

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Michael S Smith
U.S. Patent No.: 10,656,758 Attorney Docket No. 50095-0168IP1
Issue Date: May 19, 2020
Appl. Serial No.: 16/664,780
Filing Date: October 25, 2019
Title: GESTURE-EQUIPPED TOUCH SCREEN SYSTEM,
METHOD, AND COMPUTER PROGRAM PRODUCT

SUPPLEMENTAL DECLARATION OF DR. ANDREW COCKBURN

TABLE OF CONTENTS

I.	INTRODUCTION	23
II.	LEGAL PRINCIPLES.....	25
	A. Person of Ordinary Skill in the Art	25
	B. Obviousness.....	26
	C. Claim Interpretation	28
III.	ANALYSIS.....	28
	A. A POSITA Would Have Had Multiple Reasons to Combine Chaudhri, Martyn, and Griffin to Provide a Contextual Sub- Menu When the Duration of the User’s Touch Input Surpasses a First Time Threshold.....	29
	B. Dr. Madisetti’s Alternative Versions of the Chaudhri-Martyn Combination Are Not Superior to the Touch-and-Hold	36
	C. Grounds 1 & 2 Each Render Obvious the Claim Features of Detecting a Touch Input on an Icon and Moving the Icon	45
	1. First Application Icon Mapping.....	47
	2. Continuous Touch	53
	D. The Challenged Dependent Claims Are Obvious	59
	1. Claims 21, 22	59
	2. Claim 47	60
	3. Claim 57	62
IV.	CONCLUSION.....	64

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1.pre]	An electronic device, comprising:
[1.a]	a display;
[1.b]	a touch-sensitive surface;
[1.c]	one or more processors;
[1.d]	memory; and
[1.e]	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:
[1.f]	displaying, on the display, an application launching user interface that includes a plurality of application icons for launching corresponding applications;
[1.g]	while displaying the application launching user interface, detecting a first single-finger touch input that includes detecting the first single-finger touch input at a location on the touch-sensitive surface that corresponds to a first application icon of the plurality of application icons, wherein the first application icon is for launching a first application that is associated with one or more corresponding action options; and

[1.h]	in response to detecting the first single-finger touch input, determining a response to the first single-finger touch input based on evaluating the first single-finger touch input against at least one of a plurality of criteria, including evaluating a duration of the first single-finger touch input against at least one of: one or more application-launch criteria, one or more action-option-display criteria, or one or more operation criteria, and further including evaluating a movement of the first single-finger touch input against one or more movement criteria, for:
[1.i]	in accordance with a determination that the first single-finger touch input meets the one or more application-launch criteria that is met when the duration of the first single-finger touch input is evaluated to be less than a first time threshold, launching the first application,
[1.j]	in accordance with a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application,
[1.k]	in accordance with a determination that the first single-finger touch input meets the one or more operation criteria that is met when the duration of the first single-finger touch input is evaluated to be greater than a second time threshold that is greater than the first time threshold, performing an operation in connection with the first application icon, and
[1.l]	in accordance with a determination that the first single-finger touch input meets the one or more movement criteria, moving the first application icon in a foreground virtual display layer so that the first application icon appears to float above a background virtual display layer.

Claim 9	
[9]	The electronic device of claim 1, wherein the one or more programs include instructions for, in response to determining that the one or more action-option-display criteria has been met, generating a first tactile output that is indicative of a satisfaction of the one or more action-option-display criteria.

Claim 12	
[12]	The electronic device of claim 1, wherein displaying the one or more action option objects includes displaying the one or more action option objects in a menu that includes a plurality of action option objects.

Claim 19	
[19]	The electronic device of claim 1, wherein: the first application is a content creation application; the one or more action option objects include a respective action option object that is associated with creating a new object; and wherein the one or more programs include instructions for: detecting selection of the respective action option object, and, in response to detecting selection of the respective action option object, creating the new object.

Claim 20	
[20]	The electronic device of claim 19, wherein the content creation application includes an email application, and the new object includes a new email message.

Claim 21	
[21]	The electronic device of claim 1, wherein: the first application is a content creation application; the one or more action option objects include a respective action option object that is associated with displaying at least a portion of previously created content; and wherein the one or more programs include instructions for: detecting selection of the respective action option object, and, in response to detecting selection of the respective action option object, displaying the at least portion of previously created content.

Claim 22	
[22]	The electronic device of claim 21, wherein the content creation application includes an email application, and the displayed at least portion of previously created content includes a previous email message, for generating a reply thereto.

Claim 24	
[24]	The electronic device of claim 1, wherein the electronic device is configured such that the operation is related to a delete operation.

Claim 25	
[25]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes displaying a graphical user interface element for, in response to detecting a selection thereof, deleting the first application icon.

Claim 26	
[26]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes entering an icon reconfiguration mode in which one or more of the plurality of application icons is capable of being reorganized on the application launching user interface.

Claim 27	
[27]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes changing a mode of operation.

Claim 28	
[28]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes adding a graphical user interface element to the first application icon.

Claim 29	
[29]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes moving the first application icon.

Claim 30

[30]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes at least partially rotating the first application icon.
------	--

Claim 31

[31]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes animating the first application icon.
------	--

Claim 32

[32]	The electronic device of claim 1, wherein the electronic device is configured such that the operation includes cyclically animating the first application icon.
------	---

Claim 35

[35]	The electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, multiple of the plurality of application icons is moved in the foreground virtual display layer so that the multiple of the plurality of application icons appear to float above the background virtual display layer.
------	--

Claim 36	
[36]	The electronic device of claim 1, wherein the electronic device is configured such that the one or more action option objects associated with the first application is displayed along with the first application icon.

Claim 37	
[37]	The electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of a different user interface is displayed on at least a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface.

Claim 38	
[38]	The electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of a different user interface is displayed on at least a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface immediately prior to detecting the first single-finger touch input.

Claim 39	
[39]	The electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of the application launching user interface is changed to display at least a portion of a different user interface on a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface immediately prior to detecting the first single-finger touch input.

Claim 40	
[40]	The electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of the application launching user interface is ceased to be displayed.

Claim 41	
[41]	The electronic device of claim 1, wherein the electronic device is configured such that the floating is a function of the movement of the first application icon.

Claim 42	
[42]	The electronic device of claim 1, wherein the electronic device is configured such that an action is performed utilizing the first application in response to detection of a selection of at least one of the one or more action option objects.

Claim 43	
[43]	The electronic device of claim 1, wherein the electronic device is configured such that an action is performed utilizing another application other than the first application in response to detection of a selection of at least one of the one or more action option objects.

Claim 47	
[47]	The electronic device of claim 1, wherein the electronic device is configured such that, during detection of the first single-finger touch input and prior to displaying the one or more action option objects, a user interface element is displayed to appear at least partially behind the first application icon.

Claim 57	
[57]	The electronic device of claim 1, wherein the electronic device is configured such that at least one of the one or more action option objects includes content that is dynamically generated.

Claim 58	
[58]	The electronic device of claim 1, wherein the electronic device is configured such that at least one of the one or more action option objects includes content that is dynamically generated based on historical usage.

Claim 59	
[59]	The electronic device of claim 1, wherein the electronic device is configured such that at least one of the one or more action option objects includes content that is dynamically generated based on historical usage of the first application.

Claim 63	
[63.pre]	A method, comprising:
[63.a]	at an electronic device including a display and a touch-sensitive surface:
[63.b]	displaying, on the display, an application launching user interface that includes a plurality of application icons for launching corresponding applications;
[63.c]	with the application launching user interface displayed, detecting a first single-finger touch input that includes detecting the first single-finger touch input at a location on the touch-sensitive surface that corresponds to a first application icon of the plurality of application icons, wherein the first application icon is for launching a first application that is associated with one or more corresponding action options; and

[63.d]	in response to detecting the first single-finger touch input, determining a response to the first single-finger touch input based on evaluating the first single-finger touch input against at least one of a plurality of criteria, including evaluating a duration of the first single-finger touch input against at least one of: one or more application-launch criteria, one or more action-option-display criteria, or one or more operation criteria, and further including evaluating a movement of the first single-finger touch input against one or more movement criteria, for:
[63.e]	based on a determination that the first single-finger touch input meets the one or more application-launch criteria that is met based on the duration of the first single-finger touch input being evaluated to be less than a first time threshold, launching the first application,
[63.f]	based on a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met based on the duration of the first single-finger touch input being evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application,
[63.g]	based on a determination that the first single-finger touch input meets the one or more operation criteria that is met based on the duration of the first single-finger touch input being evaluated to be greater than a second time threshold that is greater than the first time threshold, performing an operation on the first application icon, and
[63.h]	based on a determination that the first single-finger touch input meets the one or more movement criteria, moving the first application icon in a foreground virtual display layer so that the first application icon appears to float above a background virtual display layer.

Claim 71	
[71]	The method of claim 63, and further comprising: in response to determining that the one or more action-option-display criteria has been met, generating a first tactile output that is indicative of a satisfaction of the one or more action-option-display criteria.

Claim 74	
[74]	The method of claim 63, wherein displaying the one or more action option objects includes displaying the one or more action option objects in a menu that includes a plurality of action option objects.

Claim 81	
[81]	The method of claim 63, wherein: the first application is a content creation application; the one or more action option objects include a respective action option object that is associated with creating a new object; and and further comprising: detecting selection of the respective action option object, and, in response to detecting selection of the respective action option object, creating the new object.

Claim 82	
[82]	The method of claim 81, wherein the content creation application includes an email application, and the new object includes a new email message.

