B). As explained, however, the purposes, functions, layout, buttons, and design constraints for the main menu and music player of Hotelling are different. *See supra*, §IX.A. Dr. Cockburn further asserts that "a POSITA would have been prompted to seek out references that provide additional details on how to beneficially implement such a context sensitive menu pop-up," referring to the "context sensitive menu pop-up" in Hotelling's music player. EX1003, ¶83. Dr. Cockburn fails to explain why a POSITA would look to a design for a menu that is *not* for a music player. Since he confuses Hotelling's music player touch vocabulary with its main menu GUI, Dr. Cockburn has missed a critical detail that would change a POSITA's search. As explained, context is critical when sculpting a user interface.

89. The "context sensitive menu pop-up" of Hotelling's "music player mode" is just that: **context** sensitive. *See* EX1004, FIGS. 27A-E. Dr. Cockburn points to Martyn's "exemplary GUI element," "content," and "various ways to interact with the menu" as three reasons that a POSITA would see Martyn as "a natural and obvious extension of Hotelling." EX1003, ¶¶83-84. But, this conclusion only makes sense if one confuses the main menu GUI of Hotelling with its music player mode, as Dr. Cockburn does.

90. Hotelling teaches that all three of these examples are different between main menus and music players. First, the GUI elements between Hotelling's main menu and music player are different.

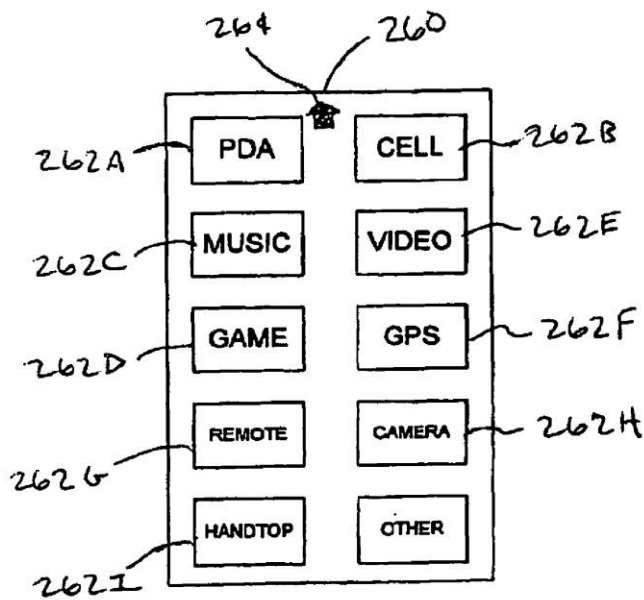


FIG. 18

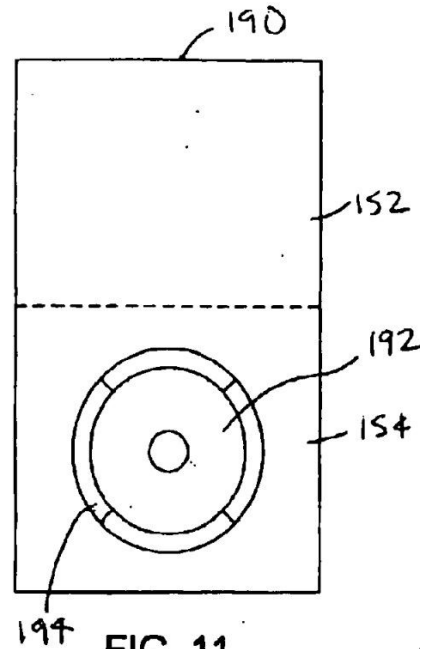


FIG. 11

EX1004, Figures 11 and 18. Second, the content is also different between the two interfaces. Finally, with different GUI elements and content, the various ways to interact with the menu would also be different. Hotelling understood this. Hotelling makes clear that the touch vocabulary described in Figures 27A-E is for a “music player mode,” specifically. EX1004, [0038]. A POSITA would understand this to imply that touch vocabularies should be interface-specific. This is also consistent with a POSITA’s background knowledge that context matters in user interface design.

91. Dr. Cockburn’s reliance on Hotelling’s disclosure of a “context sensitive menu pop-up” for a music player mode as a suggestion that Martyn’s App Snapshot should be combined with Hotelling’s music player is misguided because it misunderstands Hotelling’s distinct interface designs. As a result, Dr. Cockburn’s third reason to combine Hotelling and Martyn fails.

