



- (51) **International Patent Classification:**
G06F 3/048 (2006.01) *G06F 3/14* (2006.01)
- (21) **International Application Number:**
PCT/US2012/064259
- (22) **International Filing Date:**
9 November 2012 (09.11.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/295,376 14 November 2011 (14.11.2011) US
- (71) **Applicant (for all designated States except US):** MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) **Inventors:** ALLYN, Barry Christopher; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). ADAMS, Tyler Robert; c/o Microsoft Corporation, LCA - Interna-

tional Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

[Continued on next page]

(54) **Title:** CROSS WINDOW ANIMATION

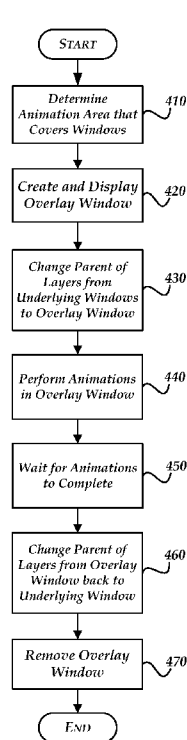


Fig. 4

(57) **Abstract:** Animations may cross different windows. These cross window animations may include interwindow animations and/or window transition animations. Interwindow animations are used to animate elements across windows. Window transitions are used to animate moving/resizing/showing/hiding windows simultaneously. To perform these animations, an overlay window covers the animation area that includes at least a portion of two different windows. Layers may be used to represent images or visual elements that can be moved as part of an animation. These layers may be placed in the animation overlay window to move visual elements across the underlying windows. Supported underlying windows are directed to draw content directly to the overlay window during the animation. A picture of the underlying window is drawn to the animation overlay window when the underlying window is not-supported such that the underlying window is included within the animation. When the animation is complete, the animation overlay window is removed.



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

CROSS WINDOW ANIMATION

BACKGROUND

[0001] Many applications use animations to enrich a user experience. For example, animations may be performed in response to a user selecting a user interface element, advancing to a next slide, opening/closing a document, changing a view, and the like. These animations are performed within a window of the application. For example, one animation may be performed within a user interface window and another animation may be performed within a document window. Often it is desirable to create animations that move, resize, or cross these windows. These types of animations, however, can be difficult.

SUMMARY

[0002] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0003] Animations may cross different windows. These cross window animations may include interwindow animations and/or window transition animations. Interwindow animations are used to animate elements across windows. Window transitions are used to use animation to move/resize/show/hide windows simultaneously. To perform these animations, an overlay window is created to cover the animation area that includes at least a portion of different windows. Layers may be used to represent images or visual elements that can be moved as part of an animation. These layers may be placed in the animation overlay window to move visual elements across the underlying windows. Supported underlying windows are directed to draw content directly to the overlay window during a time of the animation. A picture of the underlying window is drawn to the animation overlay window when the underlying window is not-supported such that the underlying window is included within the animation. When the animation is complete, the animation overlay window may be removed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIGURE 1 illustrates an exemplary computing device;

[0005] FIGURE 2 illustrates an exemplary system for performing cross window animations;

[0006] FIGURE 3 shows exemplary displays showing an interwindow animation and a windows transition animation;

[0007] FIGURE 4 shows a process for cross window animation using an overlay window to animate a layer between different windows;

[0008] FIGURE 5 shows a process for cross window animation including moving/resizing different windows using an overlay window;

5 [0009] FIGURE 6 illustrates a system architecture used in cross window animations; and

[0010] FIGURES 7-10 show exemplary displays illustrating cross window animations.

DETAILED DESCRIPTION

10 [0011] Referring now to the drawings, in which like numerals represent like elements, various embodiments will be described. In particular, FIGURE 1 and the corresponding discussion are intended to provide a brief, general description of a suitable computing environment in which embodiments may be implemented.

[0012] Generally, program modules include routines, programs, components, data
15 structures, and other types of structures that perform particular tasks or implement particular abstract data types. Other computer system configurations may also be used, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. Distributed computing environments may also be used where tasks are performed by
20 remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0013] Referring now to FIGURE 1, an illustrative computer architecture for a computer 100 utilized in the various embodiments will be described. The computer
25 architecture shown in FIGURE 1 may be configured as a server computing device, a desktop computing device, a mobile computing device (e.g. smartphone, notebook, tablet ...) and includes a central processing unit 5 ("CPU"), a system memory 7, including a random access memory 9 ("RAM") and a read-only memory ("ROM") 10, and a system bus 12 that couples the memory to the central processing unit ("CPU") 5.