Claim 83	
[83]	The method of claim 63, wherein: the first application is a content creation application; the one or more action option objects include a respective action option object that is associated with displaying at least a portion of previously created content; and and further comprising: detecting selection of the respective action option object, and, in response to detecting selection of the respective action option object, displaying the at least portion of previously created content.

Claim 84	
[84]	The method of claim 83, wherein the content creation application includes an email application, and the displayed at least portion of previously created content includes a previous email message, for generating a reply thereto.

Claim 86	
[86]	The method of claim 63, wherein the operation is related to a delete operation.

Claim 87	
[87]	The method of claim 63, wherein the operation includes displaying a graphical user interface element for, in response to detecting a selection thereof, deleting the first application icon.

Claim 88	
[88]	The method of claim 63, wherein the operation includes entering an icon reconfiguration mode in which one or more of the plurality of application icons is capable of being reorganized on the application launching user interface.

Claim 89	
[89]	The method of claim 63, wherein the operation includes changing a mode of operation.

Claim 90	
[90]	The method of claim 63, wherein the operation includes adding a graphical user interface element to the first application icon.

Claim 91	
[91]	The method of claim 63, wherein the operation includes moving the first application icon.

Claim 92

[92]	The method of claim 63, wherein the operation includes at least partially rotating the first application icon.
------	--

Claim 93

[93]	The method of claim 63, wherein the operation includes animating the first application icon.
------	--

Claim 94

[94]	The method of claim 63, wherein the operation includes cyclically animating the first application icon.
------	---

Claim 97

[97]	The method of claim 63, wherein, based on the determination that the first single-finger touch input meets the one or more movement criteria, multiple of the plurality of application icons is moved in the foreground virtual display layer so that the multiple of the plurality of application icons appear to float above the background virtual display layer.
------	--

Claim 98

[98]	The method of claim 63, wherein the one or more action option objects associated with the first application is displayed along with the first application icon.
------	---

Claim 99	
[99]	The method of claim 63, wherein, based on the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of a different user interface is displayed on at least a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface.

Claim 100	
[100]	The method of claim 63, wherein, based on the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of a different user interface is displayed on at least a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface immediately prior to detecting the first single-finger touch input.

Claim 101	
[101]	The method of claim 63, wherein, based on the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of the application launching user interface is changed to display at least a portion of a different user interface on a portion of the display that was previously occupied by at least multiple of the plurality of application icons in the application launching user interface immediately prior to detecting the first single-finger touch input.

Claim 102	
[102]	The method of claim 63, wherein, based on the determination that the first single-finger touch input meets the one or more movement criteria, at least a portion of the application launching user interface is ceased to be displayed.

Claim 103	
[103]	The method of claim 63, wherein the floating is a function of the movement of the first application icon.

Claim 104	
[104]	The method of claim 63, wherein an action is performed utilizing the first application in response to detection of a selection of at least one of the one or more action option objects.

Claim 105	
[105]	The method of claim 63, wherein an action is performed utilizing another application other than the first application in response to detection of a selection of at least one of the one or more action option objects.

Claim 109	
[109]	The method of claim 63, wherein, with the first single-finger touch input detected and prior to displaying the one or more action option objects, a user interface element is displayed to appear at least partially behind the first application icon.

Claim 119	
[119]	The method of claim 63, wherein at least one of the one or more action option objects includes content that is dynamically generated.

Claim 120	
[120]	The method of claim 63, wherein at least one of the one or more action option objects includes content that is dynamically generated based on historical usage.

Claim 121	
[121]	The method of claim 63, wherein at least one of the one or more action option objects includes content that is dynamically generated based on historical usage of the first application.

Claim 124	
[124.pre]	A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions which, when executed by an electronic device with a display and a touch-sensitive surface, cause the electronic device to:

[124.a]	display, on the display, an application launching user interface that includes a plurality of application icons for launching corresponding applications;
[124.b]	with the application launching user interface displayed, detect a first single-finger touch input that includes detecting the first single-finger touch input at a location on the touch-sensitive surface that corresponds to a first application icon of the plurality of application icons, wherein the first application icon is for launching a first application that is associated with one or more corresponding action options; and
[124.c]	in response to detecting the first single-finger touch input, determine a response to the first single-finger touch input based on evaluating the first single-finger touch input against at least one of a plurality of criteria, including evaluating a duration of the first single-finger touch input against at least one of: one or more application-launch criteria, one or more action-option-display criteria, or one or more operation criteria, and further including evaluating a movement of the first single-finger touch input against one or more movement criteria, for:
[124.d]	based on a determination that the first single-finger touch input meets the one or more application-launch criteria that is met based on the duration of the first single-finger touch input being evaluated to be less than a first time threshold, launching the first application,
[124.e]	based on a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met based on the duration of the first single-finger touch input being evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application,

[124.f]	based on a determination that the first single-finger touch input meets the one or more operation criteria that is met based on the duration of the first single-finger touch input being evaluated to be greater than a second time threshold that is greater than the first time threshold, performing an operation on the first application icon, and
[124.g]	based on a determination that the first single-finger touch input meets the one or more movement criteria, moving the first application icon in a foreground virtual display layer so that the first application icon appears to float above a background virtual display layer.

I, Andrew Cockburn, PhD, declare as follows:

I. INTRODUCTION

1. I have been retained as a technical expert by counsel on behalf of Apple Inc. (“Apple”/“Petitioner”) in Case No. IPR2024-01088 with respect to U.S. Patent U.S. Patent No. 10,656,758 (“the ’758 patent”) (EX1001).

2. I understand that this declaration (“Supplemental Declaration”) will be submitted in support of Petitioner’s Reply to Patent Owner’s Response to the Petition for *inter partes* review of the ’758 patent. This Supplemental Declaration supplements certain portions of my prior declaration in support of the Petition for this IPR Proceeding (EX1003, “Prior Declaration”). In my Prior Declaration, I address many topics, including (but not limited to) my background and qualifications, the level of skill in art, an overview of the ’758 patent, a POSITA’s background knowledge, certain legal standards explained to me by counsel, and a detailed analysis of the prior art against the claim elements of the ’758 patent.

3. I maintain all of my earlier analysis and continue to hold the opinions expressed in my Prior Declaration. This Supplemental Declaration does not replace my Prior Declaration. These declarations are complementary and should be read in tandem. Accordingly, this Supplemental Declaration does not cover every topic addressed in my Prior Declaration.

4. In forming this Supplemental Declaration, I have considered the following: the declaration of Patent Owner's expert Dr. Madisetti (EX2005), the Board's institution decision (Paper 7), my knowledge and experience in this field, including my research and work experience in the field, my experience working with others involved in the field, and the third-party documents cited throughout my analysis. *See* EX1003, ¶¶4–18 (discussing qualifications), ¶¶39–40 (discussing the level of skill for a “POSITA”), and ¶¶41–53 (discussing a POSITA's background knowledge and understanding).

5. Note that my citations to non-patent literature throughout this Supplemental Declaration reference the absolute page number added to the exhibit (as opposed to the original pagination of the document). For patent literature, I've used the column/line numbers or paragraph numbers. Also note that, unless I've indicated otherwise, all emphasis (bold/italics/underline) in any quoted text are ones I have added.

6. Although I cite and quote to selected portions of various references in this Supplemental Declaration, the reader should understand that these citations are representative examples. A POSITA would have viewed each reference in its entirety and in combination with other references. Accordingly, I intend the references identified in this Declaration to be viewed as incorporated in their entireties.

7. I am not, and never have been, an employee of Apple. I received no compensation for this Supplemental Declaration beyond my normal hourly compensation based on my time actually spent preparing this declaration. I will not receive any added compensation based on the outcome of any IPR or other proceeding involving the '758 patent.

II. LEGAL PRINCIPLES

8. In forming my analysis and conclusions expressed in this Declaration, I have applied the legal principles described in the following paragraphs, which were provided to me by counsel for the Petitioner.

A. Person of Ordinary Skill in the Art

9. I understand that the factors considered in determining the ordinary level of skill in a field of art include: the level of education and experience of persons working in the field; the types of problems encountered in the field; the teachings of the prior art regarding solutions to such problems; and the sophistication of the technology at the time of the alleged invention. I understand that a person of ordinary skill in the art ("POSITA") is not a specific real individual, but rather is a hypothetical individual having the qualities reflected by the factors above and knowledge of all relevant prior art references, including the references I cite below.

B. Obviousness

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

11. While I do not know the exact date that the alleged invention of the ’758 patent was made, I do know that the ’758 patent’s earliest claimed priority date is August 5, 2011. For purposes of my obviousness analysis, I have applied a date of August 5, 2011 as the date of the alleged invention (“Critical Date”), although in many cases the same analysis would hold true even if the date of the alleged invention were earlier or later.¹

¹ Note, however, that I do not concede that the ’758 patent is entitled to its earliest claimed priority date.

12. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason that would have prompted a POSITA to modify or supplement the single prior art reference, or to combine two or more references, in a manner that provides the elements of the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference(s) themselves, the knowledge or “common sense” of a POSITA, or from the nature of the problem to be solved, and this reason may be explicit or implicit from the prior art as a whole. I have been informed that, under the law, the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

13. I understand that an obviousness determination also requires that a POSITA would have had a reasonable expectation of success. This concept has been explained to me as relating to the relative likelihood or predictability—from the perspective of a POSITA—of successfully modifying the prior art in a manner that would meet the claimed limitations of the patent being challenged (here the ’758 patent). I understand that the expectation of success need only be “reasonable” and,

thus, does not require the absolute certainty gleaned from physically creating the proposed prior art modification.

