

Apple Inc. (Petitioner)
v.
Smith Interface Technologies, LLC (Patent
Owner)
Petitioner Demonstratives

Case No. IPR2024-01087
U.S. Patent No. 10,656,755

Before Hon. Jameson Lee, Miriam L. Quinn, Jeffrey S. Smith
Administrative Patent Judges

APPLE 1036
Apple v. Smith Interface
IPR2024-01087

Table of Contents

Overview of the Instituted Grounds	3
Background Info: The ‘755 Patent and Prior Art	5
Issue 1: The Combination of Hackborn and Westerman Renders the Challenged Claims Obvious (Ground 1A)	13
1A: Hackborn’s Device Distinguishes Between Swipes and Taps	14
1B: Westerman Teaches Distinguishing Swipes and Taps	24
1C: Hackborn Explicitly Teaches Composite Wallpapers	30
Issue 2: Townsend Teaches Simulating “Real-World Behavior” of Objects (Ground 1B)	32
Issue 3: The Dependent Claims Are Obvious	43
Issue 4: The Combination of Hackborn2, Westerman, and Jobs Renders the Challenged Claims Obvious (Ground 2)	50
4A: Jobs’ Scrolling Makes Visible the Obscured Messages of Hackborn	51
4B: Hackborn’s Wallpaper is in the Background and Its System Would Work with Manual Scrolling	56
4C: Westerman’s Teachings Are Closely Related to Hackborn and Jobs	61

Overview of the Instituted Grounds

Overview of the Instituted Grounds

Ground	'755 Patent Claims	Basis for Rejection
1A	1-3, 7, 10-11, 13, 15, 17, 20, 22-25, 29, 32-33, 35, 37-38, 41-42	Hackborn, Westerman ("HW")
1B	4-5, 8-9, 21, 26-27, 30-31, 43	Hackborn, Westerman, Townsend ("HWT")
1C	16	Hackborn, Westerman, Jablonski ("HW-Jablonski")
1D	18-19, 39-40	Hackborn, Westerman, Jobs ("HWJ")
2	1, 4-8, 10-12, 15, 17-20, 22-23, 26-30, 32, 34, 37-42	Hackborn, Westerman, Jobs ("H2WJ")

Pet., 2; see Inst. Dec., 5-6, 32

Background Info: The '755 Patent and Prior Art

Exemplary Claim of the '755 Patent

1. An electronic device, comprising:

a display; a touch-sensitive surface; one or more processors; memory; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors,

the one or more programs including instructions for: displaying a first user interface on the display, wherein the first user interface includes:

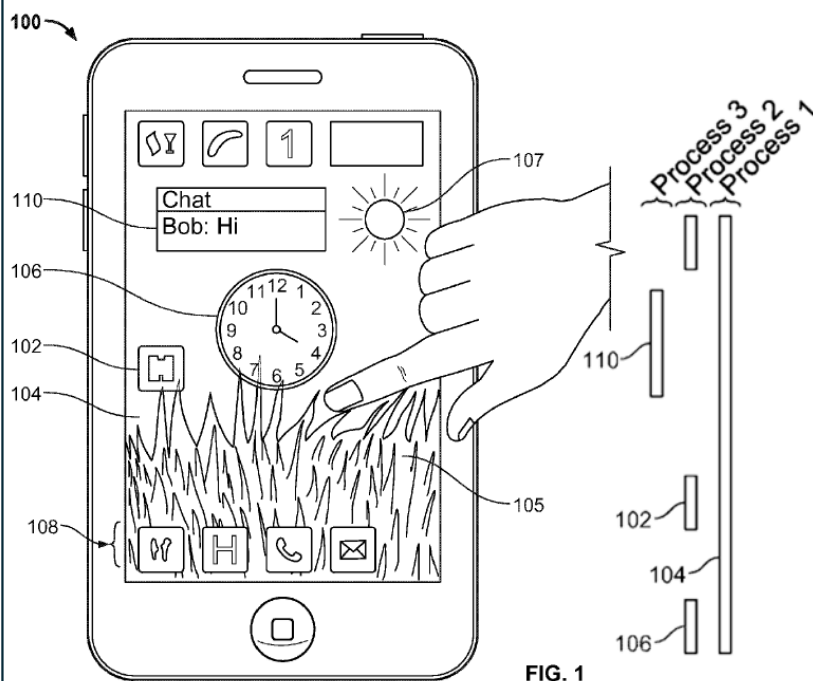
a background with an appearance, and

one or more foreground objects;

while displaying the first user interface on the display, detecting a first input by a first contact on the touch-sensitive surface at a location in the first user interface that corresponds to the background of the first user interface; and

in response to detecting the first input by the first contact, in accordance with a determination that the first contact has a magnitude that is above a threshold, dynamically changing the appearance of the background of the first user interface without changing an appearance of the one or more foreground objects in the first user interface, wherein the dynamic change in the appearance of the background of the first user interface is based at least in part on the magnitude of the first contact and wherein the dynamic change in the appearance of the background of the first user interface includes displaying in sequence at least some of a plurality of images based at least in part on the magnitude of the first contact.

Primary Reference: Hackborn



APPLE-1004, [0026], [0028], [0033],
FIGS. 1-2; Pet., 4-6, 18-19, 31

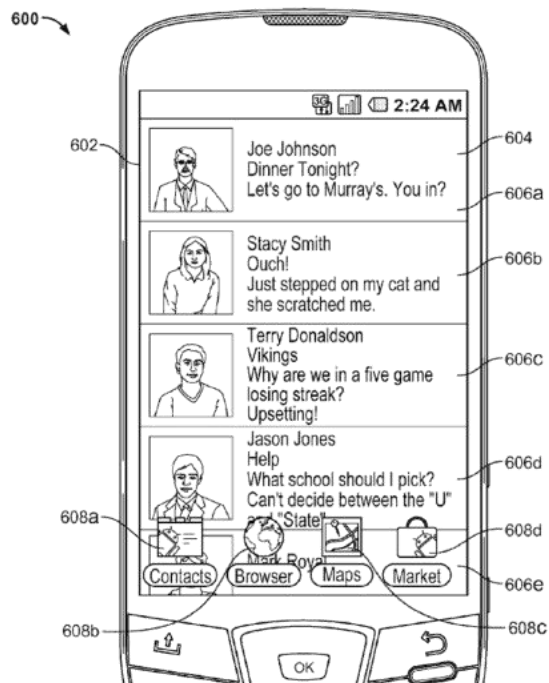
[0026] A general home screen layout is shown in FIG. 1, where a computing device 100 (e.g., a smart phone, tablet device, laptop device, pad device, personal digital assistant, electronic reading device, or the like) with a display screen presents a display, but in this example, the wallpaper 104 is live. An example of such a state of the wallpaper is shown in the figure by the wallpaper's 104 reaction to a user input on a screen of the device 100. In particular, in this example, the

[0028] In the example shown in FIG. 1, a user is swiping her finger across a touch-sensitive display screen of the computing device. In response, the blades of grass 105 displayed in the area of the screen swiped by the user are caused to bend or sway as if contacted by the user's finger during the swipe. Other examples of live wallpaper may not be immediately

Rather, the input may be received by a home screen process, which may interpret the input and pass it to the wallpaper 104 process. The message that passes the information may indi-

cate a vector representing the position, direction, and speed of the user's dragging motion. Such information may then be used by the process executing the wallpaper 104 to determine that the user's finger contacted items on the wallpaper—here, the grass 105. Such information may be used to generate an animation that makes the grass blades look as if the finger is actually contacting them as it drags across the screen of the device 100.

Primary Reference: Hackborn ("Hackborn2")



[0046] FIG. 6 shows yet another example of active content being produced as live wallpaper on a computing device 600. A display 602 of the device 600 displays the live wallpaper 604, which includes information fetched or received from one or more social services on the Internet. The information includes posts or tweets 606a-e from a social networking site (e.g., Google Buzz, Facebook, MySpace, Flickr, Twitter, and the like) posted by friends, acquaintances, group members or contacts of the user, for example and the information may be received by the device from a web server associated with the networking site. The depicted posts 606 include an image in

[0047] The posts 606a-e may be displayed in various manners. For example, as new information is received, already displayed posts may be scrolled (e.g., downward) and the new post may be displayed in a predetermined position. For example, post 606a in the present example may be the most recently received message, and may be displayed near the top of the display 602. When post 606a is received, the previously displayed posts (e.g., posts 606b, 606c, 606d, 606e) may be shifted downward. As can be seen, post 606e is only partially displayed near the bottom of the display 602. Icons 608a-d are shown on a display layer above the wallpaper 604. While no application windows are shown in FIG. 6, it is understood that a user could launch an application and work in an application area on the display 602, where the application area would be displayed above the wallpaper layer and the icon layer, for example. In various implementations, there can be multiple application layers. The live wallpaper 604 may continue to update in the background, whether the user is working on applications or not, and may be updated in the background. When wallpaper updates occur and the appearance of the wallpaper 604 is caused to change, the wallpaper may continue to occupy the bottom-most visual display layer or plane of the display 602, for example below the icon or gadget layer and further below one or more application layers, if applicable.

Secondary Reference: Westerman

SUMMARY

[0018] The present invention can relate, for example, to a method of interpreting swipe gesture input to a device having a touch-sensitive input. The touch-sensitive input can include

[0061] In either case, the path data structure (1004 or 1005) is submitted to a displacement measurement process (block 1006). The displacement measurement process can determine how much the path has moved in both horizontal direction ($D[i].x$), how much the path has moved in the vertical direction ($D[i].y$), and over what time ($T[i]$). The total distance moved can then be compared to a minimum threshold of movement used to determine whether the touch event is a tap or a swipe (block 1007). If there is little motion, i.e., less than the threshold, the event is interpreted as a key tap, and the system updates the key choice that corresponds to the location of the finger (block 1008).

[0019] Detecting a swipe gesture can include acquiring touch image data from the touch-sensitive device, processing the image to generate one or more finger path events, determining a displacement of the one or more finger path events, and detecting a swipe gesture if the displacement exceeds a predetermined threshold. If the displacement does not exceed the threshold, the input can be interpreted as a conventional tap. The time of the motion associated with the input can also be compared to a maximum swipe gesture timeout threshold. If the timeout threshold is exceeded, the input can be interpreted as a conventional tap.

Secondary Reference: Townsend

[0004] Another problem is the difficulty for applications to determine a user's intentions based on touch input received from multi-touch hardware. Touch input may be received as a list of coordinates where the hardware senses touch input at any given time. Each application has to include software to interpret the coordinates and determine the user's intention. In addition, the user's intention may extend beyond the actual touch input received. The user may expect virtual objects to behave how they do in the physical world. For example, a user

may expect to be able to "toss" a file from one side of the desktop to another by flicking his/her finger. This type of movement is not supported by current multi-touch applications, which would expect the user to drag his/her finger from one side of the screen all the way to the other. Even if an application provides support for this type of movement, other applications could not benefit from it and thus application authors would have to repeat the work of the first application author to offer the same functionality in their applications.