4. Petitioner’s fourth asserted motivation ignores the difference between the user interface challenges of Hotelling and Martyn

92. Similar to the first three reasons in his analysis, Dr. Cockburn’s fourth reason to combine ignores fundamental differences in the obstacles faced by Martyn and Hotelling. Dr. Cockburn asserts that Martyn identifies “specific advantages” of Martyn’s App Snapshot, which “yields many advantages in ease and speed of navigation. *See* EX1003, ¶85 (internal quotation marks omitted). Dr. Cockburn again asserts that a POSITA “would have sought out relevant teachings (such as those of Martyn) to predictably improve Hotelling’s interface.” *Id.*, ¶86. And again, Dr. Cockburn conflates main menu with music player, touch screen device with legacy device, and user interface design in 2001 with user interface design in 2011.

93. All of these topics are similar, but they are not the same. Dr. Cockburn, however, fuses them without further explanation. Dr. Cockburn asserts that a POSITA would have “sought out” Martyn’s teachings to “predictably improve” Hotelling. *Id.* However, this conclusion is based on the flawed premise that Hotelling’s main menu GUI and music player GUI are the same interface with the

same design parameters. As I have explained, this is not true. A POSITA would appreciate the differences in these contexts, exercising care in merging solutions from outdated devices, whose solutions themselves may also be outdated.

94. As another example, Dr. Cockburn ignores that Martyn’s “advantages” were that navigation was “far faster and easier than *conventional* navigation approaches.” EX1005, [0015], *see* EX1003, ¶85. Again, Dr. Cockburn ignores the differences between what was conventional in 2001 and what was conventional in 2011. Martyn did not predict (and could not have predicted) the ways in which touch screen features would revolutionize user interface design for mobile devices. To the contrary, Martyn itself illustrates that its “improvement” would be no such thing for a more modern touch-screen device. For example, Martyn thought it an “innovative” improvement to scroll to an app and wait for longer than a second to see a pop-up menu rather than simply opening the app: “The innovative summary window functionality can be accessed as follows: should the highlight rest on the name of an application in the App Launcher for a certain amount of time (say a 1.2 second timeout), the summary window (the ‘App Snapshot’) drops down from the highlight bar.” EX1005, [0017] (emphasis added). Hotelling however benefited from a full-screen touch-screen input that allowed the selection of an app to bring up a user interface in which the corresponding app options were literally at a user’s fingertips. *See* EX1004, FIGs. 9-13.

95. For the reasons above, each of Dr. Cockburn's reason that a POSITA would look to combine the cited references are flawed. Dr. Cockburn overly simplifies what is a complex, nuanced field of user interface design for mobile devices. The design contexts of Hotelling and Martyn are different. A POSITA would have understood this and would not have assumed that the dated solution from Martyn would still be relevant in 2011.

X. GROUNDS 1 AND 2 ALSO FAIL BECAUSE, EVEN ACCEPTING THE PROPOSED COMBINATION, THAT COMBINATION DOES NOT MEET ALL ELEMENTS OF THE CHALLENGED CLAIMS

96. Even setting aside Dr. Cockburn's flawed reasons to combine Hotelling and Martyn, his proposed combination still fails to show all claimed aspects of the '212 patent. From the perspective of a POSITA, neither Hotelling, Martyn, nor a POSITA's background knowledge teach Element [1c] of the '212 Patent: outputting "feedback that is perceptible to touch" after both a first and second duration have elapsed.

A. Limitations [1.c], [16.h], [84.d]: "[when] / [based on] a second duration of the touch . . . [is] / [being] detected . . . output feedback that is perceptible by touch"

97. None of the cited references disclose "[when] / [based on] a second duration of the touch, that is different than the first duration of the touch, being applied to the touch screen [is] / [being] detected in connection with the indicia after the first duration is detected without the ceasing, output feedback that is perceptible

by touch, utilizing the actuator,” as required by Limitations [1.c], [16.h], and [84.d]. The Petition advances two alternative disclosures of these limitations in the cited prior art: (1) the “haptics” of Hotelling, or (2) the “haptics” of Hotelling in view of the knowledge of a POSITA. EX1003, ¶¶125-130. Neither alternative is sufficient.