C. Claim Interpretation

14. I understand that, for purposes of my analysis in this *inter partes* review proceeding, the terms appearing in the patent claims should be interpreted according to their “ordinary and customary meaning.” In determining the ordinary and customary meaning, I understand that the words of a claim are first given their plain meaning that those words would have had to a person of ordinary skill in the art at the time of the alleged invention. I also understand that the structure of the claims, the specification and file history may be used to better construe a claim. Moreover, I understand that even treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a person of ordinary skill in the art to a claim term.

III. ANALYSIS

15. The analysis and opinions expressed in my Prior Declaration fully explain why [1] each and every feature of the '758 patent's challenged claims is in the prior art; and [2] a POSITA would have been motivated to combine the known prior art elements in a manner that reaches the claimed invention. Patent Owner's expert, Dr. Madisetti, has considered my opinions and offered his own, some of which are inconsistent with my view. I will address certain points of dispute below.

The fact that I have not addressed all of Dr. Madiseti's opinions should not be interpreted as agreement with any of them.

A. A POSITA Would Have Had Multiple Reasons to Combine Chaudhri, Martyn, and Griffin to Provide a Contextual Sub-Menu When the Duration of the User's Touch Input Surpasses a First Time Threshold

16. As my Prior Declaration explains:

[t]he Chaudhri-Martyn-Griffin combination represents the predictable result of applying in the context of Chaudhri's user interface for a touch screen device: [a] Martyn's teachings about providing a contextual shortcut sub-menu of common application functions in response to a press-and-hold user input; and [b] Griffin's teachings about evaluating a user's touch input against predetermined threshold criteria.

EX1003, ¶61; *see also* ¶¶62-78.

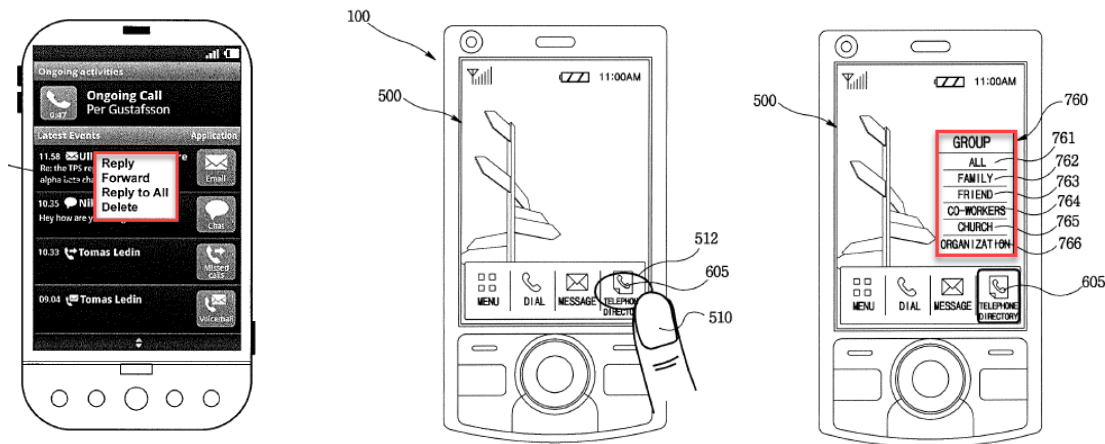
17. My Prior Declaration relies on the combined teachings of Chaudhri and Martyn for providing a contextual sub-menu—which Martyn calls the App Snapshot—when the duration of the user's touch input surpasses a first time threshold. EX1003, ¶¶63-69. I explained why a POSITA would have been motivated to apply Martyn's App Snapshot teachings in the context of Chaudhri's user interface for a touch screen device:

Multiple reasons would have motivated a POSITA to integrate Martyn's teachings about providing a contextual shortcut sub-menu in the form of a "summary window" of common application functions (an "App Snapshot") in response to a press-and-hold user input with Chaudhri's disclosure of a touch screen mobile device with a user interface for displaying and reorganizing a menu of selectable application icons. The POSITA would have understood that the analog to Martyn's press-and-hold input on a touch screen like Chaudhri's is

the well-known touch-and-hold gesture.[] Accordingly, the predictable result of the Chaudhri-Martyn combination is a mobile device featuring an application menu user interface that displays a contextual sub-menu, such as Martyn discloses, in response to a touch-and-hold gesture applied to Chaudhri's touch screen.

EX1003, ¶63; *see also* ¶¶64–69 (providing four independent motivation rationales).

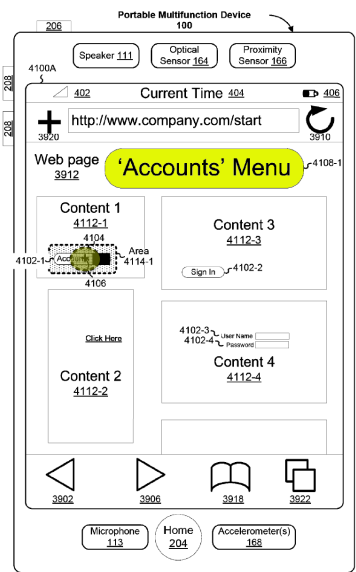
18. Dr. Madiseti's general position is that the motivation rationales provided in my Prior Declaration only account for "providing a sub-menu of selectable options corresponding to commonly used application functions" and "not that a specific method for calling those sub-menus was ubiquitous." EX2005, ¶98; *see also* ¶¶95-112. This is not true. For example, the analyses outlining the Chaudhri-Martin-Griffin combination and detailing the multiple motivation-to-combine rationales in my Prior Declaration cross-reference an earlier discussion I provided regarding a POSITA's background knowledge on topics such as menus and duration-based gestures. *E.g.*, EX1003, ¶62 ("Sections VI.B.2 (menus), and VI.B.3 (duration-based touch gestures)"), ¶63 n3 ("Section VI.B.3"). In my discussion of the POSITA background knowledge, I highlighted multiple examples of touch screen mobile devices where a "long press selection" reveals "additional functionality" in the form of "a pop up control that a user may then interact with"—*e.g.*, "a pop up contextual menu that relates to an element in the display." EX1024, 1:35-65; *see also*, 4:25-5:51, 8:66-9:19, Figure 1; EX1025, [0182]–[0183], Figures 12c-12d; EX1003, ¶¶47-48.



EX1024, Figure 1 (left, annotated); EX1025, Figures 12c-12d (right, annotated).

19. These examples, and several others from documents I considered when preparing my Prior Declaration (see table below), are consistent with my understanding that a POSITA in 2011 would have known that using a touch-and-hold gesture on user interface icons to invoke menus and control options was commonplace. Pairing helpful user interface elements like contextual menus and control options with a conventional, intuitive, and easy-to-perform gesture like a touch-and-hold was an effective technique that, accordingly, grew in popularity amongst those of skill in the art. Indeed, using a touch-and-hold gesture to activate a shortcut menu was a routine and conventional part of the most widely used operating systems at the time, such as Microsoft Windows 7: “Press and hold does the same thing as right-clicking an item. To perform the action, touch the screen where you want to right-click, hold until a complete circle appears, and then lift your finger. The shortcut menu appears after you lift your finger.” EX1071, p.1. Thus,

as explained in my Prior Declaration, a POSITA would have understood that Martyn's App Snapshot and press-and-hold user input were "well-known and beneficial" teachings that would "bolster" Chaudhri without disturbing the "fundamental aspects of Chaudhri's disclosure. EX1003, ¶¶61-63.

Document	Description
EX1023 US 7,479,949 Inventors: Jobs, <i>et al.</i> Filed: Apr. 11, 2008	72:38–42: <u>Information associated with the chosen user input element is displayed over the displayed content (e.g., Accounts Menu 4108-1, FIG. 41A).</u> In some embodiments, the displayed information associated with the chosen user input element comprises a description of the chosen user input element. 72:56–62: <u>In some embodiments, the information associated with the chosen user input element is displayed after the contact is maintained for at least a predetermined time.</u> In some embodiments, the displayed information associated with the chosen user input element is removed if the contact with the touch screen is maintained for greater than a predetermined time.  Figure 41A

Document	Description																				
EX1045 iPhone User Guide (OS 3.1 Software) Copyright 2009	Page 60: In Photos, choose a video, then <u>touch and hold the video until the Copy command appears</u>. Tap Copy. Go to Mail and create a new message. Tap to place an insertion point where you want the video, then tap the insertion point to display the edit commands and tap Paste. Page 62: <u>Touch and hold the link</u>. The address is displayed, <u>and you can choose to open the link</u> in Safari or copy the link address to the clipboard. Page 63: Save an attached video to your Camera Roll album: <u>Touch and hold the attachment, then tap Save Video</u>. If the video hasn’t been downloaded yet, tap the download notice first. Page 88: Copy a photo or video: <u>Touch and hold the attachment, then tap Copy</u>. You can paste the photo or video to a[] Mail message or another MMS message.																				
EX1037 Android Developers User Interface > Creating Menus Archived: 2011	Page 1: Contact Menu A floating list of <u>menu items</u> that appears when the user performs a <u>long-press</u> on a View. Page 4: On Android, a <u>context menu</u> is displayed when the user performs a “<u>long press</u>” (press and hold) on an item.																				
EX1031 US 2006/0197753 Inventors: Hotelling Filed: Mar. 3, 2006	Figure 27A: <table><thead><tr><th>Duration</th><th>Touchdown location (virtual button?)</th><th>Proposed Behavior</th><th>Feedback</th></tr></thead><tbody><tr><td>any</td><td>anywhere</td><td>On touch-down, do nothing. On lift-off, highlight item (if any) under touch</td><td>highlight item touched</td></tr><tr><td>short</td><td>virtual button / list item</td><td>On touch-down, do nothing. On click-down highlight item under touch (if any). On un-click, execute the item clicked</td><td>animate screen sliding RIGHT to expose new screen</td></tr><tr><td>short</td><td>PLAY /- PAUSE virtual button</td><td>On touch-down, do nothing. On click-down highlight Play/Pause icon. On un-click, PLAY/PAUSE</td><td>Change PLAY/PAUSE icon</td></tr><tr><td>LONG</td><td>virtual button</td><td>On touch-down, do nothing. On click-down highlight item under touch (if any). After delay, bring-up context sensitive menu.</td><td>Context sensitive menu pop-up</td></tr></tbody></table>	Duration	Touchdown location (virtual button?)	Proposed Behavior	Feedback	any	anywhere	On touch-down, do nothing. On lift-off, highlight item (if any) under touch	highlight item touched	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	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	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
Duration	Touchdown location (virtual button?)	Proposed Behavior	Feedback																		
any	anywhere	On touch-down, do nothing. On lift-off, highlight item (if any) under touch	highlight item touched																		
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																		
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																		
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																		