SUMMARY

[0005] The inertia system provides a common platform and application-programming interface (API) for applications to extend the input received from various multi-touch hardware devices to simulate real-world behavior of objects. The manipulations received by the application only describe the movement of an object based on the movement of contacts with the multi-touch hardware. However, to move naturally, objects should also exhibit physical characteristics such as elasticity and deceleration. When a user lifts all contacts from an object, the inertia system provides additional manipulation events to the application so that the application can handle the events as if the user was still moving the object with touch. However, the inertia system actually generates the events based on a simulation of the behavior of the objects. If the user moves an object into another object, the inertia system sends manipulation events based on the boundary characteristics of the objects. Thus, the inertia system provides more realistic movement for application objects that a user manipulates using multi-touch hardware and the API provides a consistent feel to manipulations across applications.

Pet., 44-45; APPLE-1006, [0004]-[0005]

Secondary Reference: Jobs

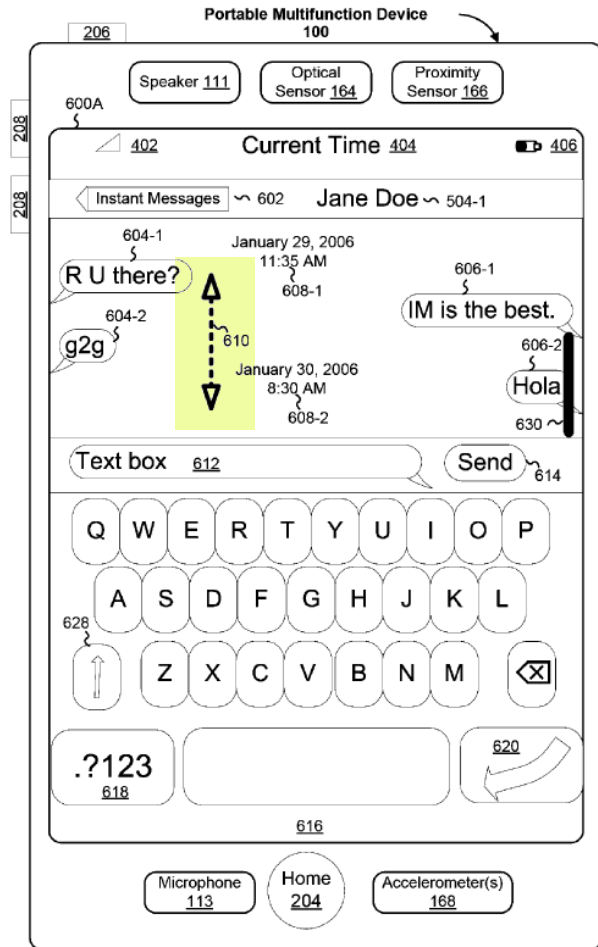


Figure 6A

Hackborn

ners. For example, as new information is received, already displayed posts may be scrolled (e.g., downward) and the new post may be displayed in a predetermined position. For example, post 606a in the present example may be the most recently received message, and may be displayed near the top of the display 602. When post 606a is received, the previously displayed posts (e.g., posts 606b, 606c, 606d, 606e) may be shifted downward. As can be seen, post 606e is only partially displayed near the bottom of the display 602. Icons 608a-d

Jobs

[0243] In some embodiments, a user can scroll through the message conversation (comprised of messages 604 and 606) by applying a vertical swipe gesture 610 to the area displaying the conversation. In some embodiments, a vertically downward gesture scrolls the conversation downward, thereby showing older messages in the conversation. In some embodiments, a vertically upward gesture scrolls the conversation upward, thereby showing newer, more recent messages in the conversation. In some embodiments, as noted above, the last message in the conversation (e.g., 606-2) is displayed in the list of instant messages 500 (e.g., 506-1).

Secondary Reference: Jablonski

Jablonski

(57)

ABSTRACT

An interactive system and method is disclosed for providing wallpaper or screen saver images for display on the background of a personal computer visual display. Wallpaper (or screen saver) images may be conveniently located by a consumer and downloaded from a wallpaper web site for use on the desktop of the consumer's personal computer. A clickable icon is incorporated into the wallpaper image or overlaid on the wallpaper image. When the icon is clicked on using a mouse, or otherwise activated by the consumer, the consumer's personal computer is automatically connected to a predetermined location on the Internet. Furthermore, sponsors may market their goods or services to consumers by paying a fee to have their clickable icons or advertisements incorporated into or overlaid on the wallpaper images. By clicking on the clickable icons or advertisements, the consumer is automatically connected to the sponsor's web site. In addition, a number of web sites may be affiliated with the wallpaper site, and consumers who subscribe to the wallpaper site may be able to conveniently purchase products or services from these affiliated sites by merely clicking on one or more pop-up buttons, or the like, after necessary consumer information has been given to the wallpaper web site.

The placement of wallpaper images on the background serves to provide a decorative touch to the overall computer display, and gives the computer user the opportunity to personalize the desktop display of his or her computer. Wallpaper images are usually included as part of the computer's operating system, or are added later by the user with the addition of a designated wallpaper program or other software, or by downloading images from various sources on the Internet, such as web pages, commercial sites, or the like. Often wallpaper images are acquired based upon a particular user's personal interests or pursuits.

Issue 1

The Combination of Hackborn and Westerman
Renders the Challenged Claims Obvious
(Ground 1A)

Issue 1A

Hackborn's Device Distinguishes
Between Swipes and Taps

Hackborn teaches “live wallpaper” with overlaying objects

[0026] A general home screen layout is shown in FIG. 1, where a computing device 100 (e.g., a smart phone, tablet device, laptop device, pad device, personal digital assistant, electronic reading device, or the like) with a display screen presents a display, but in this example, the wallpaper 104 is live. An example of such a state of the wallpaper is shown in the figure by the wallpaper's 104 reaction to a user input on a screen of the device 100. In particular, in this example, the wallpaper shows grass 105 below a sunny sky 107, and may ordinarily look like a traditional wallpaper in the form of a static image. However, the wallpaper 104 in this example may

[0015] FIG. 1 shows a computing device displaying a home screen layout that includes live wallpaper.

[0016] FIG. 2 shows a conceptual side diagram of the display shown on the device of FIG. 1.

display, and is shown furthest right in FIG. 2. Thus, the wallpaper 404 tends to appear to the user as though it were behind all icons, gadgets, images or other items that appear on the screen. Icon 102 and gadget 106 are shown running in a second environment that corresponds to a plane, surface, or layer of the display at a second or middle level of the display, and is shown left of the first environment in FIG. 2. Application 110 is shown running in a third environment that corresponds to a plane, surface, or layer of the display at a third or top level of the display, and is shown furthest left in FIG. 2.

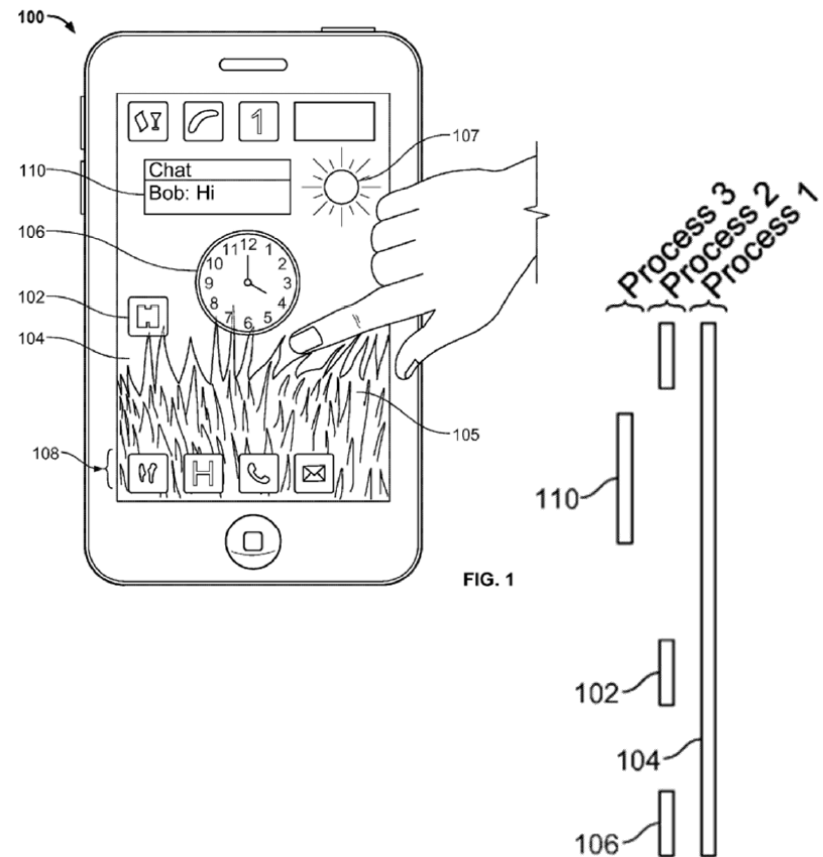


FIG. 2

Pet., 9, 10, 17, 47; APPLE-1004, [0015]-[0016],
[0031], FIG. 2

Hackborn teaches “live” grass wallpaper responsive to swiping

[0026] A general home screen layout is shown in FIG. 1, where a computing device 100 (e.g., a smart phone, tablet device, laptop device, pad device, personal digital assistant, electronic reading device, or the like) with a display screen presents a display, but in this example, the wallpaper 104 is live. An example of such a state of the wallpaper is shown in the figure by the wallpaper's 104 reaction to a user input on a screen of the device 100. In particular, in this example, the wallpaper shows grass 105 below a sunny sky 107, and may ordinarily look like a traditional wallpaper in the form of a static image. However, the wallpaper 104 in this example may

[0028] In the example shown in FIG. 1, a user is swiping her finger across a touch-sensitive display screen of the computing device. In response, the blades of grass 105 displayed in the area of the screen swiped by the user are caused to bend or sway as if contacted by the user's finger during the swipe. Other examples of live wallpaper may not be immediately

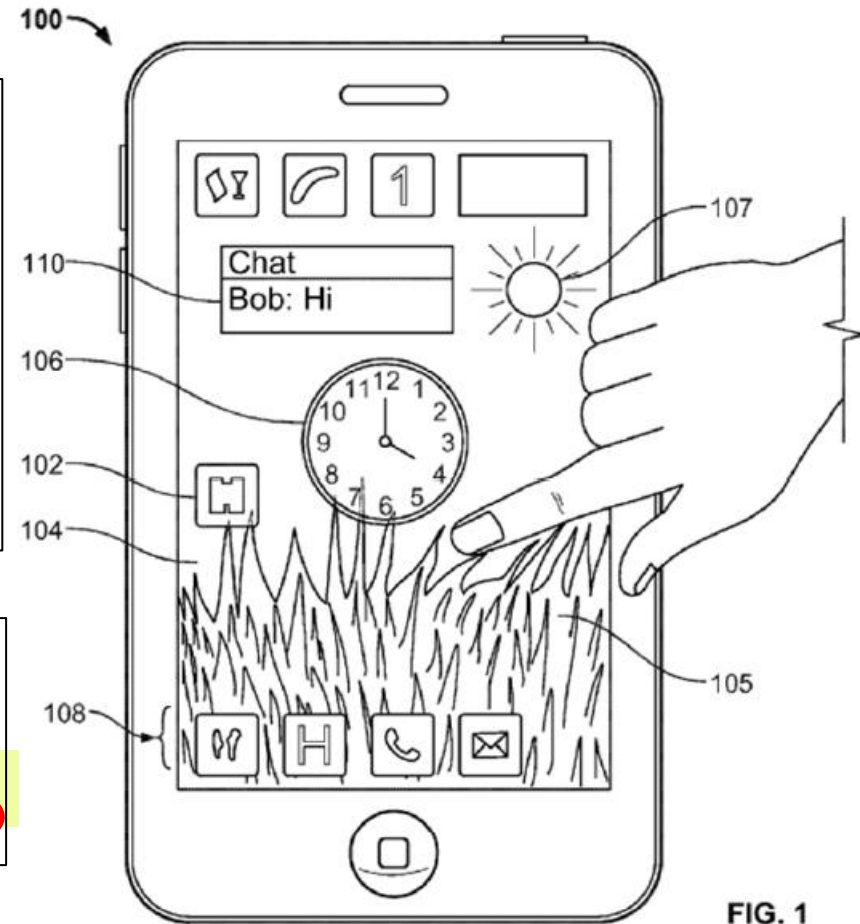


FIG. 1

The experts agree that Hackborn only explicitly teaches that its “live” grass wallpaper responds to swiping

Dr. Mahon's Deposition

15 Q. Does Hackborn provide any other
16 explicit examples of gestures that cause the
17 blades of grass to move?

18 A. Hackborn does not limit the gestures
19 that would be determined to say that the user
20 is making contact with the grass, so I don't
21 recall any others specifically but it doesn't
22 limit it.