1. First Theory

98. Dr. Cockburn asserts: “Specifically, Hotelling explains that ‘tactile...feedback can be used in conjunction with an input event including touch events’—e.g., short and long (second) duration touches on ‘a virtual button.’ *Id.*” EX1003, ¶125. To be clear, the quoted portion of the passage is an accurate (partial) quote from paragraph [0163] of Hotelling, but the “e.g.” is neither found in nor suggested by Hotelling. To the contrary, as discussed in detail below, Hotelling discloses haptic feedback to “make[] the virtual UI feel more like a physical UI” by “provid[ing] force feedback in the form of vibration” “in response to touching a virtual button on the GUI.” EX1004, [0161], [0163]; *see also id.*, [0067]-[0070]. Hotelling is clear that this is based on a ***pressure*** (not time) threshold that is independent of the duration of the touch. *See e.g., id.*, [0163] (“The force threshold used may be similar to the threshold used to determine whether a touch is a light or hard touch.”), (“The feedback may provide information so that the user knows that they actually implemented an input (simulates the audio and tactile feel of a button or switch).”).

99. Hotelling explains that the “haptics mechanism . . . enhances the users experience and ***makes the virtual UI feel more like a physical UI.***” EX1004, [0161] (emphasis added); *see also id.*, [0067]-[0070] (describing the elimination of physical buttons and replacement with “virtual controls that mimic the physical controls are displayed in the control region 154.”). Hotelling explains further:

For example, when a certain force threshold is reached, the audio feedback device may create a “click” on the press and a “clock” on the release. ***The force threshold used may be similar to the threshold used to determine whether a touch is a light or hard touch. The “click” and “clock” may be used to simulate a button click when a hard touch is made.***

EX1004, [0163].

100. Based on the above disclosure, a POSITA would have understood Hotelling as teaching “outputting tactile feedback” with a haptics mechanism in order to ***mimic physical buttons*** through haptic feedback used in conjunction with a “clicking” sound upon a button press and a “clocking” sound upon release. This is unsurprising as one of Hotelling’s goals is to merge the various separate legacy devices that featured physical buttons. Users of the Hotelling device would be accustomed to such devices and could be confused by new, virtual buttons that do not look and feel like the old, physical buttons they are used to. The device “clicks” and vibrates when a button touch passes a certain ***force*** threshold (i.e., when a user pushes a virtual button), outputting both audio and haptic feedback to simulate a

physical button. Thus, users are able to easily understand that the GUI element they are selecting is a button like the physical buttons of legacy devices.

101. Providing haptic feedback in response to exceeding a *pressure* threshold or upon a release of a touch does not meet the plain language of Limitations [1.c], [16.h], and [84.d], which require providing haptic feedback “[when] / [based on] a second duration of the touch.” Neither a pressure threshold nor a release event can meet the plain claim language which requires providing haptic feedback “[when] / [based on]” a touch is held for a period of time.

102. A POSITA also would not understand Hotelling to imply such functionality. Today, multi-function buttons are commonly used in touch screen user interfaces. To say, however, that it was obvious in 2011 that commercial user interfaces would develop in that direction is the influence of hindsight. It is possible that users would reject multi-function buttons as too complicated. It is also possible that users would simply prefer other options. It depends very very much on the context when multifunction buttons makes sense.

103. More broadly, Dr. Cockburn’s analysis overly simplifies the use of haptic feedback in user interface design. Haptic feedback is merely a tool in the user interface designer’s toolbox. The designer must still make strategic decisions about when and how to utilize haptic feedback in order to augment the user interface without overly confusing the user. This point is similar to my earlier point regarding

menus, generally. *See supra*, ¶74. Similar to the different design strategies employed with respect to menus in Hotelling and Martyn, Hotelling and Limitation [1c] employ different design strategies with respect to haptic feedback. Dr. Cockburn downplays the fact that user interface designers make deliberate, considered choices with respect to how haptic feedback is used by their interfaces.