20. Another problem with Dr. Madisetti's approach is that he attempts to isolate Martyn's App Snapshot teachings from Martyn's corresponding disclosure of a press-and-hold user input as a technique for calling the App Snapshot. I did not separate these disclosures because a POSITA would not have done so. The POSITA would have instead taken Martyn's teachings as a whole, including Martyn's disclosure of calling the App Snapshot with a press-and-hold user input. It would not make sense for someone motivated to incorporate Martyn's App Snapshot with Chaudhri to completely ignore Martyn's other related teachings. Given that Martyn and Chaudhri both describe similar mobile devices like smartphones and PDAs, the POSITA would have appreciated that Martyn's teaching to employ a press-and-hold user input for calling the App Snapshot would be similarly effective in the context of Chaudhri. *See* EX1004, [0073] ("iPhone" and "PDA"); EX1005, [0002] ("smart phones" and "PDAs"). Accordingly, when I articulated multiple motivation rationales in my Prior Declaration for combining Chaudhri and Martyn, I referred not to the App Snapshot in isolation, but instead to Martyn's teachings as a whole regarding the App Snapshot, including the press-and-hold user input. *See* EX1003, ¶63 ("Multiple reasons would have motivated a POSITA to integrate Martyn's teachings about providing a contextual shortcut sub-menu in the form of a "summary window" of common application functions (an 'App Snapshot') in response to a press-and-hold user input with Chaudhri's disclosure..."), ¶¶64-69 ("ubiquitous

feature,” “improve[d]...usability,” “advantages in ease and speed of navigation,” “a natural and obvious extension of Chaudhri”).

21. I further explained in my Prior Declaration that a POSITA would have understood that a touch-and-hold gesture performed on a touch screen device is the “analog” of Martyn’s press-and-hold input. EX1003, ¶63. This fact is essentially self-evident given the similarity of the terms “touch-and-hold” and “press-and-hold.” Indeed, these terms are often used interchangeably in the technical and patent literature, and also amongst ordinarily skilled people in the field. As one example, I note that the online Android Developer’s guide in 2011 described an event listener called “onLongClick()” that applied equally to long duration inputs based on touch and physical key presses:

onLongClick()

From [View.OnLongClickListener](#). This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).

EX1061, p.1.

22. Lastly, I note that Dr. Madisetti contends that I cannot “motivate the commonality or longstanding use of” the touch-and-hold calling technique because my Prior Declaration describes a variety of different calling methods. EX2005, ¶¶100-101. This critique mischaracterizes my testimony. My Prior Declaration does not say that a touch-and-hold gesture “must be used.” EX2005, ¶100. My testimony

is that a POSITA would have been motivated to use a touch-and-hold gesture for calling Martyn's App Snapshot for all the reasons I discussed above and in my Prior Declaration.

B. Dr. Madisetti's Proposed Alternatives Are Not Superior to a Basic Touch-and-Hold for Calling Martyn's App Snapshot

23. Dr. Madisetti presents several alternative gestures for calling Martyn's App Snapshot without a touch-and-hold gesture. EX2005, ¶¶113-122, ¶¶126-135. Dr. Madisetti states or suggests that his alternatives are "superior" to a basic touch-and-hold gesture. *E.g.*, EX2005, ¶119. I disagree for the reasons provided below.

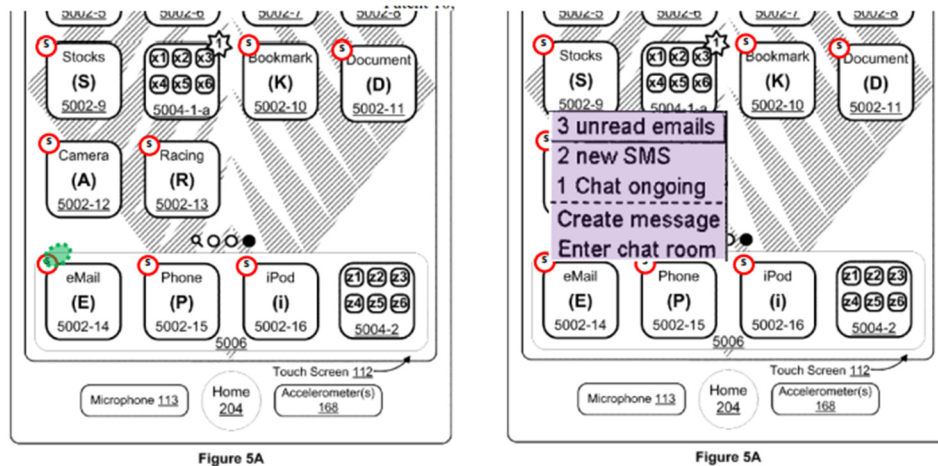
24. First, as a threshold matter, I have been informed by counsel for Petitioner Apple that a proposed combination of predictable elements need not be the best option or an improvement over other known options. It is enough for a motivation-to-combine analysis to establish that the proposed combination presents a suitable option that a POSITA would have had good reasons to pursue and from which the prior art does not teach away. Regarding "teaching away," I have been informed that a reference does not teach away if it does not criticize, discredit, or otherwise discourage investigation into the invention claimed. Based on these understandings, it is clear that Dr. Madisetti's alternatives do not detract from the motivation-to-combine analysis that I presented in my Prior Declaration, and that I discussed at length above. For example, I am not aware of any references that

discourage the well-known technique of using a touch-and-hold gesture to call a contextual menu like Maryn's App Snapshot. Moreover, I explained in my Prior Declaration and above why a POSITA would have had good reasons to employ the touch-and-hold for this purpose in the Chaudhri-Martyn-Griffin combination. Nevertheless, I believe Dr. Madisetti is wrong to suggest that his alternative command gestures are "superior" to a basic touch-and-hold on an application icon, as proposed in the Chaudhri-Martyn-Griffin combination.

25. To start, Dr. Madisetti appears to assume that using a touch-and-hold gesture for calling the App Snapshot causes a "conflict" with Chaudhri's disclosed touch-and-hold gesture for entering the reconfiguration mode. EX2005, ¶92. That assumption is wrong. As I explained in my Prior Declaration, Chaudhri discloses "a touch that exceeds the criterion of a 'predetermined period of time (e.g., 2 seconds),' " and a "tap" gesture was commonly associated with duration thresholds on the order of "less than 500 milliseconds." EX1003, ¶¶75-77 (quoting EX1004, [0217]; EX1035, 11:19-24). As such, a "POSITA would have known to implement the duration criterion for the corresponding touch-and-hold gesture [that calls the App Snapshot] as being greater than the first time threshold of the tap [e.g., 500 milliseconds], while also less than a second time threshold beyond which the interface transitions out of the 'normal' mode and into the 'reconfiguration' mode [e.g., 2 seconds]." EX1003, ¶77; *see also* ¶¶49-53 (describing the advanced state of

the art on “duration-based touch gestures”). Griffin provides helpful implementation details for “evaluat[ing] a duration parameter of the user’s touch against certain predetermined threshold criteria,” and “[e]stablishing such criteria would have been simple for a POSITA exercising common sense and ordinary creativity in view of the problem to be solved.” EX1003, ¶76.

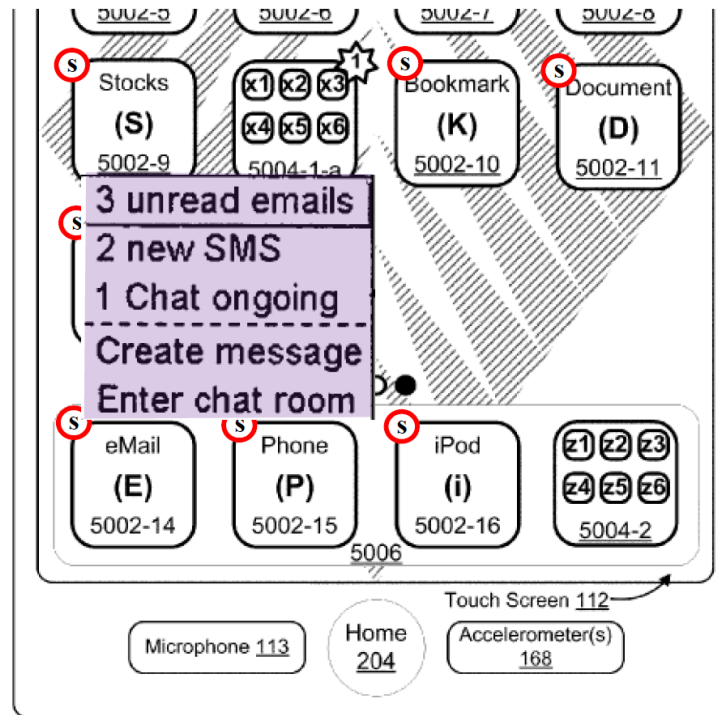
26. Dr. Madisetti derives one of his proposed alternative techniques by integrating teachings from Chaudhri’s reconfiguration mode with Chaudhri’s normal mode. *See* EX2005, ¶¶113-122. According to Dr. Madisetti, Chaudhri discloses a technique for calling a context menu when the device is in the reconfiguration mode, and a POSITA would have implemented Martyn’s App Snapshot with a modified version of this technique:



EX2005, ¶¶113-117 (Dr. Madisetti’s visual aids).