Dr. Cockburn's Declaration

19. First, as Patent Owner's expert Dr. Mahon acknowledged, while Hackborn discloses animating grass in response to a *swipe*, Hackborn provides no other explicit example input gestures that cause the glass wallpaper to move.

APPLE-1032, 72:3-11, 119:15-21 (“I don't recall any other [gestures] specifically”). Rather, the entirety of Hackborn's discussion of the live grass wallpaper pertains to its reaction to a *swipe*. For example, Hackborn discloses that swipes animate grass in its live wallpaper, explaining, “a user is *swiping* her finger across a touch-sensitive display screen of the computing device,” and “the blades of grass 105 displayed in the area of the screen *swiped* by the user are caused to bend or sway as if contacted by the user's finger during the *swipe*.” APPLE-1004, [0028].

APPLE-1032, 119:15-22; APPLE-1031, ¶19; Reply, 5

Hackborn's sole mention of "tapping" is unrelated to the live grass wallpaper

[0028] In the example shown in FIG. 1, a user is swiping her finger across a touch-sensitive display screen of the computing device. In response, the blades of grass 105 displayed in the area of the screen swiped by the user are caused to bend or sway as if contacted by the user's finger during the swipe. Other examples of live wallpaper may not be immediately responsive to user input on the screen of the device. Rather, some examples of live wallpaper may be displayed and updated so that an appearance of the wallpaper is changed, but may become responsive to direct user input only after a predetermined user input is received by the device. Examples of predetermined user inputs that may be received by the device to cause the live wallpaper to become responsive to user input can include agitating or shaking the device, selecting an icon, button (not shown in FIG. 1), or area of the display screen (e.g., a corner of the screen) reserved for activating the live wallpaper, making a particular movement on or across the display screen, tapping on the display screen a predetermined number of times within a predetermined time period, and others. Thus, in a first mode, user input on a touchscreen may be interpreted as being directed to icons displayed on a layer over the wallpaper, and in another mode (which can be switched to using the techniques just described), user input may be interpreted as being directed to items displayed on the wallpaper. When such mode switching occurs, the appearance of the wallpaper and the icons or other overlying objects may change, such as by changing the icons to a transparent or shadowed appearance.

Pet., 29; Reply, 2; APPLE-
1004, [0028]

Smith's expert admits Hackborn does not teach how to distinguish various user inputs

Dr. Mahon's Deposition

21 Does the Hackborn reference teach how
22 the Hackborn device knows whether a user input

1 comprises tapping on the display screen a
2 predetermined number of times within a
3 predetermined time period?

4 A. Well, to a POSITA it's clear that
5 they're saying that there's multiple taps
6 within a given time period to make that
7 determination. It doesn't go beyond that
8 teaching, as far as I know.

18 Q. So you mentioned that Hackborn
19 teaches detecting multiple taps for changing
20 mode. How does Hackborn know whether multiple
21 taps are intended to change a mode or whether
22 the user was actually inputting single taps

1 separated from one another?

2 A. That's -- again, it's a general
3 question. It depends upon implementation.
4 It's clear what Hackborn is disclosing, that
5 multiple taps could be used to change the mode
6 of the device. And that's all that Hackborn
7 discloses.

Hackborn is not equipped to animate the grass wallpaper responsive to taps

Dr. Cockburn's Second Declaration

swipe”). In contrast to a swipe, which involves a side-to-side motion that directly translates to the side-to-side motion of grass on the live wallpaper (e.g., based on the speed and length of the swipe), a tap input on Hackborn’s device offers inadequate data from which to generate a responsive movement on the live grass wallpaper.

Reply, 8; APPLE-1031, ¶ 23

Smith's arguments confirm Hackborn's inability to animate the grass wallpaper based on taps

Dr. Cockburn's Second Declaration

Hackborn. Specifically, Smith states that “Hackborn does not need a threshold determination between taps and swipes to initiate movement of the grass because *the grass already moves based on the magnitude of contact,*” that “Hackborn discloses that the user input on the live grass wallpaper generates *vector data representing the position, direction, and speed of the contact,*” and that “the *vector data is what determines the magnitude of the visual effect* of the dynamically generated grass animation.” POR, 25. But the only data that represents any magnitude in the vector that Smith identifies is speed and direction; a POSITA would understand that position, in contrast, is a single coordinate in x, y space, and therefore position lacks any associated magnitude. Importantly, a tap has an x, y position, but it has no speed and no direction, and therefore there is no magnitude on which to, by Smith's own words, move the grass “based on the magnitude of the contact.”

Reply, 8; APPLE-1031, ¶ 24; see APPLE-1032, 89:13-90:12

Hackborn does not teach or suggest use of pressure or a pressure sensor

10 **Q. You earlier alluded to pressure being**
11 **used. Does Hackborn teach the use of pressure**
12 **on its device?**

13 A. I don't see that Hackborn discloses
14 pressure, and I don't recall specifically any
15 part of Hackborn discussing pressure. So I
16 can't find an area in Hackborn specifically
17 discussing pressure.

APPLE-1032, 102:13-17;
Reply, 8; APPLE-1031, ¶ 25

Claims 1 and 23: Smith's arguments concerning Hackborn's live wallpaper criteria are wrong

Patent Owner's Response

swipe.” EX1004, [0026], [0028]. Hackborn explains that the information it collects from an input “may be used by the process executing the wallpaper 104 to determine that the user’s finger *contacted* items on the wallpaper—here, the grass 105.” EX1004, [0033]. Thus, a POSITA would understand that Hackborn’s only disclosed criteria for causing a change to the grass is contact with a finger, not path length or velocity (nor whether such an input is a swipe *instead of* a tap). EX2005, ¶67. Hackborn’s reference to specific functions like a swipe “are meant to be exemplary only, and are not meant to limit implementations.” EX1004, [0061].

Patent Owner's Sur-Reply

Patent Owner’s assertion was focused on criteria for triggering (causing) an animation, which Hackborn clearly states is based on touch. POR, 27–28; EX2005, ¶¶68–69. Path length and velocity are used in determining the magnitude of the visual effect of the dynamically generated grass animation. POR, 25; EX2005, ¶64.

Hackborn

[0033] The processes for different visual layers may also communicate with each other, in some implementations. In this example, a process for the home page itself is communicating with a process for the wallpaper 104. In particular, the user input of dragging a finger across the top of the grass may not be received and processed by the wallpaper 104 process. Rather, the input may be received by a home screen process, which may interpret the input and pass it to the wallpaper 104 process. The message that passes the information may indi-

cate a vector representing the position, direction, and speed of the user’s dragging motion. Such information may then be used by the process executing the wallpaper 104 to determine that the user’s finger contacted items on the wallpaper—here, the grass 105. Such information may be used to generate an animation that makes the grass blades look as if the finger is actually contacting them as it drags across the screen of the device 100.

Issue 1B

Westerman Teaches Distinguishing Swipes and Taps

A POSITA would have looked to Westerman because of Hackborn's silence on distinguishing inputs

Dr. Cockburn's First Declaration

50. In addition to receiving swipe inputs on the touchscreen, Hackborn teaches that its device can receive other “predetermined inputs,” including “tapping on the display screen.” *Id.*, [0028]. In response, Hackborn states that “[t]he touch interface 706 may coordinate with the display controller 704 to identify what item or items on a display a user is selecting by touching at a particular location.” *Id.*, [0054]. To the extent Hackborn lacks details on how to distinguish tap from swipe inputs, a POSITA would have naturally looked to other prior art teachings that discuss “interpreting manual swipe gestures as input in connection with touch-screen keyboards,” such as Westerman. APPLE-1005, [0011].

A POSITA would have been motivated to incorporate Westerman's teachings on gesture detection

Dr. Cockburn's First Declaration

52. A POSITA reviewing the device and methods of Hackborn would have naturally looked to Westerman to better understand the implementation details of Hackborn's teachings, for example, as it relates to how to discern between different types of user inputs—namely, a tap or a swipe. Thus, given that Hackborn explains that its device can register both swipes and taps (APPLE-1004, [0028]), a POSITA would have found it obvious to implement Hackborn with methods of distinguishing between tap and swipe gestures (e.g., using “predetermined thresholds”), such as those disclosed or rendered obvious by Westerman, as indeed these and similar methods were well known by the Critical Date. *See* APPLE-1007, [0027] (defining user inputs based on threshold distances, times, etc.).

Westerman's Teachings Are Broad

Westerman

[0025] The present invention can also relate to a computer system including a multi-touch interface that has been adapted and/or programmed to detect and process swipe gesture input in the various ways described above. The computer system can take the form of a desktop computer, a tablet computer, a notebook computer, a handheld computer, a personal digital assistant, a media player, a mobile telephone, and combinations of one or more of these items. The multi-touch interface can include a touch screen.

[0018] The present invention can relate, for example, to a method of interpreting swipe gesture input to a device having a touch-sensitive input. The touch-sensitive input can include

[0019] Detecting a swipe gesture can include acquiring touch image data from the touch-sensitive device, processing the image to generate one or more finger path events, determining a displacement of the one or more finger path events, and detecting a swipe gesture if the displacement exceeds a predetermined threshold. If the displacement does not exceed the threshold, the input can be interpreted as a conventional tap. The time of the motion associated with the input can also

Westerman offers implementation details absent from Hackborn

Westerman's broad disclosure on distinguishing swipes from taps

[0019] Detecting a swipe gesture can include acquiring touch image data from the touch-sensitive device, processing the image to generate one or more finger path events, determining a displacement of the one or more finger path events, and detecting a swipe gesture if the displacement exceeds a predetermined threshold. If the displacement does not exceed the threshold, the input can be interpreted as a conventional tap. The time of the motion associated with the input can also be compared to a maximum swipe gesture timeout threshold. If the timeout threshold is exceeded, the input can be interpreted as a conventional tap.

[0061] In either case, the path data structure (1004 or 1005) is submitted to a displacement measurement process (block 1006). The displacement measurement process can determine how much the path has moved in both horizontal direction ($D[i].x$), how much the path has moved in the vertical direction ($D[i].y$), and over what time ($T[i]$). The total distance moved can then be compared to a minimum threshold of movement used to determine whether the touch event is a tap or a swipe (block 1007). If there is little motion, i.e., less than the threshold, the event is interpreted as a key tap, and the system updates the key choice that corresponds to the location of the finger (block 1008).