104. For these reasons, Dr. Cockburn’s reliance on Hotelling is misguided.

2. *Second Theory*

105. Alternatively, Dr. Cockburn asserts that a POSITA would have simply “known that tactile feedback was a well-known solution to the longstanding problem in mobile device computing of signaling significant events and/or prompting user interaction.” EX1003, ¶127. That, Dr. Cockburn believes, would have “motivated” the POSITA to “us[e] tactile feedback in response to a detected touch gesture,” such as the touch described in claim 1 of the ’212 Patent. *Id.*, ¶130. This analysis is too generic to reflect a POSITA’s design decisions.

106. Contrary to what is asserted in this portion of Dr. Cockburn’s analysis, a POSITA would understand that *when* to alert the user is just as important as *how*. Dr. Cockburn does not explain how or why a POSITA would read Hotelling to suggest the use of feedback as he proposes. As has already been discussed, user interface design involves careful, deliberate choices about when, how, and why to use tools, such as haptic feedback. Suggesting, as Dr. Cockburn does, that haptic

feedback is some kind of universal solution to “signaling significant events” in user interface design is reductive. If haptic feedback were to be used every time a device did anything “significant,” the user interface would become a confusing, unusable, constantly-vibrating mess. Instead, user interface developers carefully decide which events should receive certain which feedback (of which haptic feedback is only one option) and when that feedback should be delivered.

107. A POSITA would not interpret Hotelling to teach tactile feedback in response to *any* possible input. Hotelling discusses tactile feedback only at a high-level. Dr. Cockburn points to Hotelling’s discussion of tactile feedback that “*can* be used in conjunction with . . . touch events” to argue that a POSITA would apply tactile feedback to any of the “various types of ‘touch events’ that are described throughout Hotelling.” EX1003, ¶126. This is far too generic to be useful to a POSITA. Dr. Cockburn essentially asserts that a POSITA would understand this sentence to mean that any use of tactile feedback in conjunction with any of the touch events described in Hotelling or elsewhere is a good idea. *See* EX1003, ¶¶126-127. However, as explained, a POSITA would understand that extending the use of tactile feedback to every touch event described in Hotelling would create a confusing mess of a user interface. Hotelling does not teach this and a POSITA would understand that it is a poor design choice.

108. I do not need to take Dr. Cockburn's assertion to its logical ends, however, to demonstrate the flaws in his analysis. As an example of how a POSITA might apply Hotelling's teachings, Dr. Cockburn asserts:

A POSITA considering the Hotelling-Martyn-Cho combination would have been recognized the long-touch gesture as a significant event and, therefore, would have been motivated to output responsive tactile feedback: [1] to confirm receipt of the gesture; [2] to signal the onset of a contextual menu called by the gesture; and [3] to prompt the user to interact with the called contextual menu.

EX1003, ¶127. In this excerpt, Dr. Cockburn asserts that a POSITA would be motivated to design a device that outputs tactile feedback three times in the span of what can be no more than a few seconds to indicate three separate events. It does not take a POSITA to understand that this would cause information overload for a user, although a POSITA would certainly understand the same. A POSITA would understand that feedback, tactile or otherwise, must be used judiciously to avoid overwhelming and confusing the user. Even in the example provided by Dr. Cockburn, who presumably chose an example that is more favorable to his analysis than others, his application of Hotelling makes no sense. It goes against basic user interface design fundamentals.

109. Additionally, Dr. Cockburn's proposed modification discards Hotelling's explicit teachings. Figures 27A-27E, the music player mode touch vocabulary that Dr. Cockburn cites for the rest of the claim limitations, references the same "tactile click" discussed above. EX1004, FIGS. 27A-E. The "click" is

independent of the duration of the touch, as the chart lists the same “click” for both “short” and “long” touches. *See e.g., id.*, FIGS. 27A. Thus, the reference to “tactile click” in Figures 27A-27E is consistent with the explicit disclosure of Hotelling discussed above of using haptics to simulate pressing of physical keys rather than feedback for any “touch event.”

110. Dr. Cockburn has not shown that Limitation [1c] is disclosed by Hotelling or explained why a POSITA would modify Hotelling to fit the claim limitation. Further, Dr. Cockburn has not pointed to anything in Hotelling or otherwise that amounts to more than the mere possibility of using tactile feedback with touch screen devices. This is too generic to predictably guide a POSITA to recreate the functionality of Limitation [1c].

XI. CONCLUSION

111. 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: October 22, 2024



Rajeev Surati, Ph.D.