27. Dr. Madisetti claims this alternative gesture “is superior to Martyn’s calling methods.” EX2005, ¶119. I disagree. For example, a POSITA would not

have preferred Dr. Madisetti's alternative because it creates a crowded user interface and requires the user to precisely tap a small "object modification target" (red circles below) placed "in the upper lefthand corner on [the respective application] icons":



EX2005, ¶¶113-117 (Dr. Madisetti's visual aid). From a POSITA's perspective, this crowded interface problem is particularly concerning in the context of mobile touch screen devices like those Chaudhri describes. For example, as Chaudhri explains, "finger-based contacts and gestures...can be less precise than stylus-based input due to the larger area of contact of a finger on the touch screen." EX1004, [0091].

28. Chaudhri's accounts for this issue when the device is in reconfiguration mode (e.g., where the "object removal badges" are presented) by "not activat[ing] the application" in response to an "action icon selection input while the device is in

user interface reconfiguration mode.” EX1004, [0262]-[0263], [0307], [0312]. In other words, in the reconfiguration mode, the consequences of a misaimed finger that tries to tap an object removal badge but instead contacts the application icon are minimal because the tap will not launch the application. In this case, the user can immediately try again.

29. By contrast, in the normal mode, the same misaimed tap would launch the application instead of displaying the App Snapshot, taking the user somewhere in the interface that they did not want to go. *See* EX1004, [0015]-[0020], [0216]-[0217], [0280] (“applications are only activated when the device is in a normal operation mode”), [0301], [0310-0311], FIG. 7C. Accordingly, a POSITA would have understood that adding numerous selectable badges to Chaudhri’s user interface in the normal mode would decrease usability. Placing a selectable badge for calling the App Snapshot so close to each of the selectable application icons would make erroneous user inputs much more likely and, as a consequence, increase user frustration. *See* EX1054, p.1 (“Even better than a good error message is a careful design which prevents a problem from occurring in the first place.”). Comparatively, a simple touch-and-hold gesture anywhere on the larger application icon would not require the high levels of precision by the user that Dr. Madisetti’s proposed alternative does. In short, a POSITA would have understood that using multiple duration-based gestures on the same icon in Chaudhri’s “normal” mode

would be easier and more reliable for users than Dr. Madisetti's alleged alternative of using the same gesture on multiple different densely packed icons.

30. Dr. Madisetti also states that “a tap on a target region is faster and simpler than the two-part (first navigate a highlight to an application and then long press a select function button) calling method Petitioner proposes selecting from Martyn or the long touch Petitioner asserts is the ‘analog’ to Martyn’s actual disclosure.” EX2005, ¶121. However, Dr. Madisetti appears to misunderstand the combination. The combination outlined in my Prior Declaration employs a simple touch-and-hold gesture on an application icon of Chaudhri’s touch screen user interface, which a POSITA would have understood does not require an initial “navigat[ion]” step. EX1003, ¶63. As Chaudhri explains, the user can interact with application icons by providing a touch input at the location of the icon on the screen. There is no need in Chaudhri for the initial navigation step Dr. Madisetti references, and a POSITA motivated to incorporate Martyn’s App Snapshot in Chaudhri would have recognized as much (i.e., that a direct touch-and-hold is sufficient).

31. I turn now to Dr. Madisetti’s other alternative techniques for calling the App Snapshot, which he argues are consistent with Chaudhri and Martyn’s teachings in a way that avoids the alleged “conflict.” EX2005, ¶¶126-135. But as I explained above and in my Prior Declaration, a POSITA would have recognized that using two touch-and-hold gestures of different durations on the same application icon does not

create a conflict. EX1003, ¶¶75-77. Additionally, none of Dr. Madisetti's alternative techniques would have led the POSITA away from the time-tested solution of using a touch-and-hold to call a contextual menu.

32. Dr. Madisetti argues that Chaudhri's disclosed "physical or virtual wheel" provides the appropriate interface for implementing Martyn's press-and-hold input. EX2005, ¶¶127-135. Although Chaudhri teaches a "physical or virtual wheel," that is not Chaudhri's primary mechanism for interfacing with application icons. EX1004, [0093]. The "wheel" option is noted in a single paragraph. By contrast, Chaudhri includes numerous figures and paragraphs of description about interfacing with the application icons through touch contacts at locations on the touch screen where the application icons are displayed. *E.g.*, [0008] ("the user interacts with the GUI primarily through finger contacts and gestures on the touch-sensitive surface"), [0091] ("the user interface is designed to work primarily with finger-based contacts and gestures"), [0217] ("the device detects a contact 5008 with the solitaire application icon 5002-4"), Figures 5A-5PPP. To the extent a POSITA understood that the physical/virtual wheel Chaudhri discloses could be used to call Martyn's App Snapshot, it would have been even more obvious to use a touch-and-hold contact gesture on the application icon. The latter is far more aligned with Chaudhri's disclosure as a whole, which repeatedly discloses duration-based gestures. *E.g.*, [0102] (disclosing a "tap"), [0174] (similar), [0217] ("detects contact

5008 with the solitaire application icon 5002-4 for more than a predetermined amount of time”), [0220] (disclosing movement of the contact to the edge of the touch screen and held there “for more than a predetermined time threshold (e.g., 1 second)”), [0283] (“the device detects a pause of the contact 5040...for more than a predetermined amount of time (e.g., 0.5 seconds, 1 second, 1.5 seconds or any reasonable period of time)”), [0346] (“the automatic reconfiguration criteria are met when a predetermined period of time (e.g., 0.5 seconds, 1 second, 2 seconds, etc.) has passed (1018) since the input (or a beginning of the input) was detected”).

33. As I have explained, even if Martyn describes what Dr. Madisetti calls a “two-step process” that involves an initial navigation step before the press-and-hold, a POSITA in 2011 would have realized that the navigation step is unnecessary when employing Chaudhri’s preferred approach of interfacing with the application icons directly by touch. EX2005, ¶128. In other words, the POSITA would have adapted Martyn’s press-and-hold teaching to the context of Chaudhri and, thus, arrived at the well-known solution of a touch-and-hold on the application icon for calling the App Snapshot. Not only is a simple touch-and-hold gesture more aligned with Chaudhri, but it is also easier and more efficient than a two-step process that requires the user to perform an initial navigation action via a physical/virtual wheel, which would have been clear to the POSITA. Additionally, and consistent with the technical background discussion in my Prior Declaration, a POSITA in 2011 would

have appreciated that providing an interface that facilitates direct contacts with graphical elements like icons on a touch screen was a well-known and often preferred manner of interfacing with the kind of mobile devices disclosed in Chaudhri (EX1004, [0073] (“iPhone® and iPod Touch® devices”). *See* EX1003, ¶¶41-53.

34. Dr. Madisetti also references Martyn’s “other methods of calling up the App Snapshot,” such as [i] scrolling a highlight to the application icon and allowing the highlight to “rest” on the icon for a predetermined amount of time; and [ii] voice activation. EX2005, ¶¶136-141. However, as I discussed above, a POSITA would have appreciated that the most natural way to interface with touch screen devices like the ones disclosed in Chaudhri is through contacts on the application icons. So, even if Martyn teaches various other ways of calling the App Snapshot, the notion of a touch-and-hold, as an analog to the “press and hold” that Martyn discloses—would have stood out to the POSITA as a preferred technique.

35. Moreover, neither of Dr. Madisetti’s alternatives is categorically superior to a basic touch-and-hold gesture on an application icon. For example, as I have explained, a POSITA would have recognized that the touch-and-hold gesture would be easier and more efficient for the user to execute than a two-part process requiring the user to first navigate to an application icon using a physical or virtual wheel. As for the notion of using voice commands, a POSITA would not have

abandoned touch gestures altogether in the context of reference like Chaudhri that is focused on touch screen technology. To a POSITA, voice commands would have been an option for supplementing touch gestures, but they would not have been seen as a superior substitute. For example, the POSITA would have appreciated that voice commands would not be effective in noisy environments and that users may prefer not to use voice commands in settings where talking is prohibited or discouraged.