The Petition does not bodily incorporate Westerman

Petition

Since Hackborn explains that its device can register both swipes and taps (APPLE-1004, [0028]), a POSITA would have found it obvious to implement Hackborn with methods such as those disclosed or rendered obvious by Westerman for distinguishing between tap and swipe gestures (e.g., using “predetermined thresholds”). APPLE-1003, ¶52.

A POSITA would have understood that implementing Hackborn’s device to “determin[e] a displacement of one or more finger path events” and “detecting a swipe gesture if the displacement exceeds a predetermined threshold” would provide a clear method for distinguishing between taps and swipes. APPLE-1003, ¶53; APPLE-1005, [0019]. A POSITA would have further understood that implementing Hackborn’s device to use the “initial velocity at touchdown [of the input]” to compare with a particular threshold value” would provide Hackborn’s device with a method “to distinguish a swipe from a tap.” APPLE-1003, ¶53;

Issue 1C

Hackborn Explicitly Teaches Composite Wallpapers

Claims 1 and 23: Hackborn explicitly discloses composite wallpaper

Patent Owner's Response

Petitioner argues that such composites are taught merely because Hackborn states that “live wallpaper can include a combination of the described wallpaper examples.” Pet., 25 (quoting EX1004, [0050]). Petitioner manufactures these composites to insert specific features from one embodiment into another in order to meet a specific claim limitation, and does not provide any explanation of why a POSITA would have combined these particular aspects of those wallpapers *in the manner proposed*. See *Abiomed, Inc. v. Maquet Cardiovascular, LLC*, IPR2017-

Patent Owner's Sur-Reply

interactions. EX1004, [0050]. There is no textual anchor for the specific composites that the Petition manufactures and attempts to rely upon. POR, 29–30.

Hackborn's “textual anchor”

[0050] In various implementations, live wallpaper can include a combination of the described wallpaper examples. For example, images (such as images 406a-h in FIG. 4) and information (such as posts 606a-e in FIG. 6) from social networking sites may be combined for presentation in a live wallpaper display. Similarly, weather information, traffic information, sports scores, stock prices or other information may be shown alongside other live wallpaper content.

Issue 2

Townsend Teaches Simulating “Real-World Behavior” of Objects (Ground 1B)

Townsend's applicability is broad, and consistent with Hackborn

Rebuttal
to:

"Townsend's simulation is not applicable to Hackborn's swaying grass " Sur-reply, 9

Townsend

[0004] Another problem is the difficulty for applications to determine a user's intentions based on touch input received from multi-touch hardware. Touch input may be received as a list of coordinates where the hardware senses touch input at any given time. Each application has to include software to interpret the coordinates and determine the user's intention. In addition, the user's intention may extend beyond the actual touch input received. The user may expect virtual objects to behave how they do in the physical world. For example, a user

may expect to be able to "toss" a file from one side of the desktop to another by flicking his/her finger. This type of movement is not supported by current multi-touch applications, which would expect the user to drag his/her finger from one side of the screen all the way to the other. Even if an application provides support for this type of movement, applications could not benefit from it and thus a second author would have to repeat the work of the first author to offer the same functionality in their application.

Pet. 44-46; APPLE-1006, [0004]; APPLE-1004, [0028]

Hackborn

[0028] In the example shown in FIG. 1, a user is swiping her finger across a touch-sensitive display screen of the computing device. In response, the blades of grass 105 displayed in the area of the screen swiped by the user are caused to bend or sway as if contacted by the user's finger during the swipe. Other examples of live wallpaper may not be immediately

Townsend teaches simulating the behavior of real-world objects

Rebuttal “Townsend’s simulation is not applicable to Hackborn’s swaying grass ” Sur-reply, 9
to:

Townsend

[0016] The manipulations handled by the application only describe the movement of an object based on the movement of contacts. However, to move naturally, objects should also exhibit physical characteristics such as elasticity and deceleration. When a user lifts all contacts from an object, the inertia system provides additional manipulation events to the application so that the application can handle the events as if the user was still moving the object with touch. However, the inertia system actually generates the events based on a simulation of the behavior of the objects. For example, if the user lifted the contacts while the object had a velocity in a particular direction, then inertia system continues sending events that indicate that the object is moving in that direction, slowing down over time as the object decelerates. If the user moves an object into another object, such as the edge of the screen, the inertia system sends manipulation events based on the boundary characteristics of the objects. For example, if an application author defines two objects as being elastic, then the two objects may bounce off each other when a user moves the objects into each other. Thus, the inertia system provides more realistic movement for application objects that a user manipulates using multi-touch hardware and the API provides a consistent feel to manipulations across applications.

Reply, 14-15; APPLE-
1006, [0016]; see
POR, 36

Townsend's teachings are applicable to the “swaying, anchored grass of Hackborn”

Patent Owner's Response

the colliding objects. EX1006, [0016], [0041], [0050]–[0051]. A POSITA would understand that Townsend's inertia system operates on the motion of whole objects moving across a screen, not a comprehensive physics engine that simulates the real world behavior of any object under any condition. EX2005, ¶83.

Townsend

[0014] The inertia system provides a common platform and API for applications to extend the input received from various multi-touch hardware devices to simulate real-world behavior of objects. For example, real world objects do not typically

cation objects may support scaling but not rotation. The initialization information may also specify a pivot point of the object. The inertia system provides manipulation transforms to the application through the application interface. For example, when the inertia system receives low-level touch input that the system interprets as a recognized transform (e.g., a rotation), the system fires an event to notify the appli-

Pet., 45-46; Reply, 15;
POR, 36-37; APPLE-1006,
[0014], [0022]

Claims 5 and 27: Decay occurs after and in response to detecting cessation of the contact

5. The electronic device of claim 1, wherein the electronic device is configured such that a magnitude of the dynamic change in the appearance of the background of the first user interface decays following detection of a cessation of the first contact.

Hackborn

[0008] In various implementations, generating the graphical wallpaper representation may include displaying a visual image that covers substantially an entirety of a display on the computing device. The method may also include changing a manner in which the graphical wallpaper representation is displayed using the generated information. The user interactions may include dragging across the display, and changing the manner in which the graphical wallpaper representation is displayed may include animating motion in the graphical wallpaper representation in a manner that simulates frictional dragging of elements of the graphical wallpaper representation. The animating motion may include bending one or more graphical elements in a direction of the dragging input. The

Townsend

contacts. However, to move naturally, objects should also exhibit physical characteristics such as elasticity and deceleration. When a user lifts all contacts from an object, the inertia system provides additional manipulation events to the application so that the application can handle the events as if the user was still moving the object with touch. However, the inertia system actually generates the events based on a simulation of the behavior of the objects. For example, if the user lifted the contacts while the object had a velocity in a particular direction, then inertia system continues sending events that indicate that the object is moving in that direction, slowing down over time as the object decelerates. If the user moves an object into another object, such as the edge of the screen, the inertia system sends manipulation events based on the boundary characteristics of the objects. For example, if an application author defines two objects as being elastic, then the two objects may bounce off each other when a user moves the objects into each other. Thus, the inertia system provides more realistic movement for application objects that a user manipulates using multi-touch hardware and the API provides a consistent feel to manipulations across applications.

APPLE-1004, [0008]; APPLE-1006, [0016]

Claims 5 and 27: Decay occurs after and in response to detecting cessation of the contact

5. The electronic device of claim 1, wherein the electronic device is configured such that a magnitude of the dynamic change in the appearance of the background of the first user interface **decays following detection of a cessation of the first contact.**

Dr. Cockburn's First Declaration

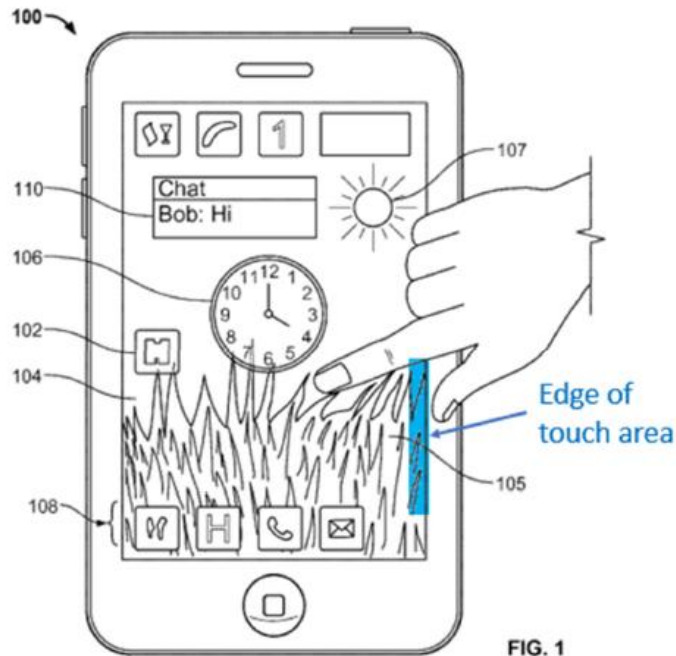
[0028], claims 4 and 19; APPLE-1006, [0016], [0039]. For example, Hackborn teaches that “the blades of grass 105 displayed in the area of the screen swiped by the user are **caused to bend or sway as if contacted by the user’s finger during the swipe**” and doing so “**in a manner that simulates frictional dragging of elements.**” APPLE-1004, [0008], [0028]. A POSITA would have understood or found obvious that Hackborn’s “simulate[d] frictional dragging” of grass movement results in the ceasing of grass movement (e.g., due to friction) after cessation of the user input. Additionally, Townsend teaches that “the simulation engine may provide realistic deceleration or elasticity behavior for application objects set in motion by user touch input,” e.g., “**if the user lifted the contacts while the object had a velocity in a particular direction, then inertia system continues sending**

events that indicate that the object is moving in that direction, **slowing down over time as the object decelerates,**” or decays. APPLE-1006, [0016], [0039]. A POSITA would have recognized that the motion of grass decays (decelerates) once the swipe ceases, per Townsend. Moreover, Hackborn describes “changing the manner in which the graphical wallpaper representation is displayed” by “animating motion in the graphical wallpaper representation in a manner that simulates frictional dragging of elements of the graphical wallpaper representation.” APPLE-1004, [0008], claims 4-9. A POSITA would have understood that, by simulating “frictional dragging,” the animating motion of the wallpaper representation (e.g., blades of grass) would appear to slow down, or decay, over time as a result of a force (e.g., simulated friction) that opposes the “force” provided by the user input (e.g., simulated touching of grass).

APPLE-1003, ¶121

Claims 5 and 27: Decay occurs after detecting cessation of the contact

5. The electronic device of claim 1, wherein the electronic device is configured such that a magnitude of the dynamic change in the appearance of the background of the first user interface decays following detection of a cessation of the first contact.



APPLE-1004, FIG. 1 (annotated)

Dr. Cockburn's Second Declaration

53. Additionally, for example, when the grass is at the outer edge of the touch screen (as illustrated below), and the user is swiping left-to-right, the decay will begin following detection of a cessation of the first contact and corresponding to the cessation of contact with the touch screen. APPLE-1003, ¶121.

54. Similarly, when the end of the touch occurs as the user's finger is removed from a blade of grass, decay will occur upon removal of the user's finger since the end of contact with the screen and with the blade of grass occurs at the same time, as illustrated below.