C. Grounds 1 & 2 Each Render Obvious the Claim Features of Detecting a Touch Input on an Icon and Moving the Icon

36. Elements [1.g] and [1.l] require “detecting a first single-finger touch input...at a location...that corresponds to a first application icon,” (Element [1.g]) and, when it is “determin[ed] that the first single-finger touch input meets the one or more movement criteria, moving the first application icon, (Element [1.l]). As explained throughout my Prior Declaration, concepts like detecting a touch input on an icon and moving an icon were exceedingly well known by the time of the ’758 patent’s earliest claimed priority date in 2011. Beyond high-level descriptions in technical papers and patent literature, this technology was implemented in real-world commercial products, including Apple’s iPhone. As indicated in my Prior Declaration, at least by 2009, years before the ’758 patent, the iPhone (OS 3.1 Software) included a feature that allowed users to “customize the layout of icons on

the Home screen” by performing a “touch and hold” on “any Home screen icon until the icons begin to wiggle” and “arrang[ing] the icons by dragging them.” EX1045, p.25; EX1003, ¶¶101, ¶41. The Chaudhri reference (EX1004) describes this same technology and more.

37. Accordingly, I provided the following general descriptions of Chaudhri in my Prior Declaration (EX1003, ¶¶55-58):

- When the device detects a touch input associated with one of the application icons, it determines a responsive action based on certain gesture characteristics of the touch. EX1004 (Chaudhri), [0102], [0174], [0216-0217], [0276], [0280], [0302], [0310-0311], Figures 4A, 5A-B, 7A-B.
- In the “normal” mode of operation for the application menu interface, Chaudhri’s device launches an application when the matching icon is selected with a “tap” gesture (also colloquially called a “short touch”). EX1004 (Chaudhri), [0101-0102], [0174], [0217], [0276], [0280], [0289], [0302], Figure 5A.
- In addition to the “tap” gesture, Chaudhri’s device also recognizes and responds to a touch-and-hold (also colloquially called a “long touch”) at an application icon. EX1004 (Chaudhri), [0217-0218], [0313], Figures 5B, 7A, 7B.
- While the interface for the application menu is in the “reconfiguration mode,” Chaudhri’s device recognizes yet another gesture—a drag—as a command to move an application icon. EX1004 (Chaudhri), [0087], [0101], [0218], [0220], [0328-0331], Figures, 5B-C, 9A-9B, 5KK-5MM.

38. Like the iPhone in 2009, Chaudhri describes functionality where the touch screen mobile device detects the user’s contact on an application icon, triggers the “reconfiguration mode” when the user performs a touch-and-hold gesture on an

icon, and moves an application icon when the user performs an associated drag gesture. That this aspect of Chaudhri's disclosure corresponds to well-known iPhone functionality is no surprise because Chaudhri expressly identifies the iPhone as one of the "[e]xemplary embodiments." EX1004, [0073]; *see also* [0088], [0322].

1. First Application Icon Mapping

39. Dr. Madisetti argues that Apple's Petition is inconsistent because it discusses the claimed *first application icon* with reference to "either the Photos icon or the Solitaire icon" in its Element [1.g] analysis about *detecting* the user's touch on an icon and with reference to "the Racing icon" in its Element [1.l] analysis about *moving* the icon. EX2005, ¶76; *see also* ¶¶73-76. Having reviewed Dr. Madisetti's theory, I believe he is both missing the point and promoting an unreasonably narrow view of Chaudhri's disclosure. Dr. Madisetti's theory overemphasizes the non-limiting examples in Chaudhri's figures. His theory assumes a POSITA would view Chaudhri's disclosed functionality as limited to only specific icons highlighted in specific figures.² But that is not how a POSITA would have understood Chaudhri.

² Note that I refer to the interpretation of Chaudhri that Dr. Madisetti's theory assumes because Dr. Madisetti does not expressly state that a POSITA would have taken such a narrow view of Chaudhri. For example, when summarizing Chaudhri's disclosure, Dr. Madisetti does not identify any specific application icons. He generically refers to "applications." EX2005, ¶63.

40. The figures in Chaudhri provide visual aids to help the reader visualize certain functionality. They are just examples, and a POSITA would have understood them to be nothing more or less. Chaudhri makes this understanding explicit by consistently using terms like “exemplary,” “example,” “embodiment,” “e.g.,” and “illustrates” when introducing and describing the figures. *E.g.*, EX1004, [0213] (“FIGS. 5A-5PPP illustrate exemplary user interfaces for creating and managing folders that contain one or more of the selectable user interface objects in accordance with some embodiments.”).

41. Moreover, a POSITA would have appreciated that Chaudhri’s disclosure focuses on techniques for reorganizing and rearranging the icons presented on a user interface, including selectable application icons. It is apparent from reading Chaudhri that the specific kind of application icons being reorganized/rearranged—e.g., Solitaire, Photos, Racing, etc.—is unimportant. They are parallel examples that are treated equally in the portions of Chaudhri’s disclosure about icon selection, icon movement, and entering the reconfiguration mode. Note, for instance, that the excerpt below from Chaudhri refers to a broad category of “selectable user interface objects” that includes “a plurality of action icons 5002,” such as exemplary “application icons” 5002-2 through 5002-16.

Attention is now directed towards FIG. 5A, which includes a plurality of selectable user interface objects, including a plurality of action icons 5002 and a plurality of folder icons 5004. For example, in FIG. 5A the action icons 5002 include a plurality of application icons (e.g., photos application icon 5002-1, clock application icon 5002-2, browser application icon 5002-3, solitaire application icon 5002-4, weather application icon 5002-5, notes application icon 5002-6, texts application icon 5002-7, maps application icon 5002-8, stocks application icon 5002-9, camera application icon 5002-12, racing application icon 5002-13, email application icon 5002-14, phone application icon 5002-15, and iPod application icon 5002-16) for activating respective applications[.]

EX1004, [0216].

42. As a result, when Chaudhri describes a general functionality in the exemplary context of one application icon, the POSITA would have understood that the same functionality applies to all of the other application icons. Using the excerpt above to illustrate, Chaudhri describes the “user interface objects”—and thus all of the action icons 5002 and constituent application icons 5002-1 through 5002-16—as being “selectable.” EX1004, [0216]. To a POSITA, that means any of the application icons 5002-1 through 5002-16 can be selected by a user to launch a respective application when the device is in a normal mode of operation. “In other words, selecting one of the action icons will activate an application (e.g., launch an application that is not currently running on the device or display a view of an application that is currently running on the device).” EX1004, [0217].

43. Likewise, when describing icon movement in the reconfiguration mode, Chaudhri broadly refers to the entire class of “user interface objects,” which includes “action icons 5002” and its constituent application icons 5002-1 through 5002-16:

In some embodiments, while the device is in the user interface reconfiguration mode, the selectable user interface objects (e.g., action icons 5002 and folder icons 5004) oscillate about respective average positions so as to indicate that the device is in user interface reconfiguration mode and that the selectable user interface objects (e.g., action icons 5002 and folder icons 5004) can be moved around the display.

EX1004, [0301].

44. This understanding is confirmed by the fact that Chaudhri consistently provides a general teaching illustrated by a non-limiting “example” that involves one or more of the specific application icons. The following excerpt is illustrative.

In some embodiments, the device detects a request to enter user interface reconfiguration mode. For example, in FIG. 5A, the device detects a contact 5008 with the solitaire application icon 5002-4 for more than a predetermined period of time (e.g., 2 seconds) and, in response, the device enters a user interface reconfiguration mode, as illustrated in FIG. 5B.

**General
Description**
**Non-Limiting
Example**

EX1004, [0220]. Chaudhri starts by generally describing that “the device detects a request to enter user interface reconfiguration mode,” and he builds on that general description by providing a non-limiting example involving the Solitaire icon.

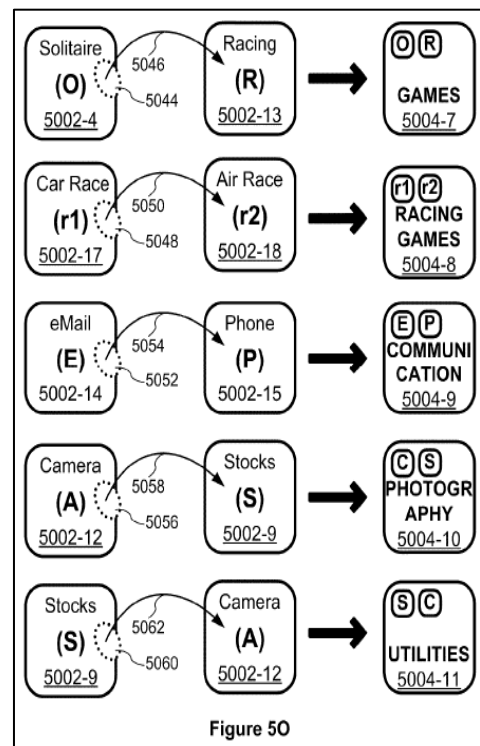
45. Chaudhri uses the same approach when discussing icon movement:

Attention is now directed towards FIGS. 5B-5F, which illustrate exemplary user interfaces for creating a new folder in accordance with some embodiments. In FIG. 5B, the device detects a request to move a respective selectable user interface object to an edge of the screen. In this example, the request includes a contact 5016-a on the touch-sensitive surface (e.g., touch screen 112) at a location that corresponds to a “racing” action icon 5002-13, and subsequent movement 5018 of the contact to the edge of the touch-sensitive surface (e.g., to contact position 5016-b on the touch screen 112, as illustrated in FIG. 5C).

**General
Description
Non-Limiting
Example**

EX1004, [0220]. Here, Chaudhri first generally describes that each “respective” application icon (“selectable user interface object”) is movable, and then illustrates the broader concept with a specific, non-limiting example—the Racing icon.

46. Under Dr. Madisetti’s theory, these passages from Chaudhri would teach that a user can only enter the reconfiguration mode with a touch-and-hold on the Solitaire icon or only move the Racing icon, because that is what Figures 5A-C illustrate. I cannot agree with such a narrow interpretation of Chaudhri. It is not reasonable from the perspective of a POSITA, and it contradicts other portions of Chaudhri, such as Figure 5O (at right), that show and describe the use of drag gestures to move a wide variety of different application icons. *E.g.*, EX1004, [0227] (Notes icon, Figure 5J), [0229]

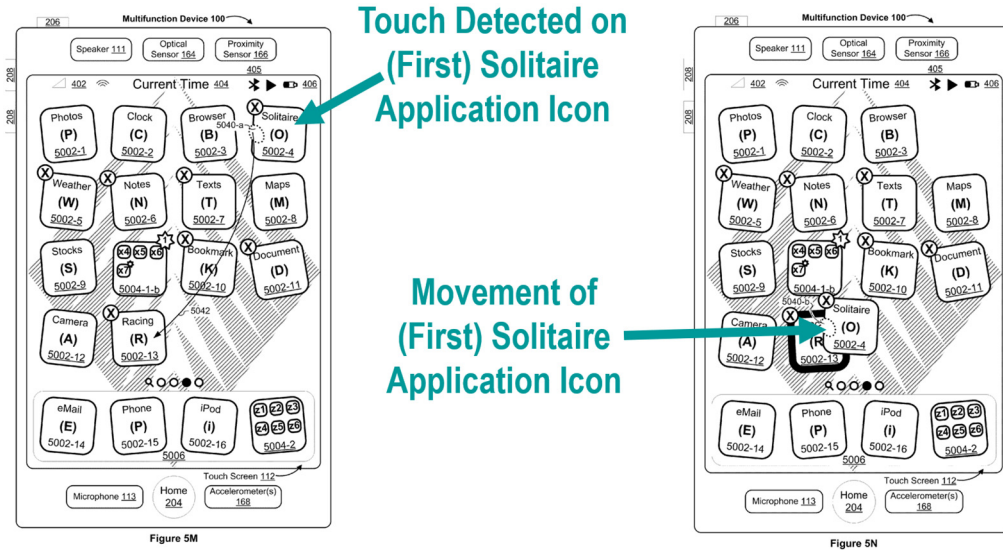


(Solitaire icon, Figures 5M-5N), [0230] (multiple icons, Figure 5O), [0248] (Stocks icon, Figures 5KK-5MM), [0252] (Maps, Figures 5OO-5PP).

47. Going back to the outset of this discussion, Apple's iPhone back in 2009 provides yet another reason why the POSITA would not adopt the narrow, improper view that Dr. Madisetti's theory demands. Again, the iPhone (OS 3.1 Software) included a feature that allowed users to "customize the layout of icons on the Home screen" by performing a "touch and hold" on "any Home screen icon until the icons begin to wiggle" and then "arrang[ing] the icons by dragging them." EX1045, p.25; EX1003, ¶101, ¶41. Chaudhri, a published patent application filed in 2010, repeatedly references to the iPhone, and a POSITA would have understood Chaudhri to be describing the same general technology (and more). *See* EX1004, [0073], [0088], [0322]. So, just as it was well known that "any" application icon on an iPhone could be (i) selected with a tap to launch an application, (ii) used to trigger the reconfiguration mode with a touch-and-hold, and (iii) moved with a drag, the POSITA would have had the same understanding about Chaudhri's disclosure.

48. Further still, even if I were to adopt the narrow interpretation of Chaudhri that Dr. Madisetti's theory requires (which I disagree with), it would not change the analysis or conclusions in my Prior Declaration. My analysis of Element [1.g] identifies an illustrated example in Chaudhri's 5A of detecting a user's touch on the Solitaire application, (EX1003, ¶¶83-87), and my analysis at Element [1.l]

identifies an illustrated example in Chaudhri's Figures 5M-N of moving the same Solitaire application when the user performs a drag, (EX1003, ¶¶100-104).



EX1004, Figures 5M-5N (annotated); *see also* [0229].

49. Additionally, in my Prior Declaration, I presented alternative obviousness rationales where a touch-and-hold gesture on an application icon for entering Chaudhri's reconfiguration mode and a drag gesture for moving the icon are part of a detected single continuous touch. *See* EX1003, ¶¶105-111, ¶¶144-153. These alternative rationales, which I discuss below, are unaffected by Dr. Chaudhri's allegation that the claimed *first application icon* has been inconsistently mapped.

2. Continuous Touch

50. In my Prior Declaration, I noted that "the '758 patent's claims do not include express language stating that the touch gestures embraced by Elements [1.i]-[1.l] must occur as part of a single continuous touch." EX1003, ¶105. Nevertheless,

I explained, “even if the ’758 patent’s claims were interpreted to require a single continuous touch between Elements [1.k] and Elements [1.l], Claim 1 would be obvious.” EX1003, ¶106. The reasoning is straightforward, as shown in the excerpts below:

[A] POSITA would have recognized that there are only three apparent design options to implement Chaudhri’s teachings for using a drag gesture to move an application icon after shifting from the “normal mode” to the “reconfiguration mode” via a touch-and-hold gesture. First, the device and interface could be designed so that the user must keep their finger down on the touch screen between the touch-and-hold gesture and the drag gesture. A POSITA would have appreciated that this option would allow the user to efficiently activate the “reconfiguration mode” at a particular target application icon and drag that target icon to a new location using a single gesture, without need to release the finger and start a second gesture at the same target icon. Second, the device and interface could be designed so that the user must break contact with the touch screen between the two gestures—i.e., inputting a touch-and-hold gesture followed by a distinct touch-and-drag gesture. Third, the device and interface could be designed so that the user may input the two gestures either as a single continuous touch or as consecutive touches with a lift off in between.

* * * *

[A] POSITA applying common sense and ordinary creativity to the problem at hand would have appreciated that the most natural way to implement Chaudhri’s teachings would be to accommodate the third of the above-discussed design options, making the device and interface easier to use by providing multiple avenues for the user to move the application icon once activating the “reconfiguration mode.” In particular, a POSITA would have understood that the third option provides the greatest flexibility to the user, including the possibility for the user to change their mind about which icon to drag to a new location after activating the “reconfiguration mode.”

EX1003, ¶¶107-108.

51. Dr. Madisetti criticizes the section of Apple's Petition that relies on my obviousness analysis excerpted above. *See* EX2005, ¶¶77-86. For example, in Dr. Madisetti's view, the analysis "assumes Chaudhri is silent" about whether a continuous touch or two non-continuous touches are employed in the process of moving application icons. EX2005, ¶83; *see also* ¶¶85-86. For all the reasons below, I disagree with Dr. Madisetti.

52. Before addressing Dr. Madisetti's position in full, I note that my Prior Declaration referred to the notion of a continuous touch between Chaudhri's touch-and-hold gesture on an icon to trigger the reconfiguration mode (Element [1.k] analysis) and Chaudhri' drag gesture for moving an icon (Element [1.l] analysis). Dr. Madisetti instead references Element [1.g], regarding input detection, and Element [1.l], regarding icon movement. This distinction does not alter my analysis or my conclusion. The continuous touch addressed in my Prior Declaration extends from the point of detected contact, through the touch-and-hold, and through the drag.

53. Getting back to the matter at hand, I disagree with Dr. Madisetti's suggestion that Chaudhri is limited to a discontinuous touch. As I explained in my Prior Declaration, a POSITA would have understood that Chaudhri explicitly discloses sufficient touch event monitoring/handling components for recognizing multiple gestures with a single continuous touch, which was well known and commercially implemented long before 2011. EX1003, ¶¶109-110. Regardless,

whether or not Chaudhri describes examples where the touch-and-hold gesture and the drag gesture are non-continuous touches does not change the fact that the three options I outlined in my Prior Declaration are the “only three apparent design options to implement Chaudhri’s teachings.” EX1003, ¶107. Dr. Madisetti does not disagree. Nor does Dr. Madisetti contest my view [i] that two of the three possible design options involve a continuous touch; and [ii] that a POSITA would have appreciated that all three options lead to the same result—“the ability to transition to a reconfiguration mode and move and application icon.” EX1003, ¶108. Thus, it follows that all three options—the majority of which involve a continuous touch—would have been obvious.

54. Dr. Madisetti also fails to effectively rebut my position that “the most natural way to implement Chaudhri’s teachings would be to accommodate the third of the above-discussed design options,” where “the user may input the two gestures either as a single continuous touch or as consecutive touches with a lift off in between.” EX1003, ¶108. He again inaccurately suggests Chaudhri is limited to two non-continuous touches, which is not true, as I have explained. EX2005, ¶86; EX1003, ¶¶109-110. And even if Chaudhri really was limited to two non-continuous touches—which is the “second” of the three possible options I outlined—a POSITA would have been motivated to improve Chaudhri by adding the additional capability to also recognize the touch-and-hold and drag gestures within a single

continuous touch. Doing so would not disturb or detract from Chaudhri's disclosure in any way. On the other hand, doing so would lead to the predictable and beneficial result of "making the device and interface easier to use by providing multiple avenues for the user to move the application icon once activating the 'reconfiguration mode.'" EX1003, ¶108. Dr. Madisetti does not contest this aspect of my analysis.