Claims 8 and 30: Dynamic changes in response to detection of cessation of the first input

8. The electronic device of claim 1, wherein the one or more programs includes instructions for:
in response to detecting a cessation of the first input,
ceasing the dynamically changing of the appearance of the background.

Townsend

[0039] FIG. 7 is a flow diagram that illustrates the processing of the simulation component of the inertia processing system, in one embodiment. In block 710, the component receives initial simulation parameters. For example, an application or manipulation processor may provide the final state of an application object when a user stopped touching the object. In block 720, the component initializes a simulation engine that performs calculations based on physics to determine the behavior of an object based on the parameters. For example, the simulation engine may provide realistic deceleration or elasticity behavior for application objects set in motion by user touch input. In block 730, the component receives a process call from the application. The application or other component drives the simulation process by repeatedly calling a processing function at regular intervals to move the simulation forward. The simulation component may also internally generate the timer.

[0044] In some embodiments, the inertia system does not provide additional movement for objects with movement below a predefined threshold. The threshold may be configurable by the application. For example, the inertia system may have a particular object linear or angular velocity below which the system will not continue movement of the object after the user releases the object. If the object is not moving very fast when the user releases it, the user may expect that the object will stay put and not continue to move. The threshold allows the application or author of the inertia system to determine the level of movement after manipulation that provides a good user experience.

Claims 8 and 30: Dynamic changes in response to detection of cessation of the first input

8. The electronic device of claim 1, wherein the one or more programs includes instructions for:
in response to detecting a cessation of the first input,
ceasing the dynamically changing of the appearance of the background.

Dr. Cockburn's First Declaration

[0039]. A POSITA would have understood, based on a basic level of Newtonian physics, that the decay in dynamic change to the background ultimately yields cessation of that dynamic change in time. Additionally, Townsend provides implementation details for such activity and a simulation engine/API to enable it for any GUI element (e.g., not limited to the grass in Hackborn). See APPLE-1006,[0016], [0034], [0039].

123. Townsend teaches that “the simulation engine may provide realistic deceleration or elasticity behavior for application objects set in motion by user touch input,” for example, “if the user lifted the contacts while the object had a velocity in a particular direction, then inertia system continues sending events that indicate that the object is moving in that direction, slowing down over time as the

object decelerates.” APPLE-1006, [0016], [0039]. In addition, Townsend teaches that “the inertia system does not provide additional movement for objects with movement below a predefined threshold,” and “the inertia system may have a particular object linear or angular velocity below which the system will not continue movement of the object after the user releases the object.” *Id.*, [0034].

Thus, a POSITA would have found obvious that the HWT device would be implemented such that the dynamic changes responsive to a user input (e.g., display of grass in “live” wallpaper moving in response to a user swipe input on a touch screen), as taught by Hackborn, would be implemented such that the

responsive display would reflect changes in the user movements, including changing velocity (e.g., slowing down or stopping) or changing direction.

Claims 9 and 31

9. The electronic device of claim 1, wherein the one or more programs includes instructions for:
in response to detecting a cessation of the first input, ceasing the dynamically changing of the appearance of the background and reverting to the display of the first user interface, including the background, on the display.

(Ground 1B). Additionally, Townsend teaches that the simulation component “uses techniques based on physics to simulate the behavior of the object for a period after the user releases the object.” APPLE-1006, [0021]. A POSITA would have understood that, according to physics, the blades of grass, at least in some instances, would return to their initial positions upon cessation of contact, and therefore the background reverts to its original state. For example, Hackborn describes that “[i]n some examples, graphics displayed on a portion of the

wallpaper may generally indicate the particular season, such as flowers for ‘spring,’ grass for ‘summer,’ piles of leaves for ‘autumn,’ or snow banks for ‘winter.’” APPLE-1004, [0043]. A POSITA would have understood that traces of the user’s swiping across “snow banks for ‘winter’” would *not* revert (e.g., because prints left in snow are enduring). In contrast, frictional dragging across grass for ‘spring’ would revert (e.g., due to the elastic nature of growing grass). This disclosure and a POSITA’s understanding of it is consistent with Townsend, which states that “[t]hose of ordinary skill in the art will recognize numerous well-known techniques for simulating the equivalent physical behavior of virtual objects in software.” APPLE-1006, [0021].

APPLE-1003, ¶124; Pet., 51

Claims 21 and 43

21. The electronic device of claim 1, wherein the electronic device is configured to, in response to detecting a cessation of the first input, revert to one of the plurality of images that is different from that which is displayed as the background prior to detecting the first input.

Townsend

[0039] FIG. 7 is a flow diagram that illustrates the processing of the simulation component of the inertia processing system, in one embodiment. In block 710, the component receives initial simulation parameters. For example, an application or manipulation processor may provide the final state of an application object when a user stopped touching the object. In block 720, the component initializes a simulation engine that performs calculations based on physics to determine the behavior of an object based on the parameters. For example, the simulation engine may provide realistic deceleration or elasticity behavior for application objects set in motion by user touch input. In block 730, the component receives a process call from the application. The application or other component drives the simulation process by repeatedly calling a processing function at regular intervals to move the simulation forward. The simulation component may also internally generate the timer.

Dr. Cockburn's First Declaration

1004, [0028]; APPLE-1006, [0039]. In particular, Hackborn teaches that the displayed grass moves as if contacted by a finger, and Townsend teaches that a “simulation engine performs calculations based on physics to determine the

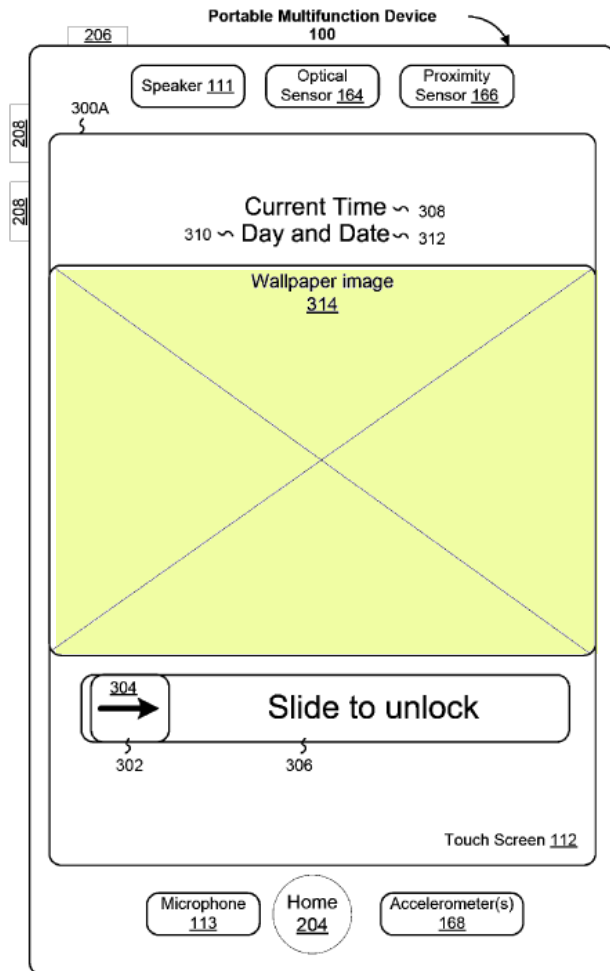
behavior of an object” so that the simulation object can provide “realistic deceleration or elasticity behavior for application objections set in motion by user touch input.” *Id.* A POSITA would have recognized and/or found obvious that, in some scenarios, after the gesture ceases, the scene will not revert to the original image. For example, when frictional dragging across dry grass for ‘summer’, the grass would have less elasticity than that of grass in spring, and therefore would not revert to the original image. APPLE-1004, [0043]. Indeed, a decision among two choices, ceasing with the grass returned to its original position and ceasing with the grass in a position other than the returned position, is a design choice well within the grasp of a POSITA. Moreover, the ’755 Patent included no discussion

Issue 3

The Dependent Claims Are Obvious

A POSITA would have been motivated to combine Hackborn-Westerman and Jobs (Ground 1D)

Jobs



Dr. Cockburn's First Declaration

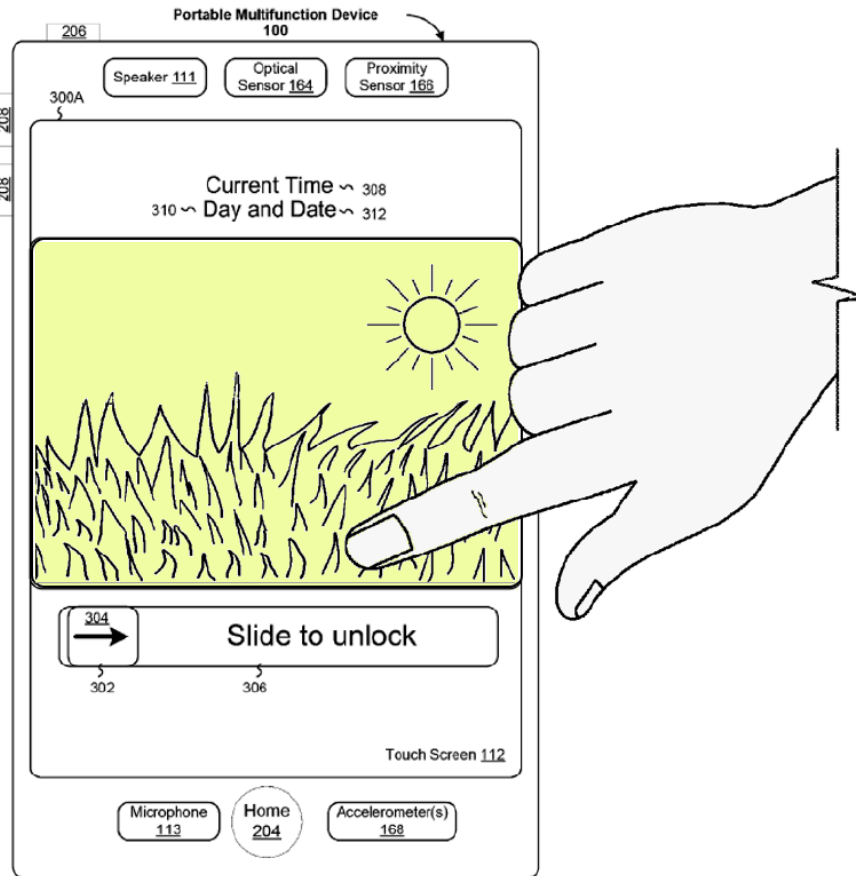
139. It would have been obvious to a POSITA for the Hackborn-Westerman device to further include a lock screen with a wallpaper image, as taught in Jobs, and as was well known by the Critical Date. For example, Chaudhri explains that a “problem associated with using touch screens on portable device,” such as Hackborn’s device, “is the unintentional activation or deactivation of functions due to unintentional contact with the touch screen.” APPLE-1010, [0004]. As such, “portable devices, touch screens on such devices, and/or

applications running on such devices may be locked upon satisfaction of predefined lock conditions, such as upon entering an active call, after a predetermined time of idleness has elapsed, or upon manual locking by a user.” *Id.*; see *id.*, FIGS. 4A-4B, 5A-5B. Moreover, it would have been further obvious and desirable to a POSITA to provide means by which “unintentional activation or deactivation of functions” can be avoided, such Jobs’ lock screen with a wallpaper image.