55. Further evidence of the error in Dr. Madisetti's suggestion that Chaudhri is limited to non-continuous touches is provided by specific examples in Chaurhri. For example, Chaudhri discloses a continuous gesture that consists of multiple parts, including contact with an icon, movement along the touch-sensitive surface, and then pausing for longer than a predetermined time at the display edge to trigger a display change. "In this example, the request includes a contact 5016-a on the touch-sensitive surface (e.g., touch screen 112) at a location that corresponds to a 'racing' action icon 5002-13, and a subsequent movement 5018 of the contact to the edge of the touch-sensitive surface (e.g., to contact position 5016-b on the touch screen 112, as illustrated in FIG. 5C). In FIG 5C, the action icon 5002-13 is moved to the edge of the display for more than a predetermined time threshold (e.g., 1 second). In response to detecting the action icon 5002-13 at the edge of the display (e.g., touch screen 112) for more than the predetermined time threshold, the device

navigates to the next set/page of selectable user interface objects (e.g., as illustrated in FIG. 5D).” EX1004, [0220].

56. As yet another alternative basis for addressing the design option of enabling Chaudhri’s touch-and-hold gesture and drag gesture to be performed as a continuous touch, I relied on the supplemental disclosure of the Os reference (EX1007). As I explained,

Os, a published patent application filed in 2010, demonstrates that “continuous multipart gesture[s]” were known and implemented before the ’758 patent was filed. EX1007 (Os), [0182]. In one example, Os describes detecting a “tap and hold gesture” on a “touch-sensitive surface...for more than a predetermined period of time...and subsequent lateral movement of the contract.” EX1007 (Os), [0182].

EX1003, ¶145. After summarizing Os’s pertinent disclosure, I provided multiple motivation-to-combine rationales. *See* EX1003, ¶¶147-148.

57. Dr. Madisetti does not dispute that Os teaches a continuous touch-and-hold plus a drag. Nor does Dr. Madisetti deny any of the motivation rationales I outlined. Instead, he suggests that Os’s teachings are inconsistent with the “third” design option that I said a POSITA would have preferred in my Prior Declaration. But that is not accurate. Again, in the “third” design option, “the user may input the two gestures either as a single continuous touch or as consecutive touches with a lift off in between.” EX1003, ¶108. Os teaches the former and is, thus, fully compatible with the idea of providing sufficient capability to also recognize the latter.

58. Dr. Madiseti's view that "[a] POSITA would not remove [the discontinuous touch] functionality already described by Chaudhri" suffers the same logical flaw. EX2005, ¶89. Os's teaching to make the touch-and-hold and drag gestures detectable as part of a "continuous multipart gesture" is merely an additional functionality that can be used in conjunction with Dr. Madiseti's view that Chaudhri discloses these gestures as two separate touches. Providing an interface that recognizes the hold and drag as either part of a single continuous touch or as multiple distinct touches would create the POSITA's preferred ("third") design option.

D. The Challenged Dependent Claims Are Obvious

1. Claims 21, 22

59. In my Prior Declaration, I explained that Martyn's selectable items, including the items for "unread emails" and "Messages," correspond to the claimed *action option objects*. EX1003, ¶116 (citing Elements [19], [20]) and EX1005 (Martyn), Figure 3, [0017-0018]). Dr. Madiseti appears to confuse the *action option objects* in the claim with the also recited *previously created content*. EX2005, ¶¶148-151. To be clear, the "3 unread emails" and "1 Chat ongoing" items in Martyn's App Snapshot Menu are mapped in my Prior Declaration to the *action option objects*, not to the *previously created content*. EX1003, ¶116; EX1005 (Martyn), Figure 3, [0017-0018]. Rather, the *previously created content*

corresponds to what the device *displays* in response to selection of one of these *action option objects*.

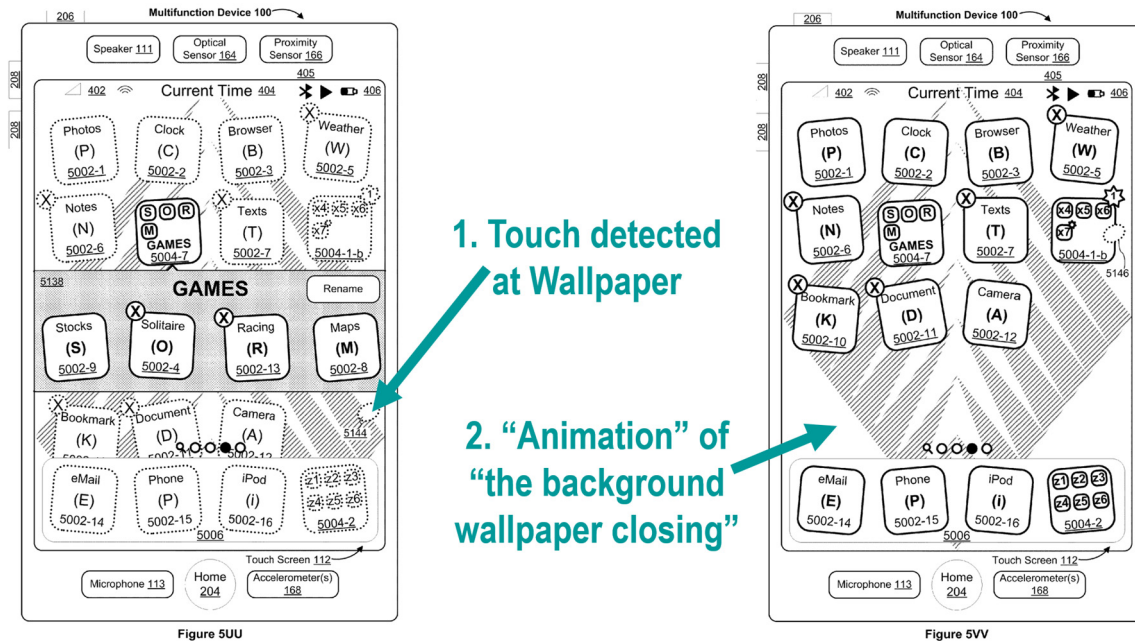
60. To illustrate, the portion of Martyn I cited explains that “when an item in the App Snapshot is selected...the device displays the relevant data.” EX1005, [0018]; EX1003, ¶116 (citing same). So, when the “3 unread emails” item is selected, the device responds by displaying the indicated unread emails. Likewise, when the “1 Chat ongoing” item is selected, the device responds by displaying the messages in the ongoing chat. It would have been obvious or apparent to a POSITA that the messages in the unread emails and the ongoing chat were created before selection of the corresponding items in the App Snapshot. Thus, they render obvious the claimed *previously created content*.

2. Claim 47

61. My Prior Declaration explains why Chaudhri’s wallpaper corresponds to the claimed *user interface element*. EX1003, ¶139. I understand that Dr. Madisetti interprets this term—“user interface element”—to further require an element “that a user must be able to interact with.” EX2005, ¶¶55-58; 157-158.

62. Without addressing the merit of Dr. Madisetti’s interpretation, I note that Chaudhri’s wallpaper still satisfies the claimed *user interface element*. For example, a user can interact with Chaudhri’s wallpaper in several different ways. As one example, Chaudhri teaches that the wallpaper responds to a user interaction by

“divid[ing] into a first portion and a second portion” “in response to detecting the [user’s] input.” EX1004 (Chaudhri), [0047]; *see also* [0045]-[0050], [0237]-[0238] (discussing wallpaper “animation”), [0239], [0244], [0257], [0268], [0368]-[0372], Figures 5X-5Y, 5DD; 5UU-5WW, 5CCC-5FFF. Chaudhri also illustrates and describes a user interaction with the wallpaper, where in response to a “tap gesture,” “the device displays an animation of...the background wallpaper closing.” EX1004, [0257]. I show this interaction in the annotated figures 5UU-5VV below.



EX1004, Figures 5UU-5VV (annotated), [0257], *see also id.* [0309], [0377].

63. As another example, a POSITA would have known that a user could interact with a background wallpaper by selecting/assigning a described photograph to the wallpaper. EX1045, p.102, *see also id.* p.23, p.25, p.97, p.142.

3. Claim 57

64. My Prior Declaration explains that Martyn *dynamically generates* the items shown in the App Snapshot based on usage patterns. EX1003, ¶140 (citing EX1005, [0026]). Dr. Madisetti contends that this teaching from Martyn is distinct from “dynamic generation.” EX2005, ¶¶159-160. Specifically, Dr. Madisetti states that “dynamic generation implies the creation of new content,” while “Martyn merely selects from already existing content.” EX2005, ¶160. I disagree that the claim “implies” creating new content in a manner that distinguishes Martyn. Dr. Madisetti does not dispute that the constituent items of Martyn’s App Snapshot vary based on usage patterns. That means the App Snapshot items—each of which includes text *content* such as “3 unread emails,” “2 new SMS,” “1 Chat ongoing,” “Create message,” and “Enter chat room,” (EX1005, Figure 3)—are *generated dynamically* depending on the circumstances (the user’s usage patterns). This generated text content is created or new in the sense that it is brought into existence by the device when called for by the user’s usage patterns.

65. Additionally, I note that to the extent there exists a corresponding embodiment in the ’758 patent’s specification it relates to a “favorites panel” that “may be automatically populated by the device, based upon factors which may include...historical usage...frequency of use...and/or any other factors.” EX1001, 69:18-24, 84:56-62. This is nearly identical to Martyn’s disclosure that the device

“may be able to learn what functionality [of the application]...[is] most likely to be of interest to a given user and which should therefore be included in a summary view to any given user.” EX1005 (Martyn), [0026]. This similarity between the ’758 patent and Martyn confirms my analysis and conclusion.

IV. CONCLUSION

I declare that all statements made in this Declaration of my own knowledge are true and that all statements made on information and belief are believed to be true.

I further declare that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both (under Section 1001 of Title 18 of the United States Code).

Executed this 4th day of August, 2025.

A handwritten signature in black ink, appearing to read 'Andrew Cockburn', written over a horizontal line.

Andrew Cockburn