APPLE-1003, ¶ 139; APPLE-1009, FIG. 3A

A POSITA would have been motivated to combine Hackborn-Westerman with Jobs (Ground 1D)

Composite Image



Dr. Cockburn's First Declaration

140. Additionally, given Hackborn's teaching of using live wallpaper, it would have been obvious to a POSITA to implement the Hackborn-Westerman-Jobs device to include a lock screen with a wallpaper, as taught by Jobs, such that the lock screen wallpaper is "live" like the unlocked screen, as taught by Hackborn. Such a modification would "provide updated information, altered aesthetics, a change of theme, environment or pace, or a diversion from the status quo," as taught by Hackborn. APPLE-1004, [0026]. In this way, "the [lock screen] wallpaper need not be a static image, but can be active and may execute code so as to produce interesting and compelling user interface examples," as Hackborn teaches. APPLE-1004, [0036]. This would have motivated a POSITA to implement the Hackborn-Westerman-Jobs device to include a lock screen with live wallpaper, thereby improving the device by preventing "unintentional activation or deactivation of functions" while still producing an "interesting and compelling user interface" in the lock screen. In the Hackborn-Westerman-Jobs device (as shown below), a user could lock the device with a lock screen having a wallpaper, as taught by Jobs, and the wallpaper is a "live" wallpaper, as taught by Hackborn, such that user touch on the background (grass) causes the grass to move but does not cause foreground objects, such as the "unlock image 302" to change appearance. APPLE-1004, [0028], FIG. 1; APPLE-1009, FIG. 3A. Moreover,

Using live wallpaper does not alter the benefits Jobs affords

Rebuttal to: “[I]ntegrating Hackborn’s constantly-responsive live wallpaper into this context is not an obvious or ‘straightforward’ step as Petitioner contends—it cuts against the lock screen’s primary purpose...” Sur-reply, 14

Hackborn

[0026] A general home screen layout is shown in FIG. 1, where a computing device 100 (e.g., a smart phone, tablet device, laptop device, pad device, personal digital assistant, electronic reading device, or the like) with a display screen presents a display, but in this example, the wallpaper 104 is live. An example of such a state of the wallpaper is shown in the figure by the wallpaper’s 104 reaction to a user input on a screen of the device 100. In particular, in this example, the wallpaper shows grass 105 below a sunny sky 107, and may ordinarily look like a traditional wallpaper in the form of a static image. However, the wallpaper 104 in this example may instead be an object that includes executable code so that the wallpaper may be active, and in this example may respond to inputs from a user of the computing device or otherwise change in response to received information from the user or from another computing device. The executable code of the object may cause the wallpaper 104 to change in appearance, perhaps at the direction of a display controller, following receipt of information, and such change in appearance may provide updated information, altered aesthetics, a change of theme, environment or pace, or a diversion from the status quo.

Jobs

[0177] In some embodiments, the device detects contact with the touch-sensitive display (e.g., a user’s finger making contact on or near the unlock image 302) while the device is in a user-interface lock state. The device moves the unlock image 302 in accordance with the contact. The device transitions to a user-interface unlock state if the detected contact corresponds to a predefined gesture, such as moving the unlock image across channel 306. Conversely, the device maintains the user-interface lock state if the detected contact does not correspond to the predefined gesture. This process saves battery power by ensuring that the device is not accidentally awakened. This process is easy for users to perform, in part because of the visual cue(s) provided on the touch screen.

Claims 18 and 39: The Petition clearly established that the “unlock image” is a foreground object

Patent Owner's Response

Claim 1 recites “one or more foreground objects.” In Ground 1A, Petitioner pointed to icon 102, gadget 106, icons 108, and application 110 of Hackborn as foreground objects because they “exist on a layer in front of the wallpaper 104. Pet., 18–19. Petitioner does not identify any foreground objects in the lock screen interface of Jobs, nor does Petitioner indicate that the foreground objects identified in Hackborn would exist in the lock screen interface of Jobs. EX2005, ¶105.

Petition

interface” in the lock screen. APPLE-1003, ¶140. In the HWJD (as shown below), a user could lock the device with a lock screen having a wallpaper, per Jobs, and the wallpaper is a “live” wallpaper, per Hackborn, such that user touch on the background (grass) causes the grass to move but does not cause foreground objects, such as the “unlock image 302” to change appearance. *Id.*; APPLE-1004, [0028], FIG. 1; APPLE-1009, FIG. 3A. Moreover, because swiping inputs were

Petitioner's Reply

HWJD. Pet., 18-19; APPLE-1031, ¶69. Contrary to Smith's assertion (POR, 48), Petitioner clearly explains that, in HWJD, example foreground objects include the “unlock image.” Pet., 59. APPLE-1031, ¶69.

Claims 18 and 39: The mapping of claim 1 is not inconsistent with claim 18

Patent Owner's Response

Claim 1 recites “one or more foreground objects.” In Ground 1A, Petitioner pointed to icon 102, gadget 106, icons 108, and application 110 of Hackborn as foreground objects because they “exist on a layer in front of the wallpaper 104. Pet., 18–19. Petitioner does not identify any foreground objects in the lock screen interface of Jobs, nor does Petitioner indicate that the foreground objects identified in Hackborn would exist in the lock screen interface of Jobs. EX2005, ¶105.

Petitioner's Reply

Additionally, HWJD's user interface (UI) includes a home screen and a lock screen. Per Hackborn, HWJD includes “a home screen [that] includes a background, which may ... include an image in the form of wallpaper.” Pet., §IV.A.4[1.g] (citing APPLE-1004, [0025]). “HWJD has a lock screen as a first interface and the lock screen includes a wallpaper, per Jobs, which is ‘live,’ per Hackborn.” Pet., §IV.D.3[18], (internal citations omitted). Since HWJD's UI (*first user interface*) includes two “screens” (home/lock screens), the Petition's treatment of claims 1 and 18 is consistent. APPLE-1031, ¶¶69-71. Thus, HWJD's UI, including both home and lock screens, renders obvious “the first user interface includes a lock screen.”

POR, 48; Reply, 22

19 and 40: The Petition describes Hackborn-Westerman-Jobs' "background" and "foreground objects"

Patent Owner's Response

as discussed in Section V.D.2 for claims 18 and 39 above, in pointing to the unlocked lock screen interface of Jobs instead of the interface of Hackborn, Petitioner fails to show that the Hackborn-Westerman-Jobs combination discloses the required "background with an appearance" and "one or more foreground objects" of claim 19. Petitioner relies solely on the same arguments for claim 40, so it is non-obvious for the same reasons.

Petition

interface" in the lock screen. APPLE-1003, ¶140. In the HWJD (as shown below), a user could lock the device with a lock screen having a wallpaper, per Jobs, and the wallpaper is a "live" wallpaper, per Hackborn, such that user touch on the background (grass) causes the grass to move but does not cause foreground objects, such as the "unlock image 302" to change appearance. *Id.*; APPLE-1 [0028], FIG. 1; APPLE-1009, FIG. 3A. Moreover, because swiping inputs w

Petitioner's Reply

As discussed in §V.B, Smith's argument that Hackborn-Westerman-Jobs fails to disclose a "background with an appearance" and "one or more foreground objects" (POR, 50-51) willfully ignores the combination's background with an appearance (live wallpaper) and the foreground objects of HWJD. Pet., 59; APPLE-1031, ¶73.

Issue 4

The Combination of Hackborn2, Westerman,
and Jobs Renders the Challenged Claims
Obvious (Ground 2)

Issue 4A

Jobs' Scrolling Makes Visible the Obscured
Messages of Hackborn

Hackborn teaches live wallpaper with social media feeds

Hackborn

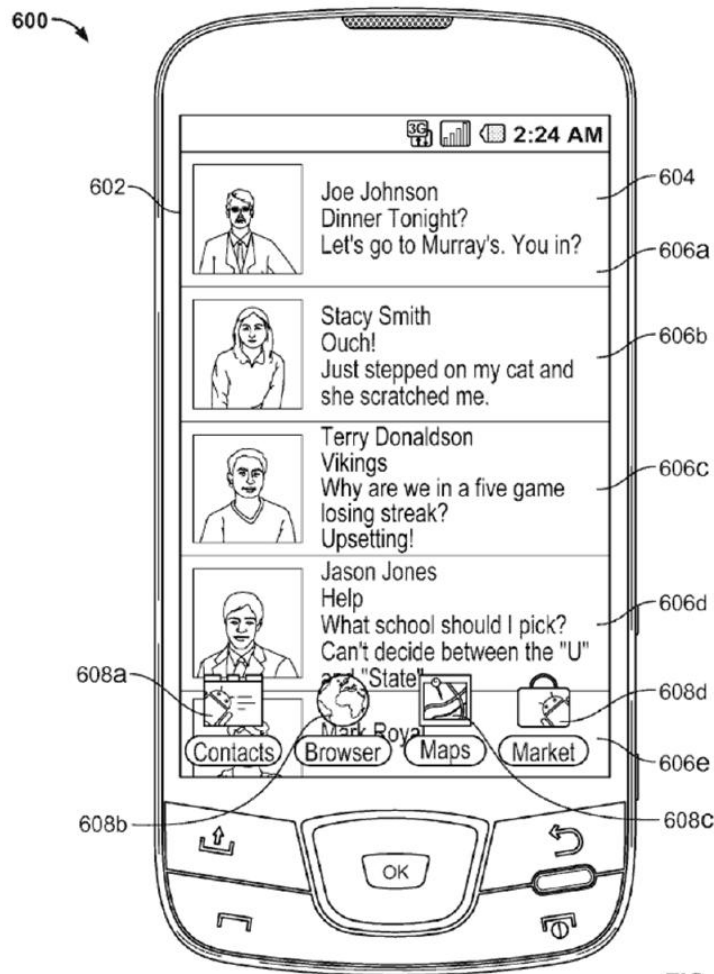


FIG. 6

[0046] FIG. 6 shows yet another example of active content being produced as live wallpaper on a computing device 600. A display 602 of the device 600 displays the live wallpaper 604, which includes information fetched or received from one or more social services on the Internet. The information includes posts or tweets 606a-e from a social networking site (e.g., Google Buzz, Facebook, MySpace, Flickr, Twitter, and the like) posted by friends, acquaintances, group members or contacts of the user, for example and the information may be received by the device from a web server associated with the networking site. The depicted posts 606 include an image in this example, but in other examples the posts 606a-e may omit the image. The posts 606a-e include textual information and an indication of the poster's identity.

Hackborn explains that older posts in the live wallpaper may scroll out of view or be obscured

Hackborn

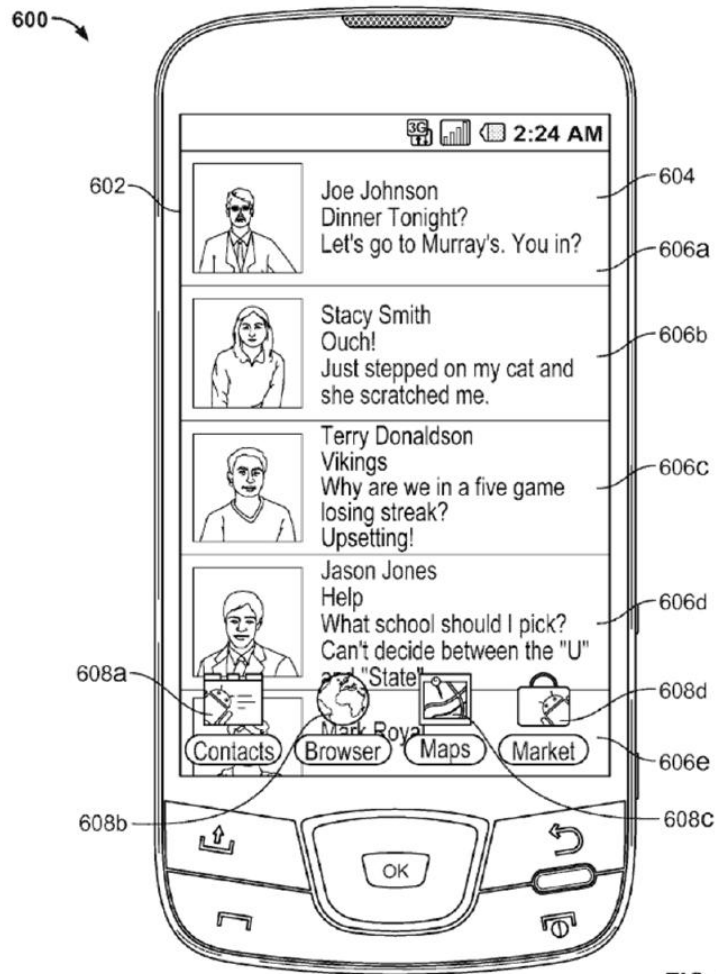


FIG. 6

[0047] The posts 606a-e may be displayed in various manners. For example, as new information is received, already displayed posts may be scrolled (e.g., downward) and the new post may be displayed in a predetermined position. For example, post 606a in the present example may be the most recently received message, and may be displayed near the top of the display 602. When post 606a is received, the previously displayed posts (e.g., posts 606b, 606c, 606d, 606e) may be shifted downward. As can be seen, post 606e is only partially displayed near the bottom of the display 602. Icons 608a-d are shown on a display layer above the wallpaper 604. While

Jobs addresses content that has scrolled off screen through manual scrolling

Jobs

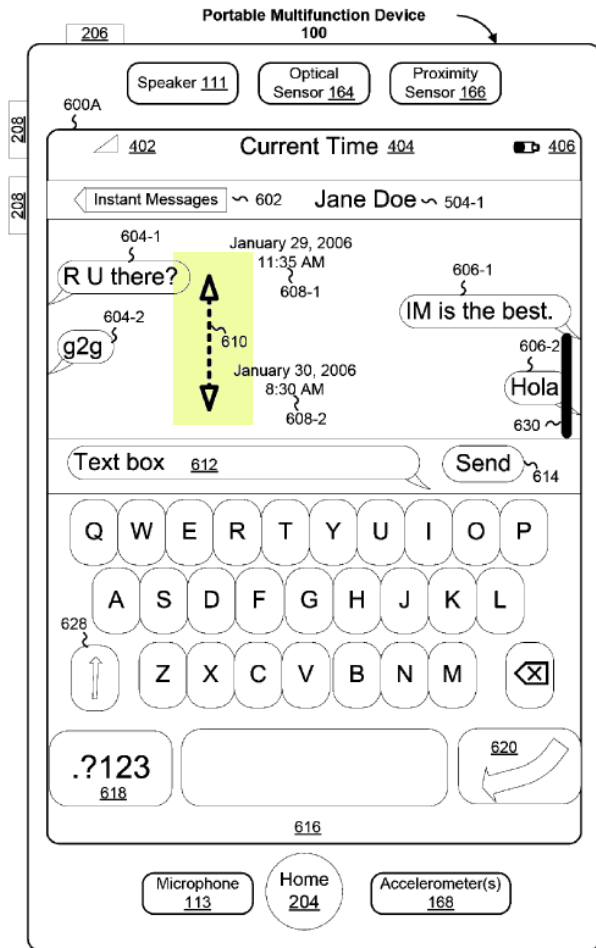


Figure 6A

[0243] In some embodiments, a user can scroll through the message conversation (comprised of messages 604 and 606) by applying a vertical swipe gesture 610 to the area displaying the conversation. In some embodiments, a vertically downward gesture scrolls the conversation downward, thereby showing older messages in the conversation. In some embodiments, a vertically upward gesture scrolls the conversation upward, thereby showing newer, more recent messages in the conversation. In some embodiments, as noted above, the last message in the conversation (e.g., 606-2) is displayed in the list of instant messages 500 (e.g., 506-1).

Jobs' solution offers a more convenient solution than requiring use of a dedicated application

Dr. Cockburn's Deposition

1 If these are Facebook posts and a user
2 missed a message or a post, they could also simply
3 open Facebook. Correct?
4 A. They could. I mean, this -- if they are
5 just wanting to keep track of what messages have
6 been received and stay up to date, they may not
7 want to open up the application. It could be
8 inconvenient. Whereas a quick glance might tell
9 them if an important message from a boss or
10 colleague that they had been awaiting has arrived
11 or not.

22 A. I think it stands for itself. If the user
1 is looking at a message or has just started
2 looking at a message, sees that it's something
3 that they are interested in and would like to see
4 more information about, and then it vanishes off
5 the display because it's been scrolled out by a
6 new arriving message, a person of skill in the art
7 would absolutely understand that this would be an
8 infuriating experience.

Issue 4B

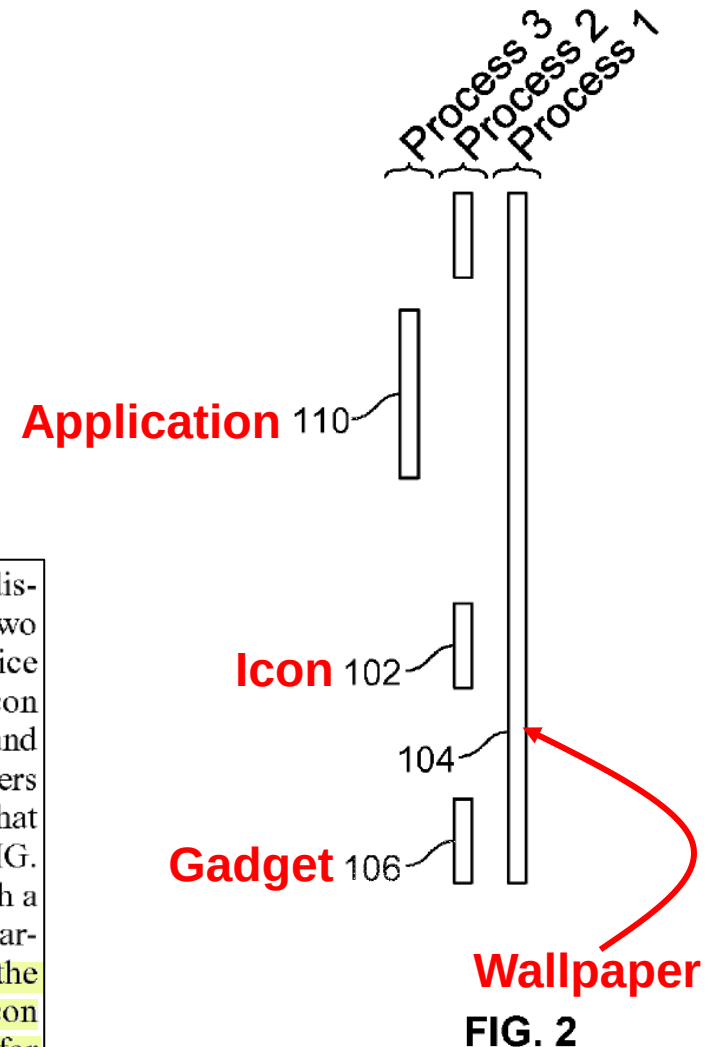
Hackborn's Wallpaper is in the Background and
Its System Would Work with Manual Scrolling

Hackborn's live wallpaper is in the background

Hackborn

[0005] The various objects that are part of a home screen may be segregated into various display layers, with some layers appearing in front of other layers. For example, when a user launches an application, the window for the application may become active, and move visually in front of other objects that were previously displayed on a home screen. Also, a home page generally has a back-most layer that fills substantially all of the display and acts as a background, or as wallpaper when the background is formed from a graphical image. Icons that can be selected by a user to launch or maximize applications may be displayed on a visual layer in front of the background but behind any open applications.

[0029] FIG. 2 shows a conceptual side diagram of the display shown on device 100 of FIG. 1. FIG. 2 shows two particularly relevant points about the display shown on device 100. First, the wallpaper 104, icons or gadgets such as icon 102 and gadget 106, and applications such as open and executing chat application 110, are on different visual layers of the display shown on device 100. This does not mean that the objects are physically arranged as they are shown in FIG. 2. Rather, it refers to their relative priority for display with a display manager or display controller on the device. In particular, when an element of the wallpaper 104 overlaps in the x-y dimension (that is, in the plane of the display) with an icon 102 or gadget 106, the icon or gadget will receive priority for display so that it looks to the user like it is positioned closer to the user than is the wallpaper.



Hackborn's language regarding changing display position/status of the wallpaper is permissive, not mandatory

Hackborn

[0048] In general and with regard to any of the displayed examples of live wallpaper in the figures, the wallpaper or a portion of the wallpaper be selected, or a particular input may be provided, to cause the wallpaper or portion thereof to assume a more prominent display position or status. For example, display planes may be adjusted based on certain inputs or information received (e.g., from another computing device). In some examples, selection of the wallpaper or a portion of the wallpaper, or provision of a predefined input, may cause an application associated with the wallpaper or a portion of the wallpaper to launch or become active, and may allow the user to work with an aspect of the wallpaper or associated with the displayed wallpaper. For example, the user may view Terry Donaldson's post 606c and select the post to provide an opinion to the question posed. The user may respond that the coaching is subpar and that a change is warranted, as by speaking or typing such a response, and a message may be sent from the device 600 to another computing device (e.g., by wireless transmission). For example, if message 606c represents a Facebook post, selection of message 606c may launch a Facebook application on device 600, which may permit the user to respond to Donaldson's post 606c.

APPLE-1004, [0048]

Hackborn's system works with scrolling in its live wallpaper

Patent Owner's Response

EX1009, [0243]; EX2005, ¶117. However, Hackborn's wallpaper—when in a background layer—would be at least partially covered by icons, widgets, or applications, leaving limited screen real estate for reading a manually scrollable feed. EX1004, FIG. 6. Further, a POSITA would understand that implementing Jobs' continuous scrolling in a background context would demand significant modification of Hackborn's wallpaper process to manage historical post data storage, collision with home-screen icons, and continuous input detection to allow up-or-down movement behind foreground objects. EX2005, ¶117. Nothing in the references or Petitioner's arguments explains why a POSITA would implement manual scrolling despite these complexities, much less do so with Hackborn's wallpaper remaining in a background layer during that scrolling. *Id.*

POR, 58-59; APPLE-1004, [0057],
FIG. 6; Pet., 66

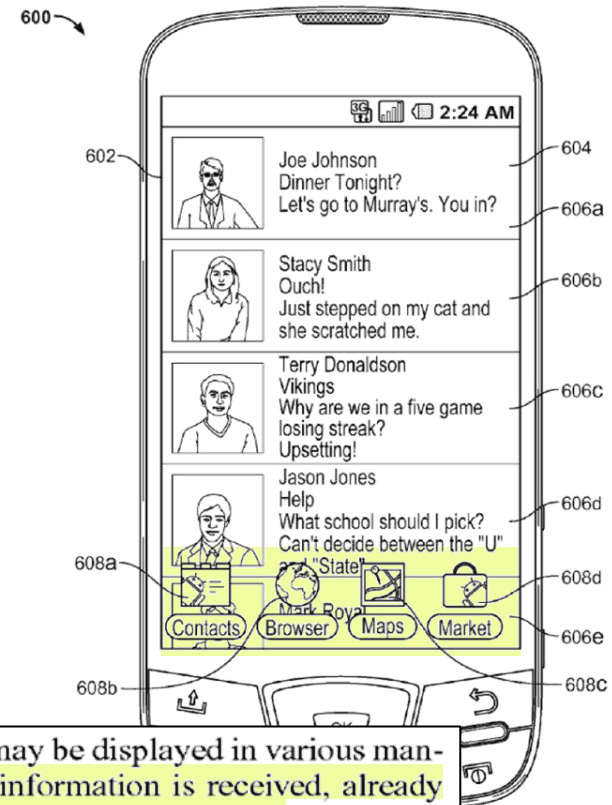


FIG. 6

Hackborn

[0047] The posts 606a-e may be displayed in various manners. For example, as new information is received, already displayed posts may be scrolled (e.g., downward) and the new post may be displayed in a predetermined position. For example, post 606a in the present example may be the most recently received message, and may be displayed near the top of the display 602. When post 606a is received, the previously displayed posts (e.g., posts 606b, 606c, 606d, 606e) may be shifted downward. As can be seen, post 606e is only partially displayed near the bottom of the display 602. Icons 608a-d are shown on a display layer above the wallpaper 604. While

A POSITA would have found including manual scrolling obvious

Petition

Given that Hackborn teaches (1) receiving user input on the background (live wallpaper), (2) causing the background to change appearance responsive to user input, (3) causing a portion of the background wallpaper to assume a more prominent display position, and (4) scrolling in new images or posts, a POSITA would have found it obvious for the device to receive a user input on the live wallpaper 604 of Figure 4, determine that input to be a swipe (per Westerman), and in response, cause the live wallpaper 604 to change its appearance by scrolling through the social media posts—or more particularly, scrolling to older posts (e.g., permit a user to view posts that were automatically scrolled out of view). APPLE-1003, ¶148. Indeed, a POSITA reviewing Hackborn's teaching of scrolling social media posts in relation to Figure 6 would have found it obvious to provide a way to view older posts (e.g., partially-displayed post 606e) that s/he may not have already read before the posts were moved downward and less (or not) visible. *Id.* Specifically, a POSITA would have found it obvious to allow for manual scrolling through the posts by the user. *Id.*

Issue 4C

Westerman's Teachings Are Closely Related to
Hackborn and Jobs

Hackborn, Westerman, and Jobs each discuss taps/swipes on a touch sensitive surface

Hackborn

[0028] In the example shown in FIG. 1, a user is swiping her finger across a touch-sensitive display screen of the computing device. In response, the blades of grass 105 displayed in the area of the screen swiped by the user are caused to bend or sway as if contacted by the user's finger during the swipe. Other examples of live wallpaper may not be immediately responsive to user input on the screen of the device. Rather, some examples of live wallpaper may be displayed and updated so that an appearance of the wallpaper is changed, but may become responsive to direct user input only after a predetermined user input is received by the device. Examples of predetermined user inputs that may be received by the device to cause the live wallpaper to become responsive to user input can include agitating or shaking the device, selecting an icon, button (not shown in FIG. 1), or area of the display screen (e.g., a corner of the screen) reserved for activating the live wallpaper, making a particular movement on or across the display screen, tapping on the display screen a predetermined number of times within a predetermined time period, and others. Thus, in a first mode, user input on a touchscreen may

Pet., 9-14, 30; Reply, 30; APPLE-1004, [0028]; APPLE, 1005, [0019], [0061]; APPLE-1009, [0164]

Westerman

direction ($D[i].y$), and over what time ($T[i]$). The total distance moved can then be compared to a minimum threshold of movement used to determine whether the touch event is a tap or a swipe (block 1007). If there is little motion, i.e., less than the threshold, the event is interpreted as a key tap, and the system updates the key choice that corresponds to the location of the finger (block 1008).

Jobs

[0164] FIG. 2 illustrates a portable multifunction device 100 having a touch screen 112 in accordance with some embodiments. The touch screen may display one or more graphics within user interface (UI) 200. In this embodiment, as well as others described below, a user may select one or more of the graphics by making contact or touching the graphics, for example, with one or more fingers 202 (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the contact may include a gesture, such as one or more taps, one or more swipes (from left to right, right to left, upward and/or downward) and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with the device 100. In some embodiments, inadvertent contact with a graphic may not select the graphic. For example, a swipe gesture that sweeps over an application icon may not select the corresponding application when the gesture corresponding to selection is a tap.

Smith improperly characterizes Jobs' teaching of distinguishing taps and swipes

Patent Owner's Response

result in less scrolling of the feed. *Id.* Indeed, Jobs teaches the receipt of “taps” but distinguishes them spatially rather than via thresholds. EX2005, ¶120; *compare* EX1009, [0232], [0237], [0239], [0240], [0241] (discussing multiple examples of “a finger tap” on various icons *outside* the area displaying the conversation), *with id.*, [0243] (discussing “swipe gesture 610 *to the area displaying the conversation*”).

Petitioner's Reply

60. In Jobs, a tap and swipe can occur at the same location (on an application icon) causing different actions, so spatial distinctions are not sufficient. APPLE-1009, [0164]; APPLE-1031; ¶81. Given Hackborn's silence regarding implementation

Jobs

[0164] FIG. 2 illustrates a portable multifunction device **100** having a touch screen **112** in accordance with some embodiments. The touch screen may display one or more graphics within user interface (UI) **200**. In this embodiment, as well as others described below, a user may select one or more of the graphics by making contact or touching the graphics, for example, with one or more fingers **202** (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the contact may include a gesture, such as one or more taps, one or more swipes (from left to right, right to left, upward and/or downward) and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with the device **100**. In some embodiments, inadvertent contact with a graphic may not select the graphic. For example, a swipe gesture that sweeps over an application icon may not select the corresponding application when the gesture corresponding to selection is a tap.

POR, 60; Reply, 30; APPLE-1009, [0164]

Claims 1 and 23: Jobs relies on a distinction between a tap and a swipe to initiate scrolling

Patent Owner's Response

Petitioner contends that “a POSITA would have found it obvious to implement the H2WJD such that a swiping gesture, as determined using the methods of Westerman, on the wallpaper 604 of Hackborn would cause the images/posts of the wallpaper 604 to scroll, per Jobs.” Pet., 74. However, there is no indication in Hackborn, Jobs, or Westerman that the feed-based wallpaper will only update its display once a certain threshold is met. EX2005, ¶122.

Hackborn explains that social-media updates shift older posts when new content arrives; it does not discuss rejecting inputs below a threshold. EX1004, [0047]; EX2005, ¶123. Jobs discloses the ability to vertically scroll through messages in an instant messaging application. EX1009, [0243], [1138]. While this might conceptually overlap with Hackborn’s notion of a feed, neither reference suggests gating display changes in the feed behind a threshold. EX2005, ¶123. The vertical scroll of Jobs is not a binary determination—it’s continuous. *Id.*; EX1009, [1138]. A POSITA would understand that Jobs does not rely on the distinction between a swipe and a tap to initiate its vertical scroll because the amount of scrolling corresponds to the magnitude of the contact movement. EX2005, ¶123. In

Petition

As discussed in §IV.E.1, a POSITA would have further found obvious that the H2WJD would be implemented such that if the user input “exceeds the minimum swipe length threshold” and/or if the “initial velocity at touchdown” is greater than a velocity threshold, the input is “interpreted as a swipe event.” APPLE-1005, [0018]-[0019], [0060]-[0064]; APPLE-1003, ¶170. Thus, the detection of the user input would be in accordance with a determination that the input has a magnitude (e.g., path length) that is above a threshold, or predetermined minimum. *Id.* For example, if the magnitude of the touch is large enough to be registered as a swipe because its “path has been determined to exceed the minimum length threshold for a swipe gesture,” then images that are part of the live wallpaper 404 are caused to scroll and show new images. APPLE-1003, ¶170; APPLE-1005, [0040]-[0041], [0064].

Reply

its vertical scroll” (POR, 62-63), gesture determination is required since swipes, not taps, cause scrolling. APPLE-1031, ¶82. Therefore, since swipes are distinguished based on a magnitude exceeding a threshold, the dynamic changes of appearance of the background (feed scrolling) occurs when the threshold is exceeded (i.e., swipe detected). APPLE-1003, ¶¶52-53; APPLE-1031, ¶¶82-83.

POR, 62; Pet., 74-76, Reply, 31

Claims 7 and 29: The live wallpaper is unresponsive to an input below a threshold (e.g., a tap)

7. The electronic device of claim 1, wherein the one or more programs include instructions for:
in response to detecting the first input by the first contact on the touch-sensitive surface, in accordance with a determination that the first input does not have a magnitude above the threshold, maintaining the appearance of the background of the first user interface.

Reply

As discussed regarding Ground 1A, H2WJD scrolls the feed responsive to a swipe, not a tap. *See* §§II.A.3-4; Pet., 79-80. Indeed, “Hackborn teaches the use of live wallpapers that can be caused to change their appearance *in response to a user input (e.g., swipe)*.” APPLE-1003, ¶145. However, a tap, differentiated from a swipe

in H2WJD using thresholds, “does not suggest the same scrolling output” on the wallpaper feed. APPLE-1003, ¶176; APPLE-1031, ¶84; *see* POR, 63.

Claim 28

28. The method of claim 23, wherein the dynamic change in the appearance of the background of the first user interface is independent of the location that corresponds to the background.

Dr. Cockburn's First Declaration

87. In H2WJD, the dynamic change in the appearance of the background is independent of the location that corresponds to the background. Pet., §§IV.E.5, IV.E.20; APPLE-1003, ¶¶175, 204. For example, Jobs states the movement of a finger “on the touch screen display” causes the “list of items displayed on the touch screen display [to] scroll[.]” APPLE-1009, [1137]-[1138]. Thus, a POSITA

would have understood that the scrolling gesture can occur at any location on the live wallpaper. APPLE-1009, [1137]-[1138]; APPLE-1003, ¶¶175, 204.

88. Smith mischaracterizes Petitioner’s position by stating that “Petitioner admits [the gesture] occurs ‘within the message list’ that represents the dynamic change of the background.” POR, 66. In fact, as described above, Petitioner made no such admission, stating instead that while a gesture in the message list is an option, the gesture can occur *at any location* on the live wallpaper and is thus independent of the location that corresponds to the background. Pet. §IV.E.5[6]; APPLE-1003, ¶175.

Petition

1003, ¶175. Jobs states that “movement of [a finger] is detected on or near the displayed portion of the list” of Figure 6A or simply “on the touch screen display.” APPLE-1009, [1137], FIG. 6A. “In response to detecting the movement, the list of items displayed on the touch screen display is scrolled.” *Id.*, [1138]. Thus, a POSITA would have understood that the scrolling gesture 610 in Figure 6A of Jobs can occur at *any* location within the message list, and similarly that the scrolling gesture (or a “focus selector,” to the extent one exists in the H2WJD) in the H2WJD can occur at any location on the live wallpaper. APPLE-1009, [1137]-[1138]; APPLE-1003, ¶175.

APPLE-1031, ¶¶ 87-88; Reply, 32-33; Pet., 79, 89