30 [0014] A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the ROM 10. The computer 100 further includes a mass storage device 14 for storing an operating system 16, application(s) 24, presentation(s)/document(s) 27, and other program

modules, such as Web browser 25, and animation manager 26, which will be described in greater detail below.

[0015] The mass storage device 14 is connected to the CPU 5 through a mass storage controller (not shown) connected to the bus 12. The mass storage device 14 and its associated computer-readable media provide non-volatile storage for the computer 100. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, the computer-readable media can be any available media that can be accessed by the computer 100.

[0016] By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, Erasable Programmable Read Only Memory (“EPROM”), Electrically Erasable Programmable Read Only Memory (“EEPROM”), flash memory or other solid state memory technology, CD-ROM, digital versatile disks (“DVD”), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 100.

[0017] According to various embodiments, computer 100 may operate in a networked environment using logical connections to remote computers through a network 18, such as the Internet. The computer 100 may connect to the network 18 through a network interface unit 20 connected to the bus 12. The network connection may be wireless and/or wired. The network interface unit 20 may also be utilized to connect to other types of networks and remote computer systems. The computer 100 may also include an input/output controller 22 for receiving and processing input from a number of other devices, such as a touch input device. The touch input device may utilize any technology that allows single/multi-touch input to be recognized (touching/non-touching). For example, the technologies may include, but are not limited to: heat, finger pressure, high capture rate cameras, infrared light, optic capture, tuned electromagnetic induction, ultrasonic receivers, transducer microphones, laser rangefinders, shadow capture, and the like. According to an embodiment, the touch input device may be configured to detect near-touches (i.e. within some distance of the touch input device but not physically touching the touch input device). The touch input device may also act as a display 28.

The input/output controller 22 may also provide output to one or more display screens, a printer, or other type of output device.

[0018] A camera and/or some other sensing device may be operative to record one or more users and capture motions and/or gestures made by users of a computing device.

5 Sensing device may be further operative to capture spoken words, such as by a microphone and/or capture other inputs from a user such as by a keyboard and/or mouse (not pictured). The sensing device may comprise any motion detection device capable of detecting the movement of a user. For example, a camera may comprise a MICROSOFT KINECT® motion capture device comprising a plurality of cameras and a plurality of
10 microphones.

[0019] Embodiments of the invention may be practiced via a system-on-a-chip (SOC) where each or many of the components/processes illustrated in the FIGURES may be integrated onto a single integrated circuit. Such a SOC device may include one or more processing units, graphics units, communications units, system virtualization units and
15 various application functionality all of which are integrated (or “burned”) onto the chip substrate as a single integrated circuit. When operating via a SOC, all/some of the functionality, described herein, may be integrated with other components of the computer 100 on the single integrated circuit (chip).

[0020] As mentioned briefly above, a number of program modules and data files may
20 be stored in the mass storage device 14 and RAM 9 of the computer 100, including an operating system 16 suitable for controlling the operation of a networked computer, such as the WINDOWS SERVER®, WINDOWS 7® operating systems from MICROSOFT CORPORATION of Redmond, Washington.

[0021] The mass storage device 14 and RAM 9 may also store one or more program
25 modules. In particular, the mass storage device 14 and the RAM 9 may store one or more applications, such as an animation manager 26, productivity applications 24 (e.g. a presentation application such as MICROSOFT POWERPOINT, a word-processing application such as MICROSOFT WORD, a messaging application such as MICROSOFT OUTLOOK, a spreadsheet application such as MICROSOFT EXCEL, and the like), and
30 may store one or more Web browsers 25. The Web browser 25 is operative to request, receive, render, and provide interactivity with electronic content, such as Web pages, videos, documents, and the like. According to an embodiment, the Web browser comprises the INTERNET EXPLORER Web browser application program from MICROSOFT CORPORATION.

[0022] Animation manager 26 may be located on a client device and/or on a server device (e.g. within service 19). Animation manager 26 may be configured as an application/process and/or as part of a cloud based multi-tenant service that provides resources (e.g. services, data ...) to different tenants (e.g. MICROSOFT OFFICE 365, MICROSOFT WEB APPS, MICROSOFT SHAREPOINT ONLINE).

[0023] Generally, animation manager 26 is configured to perform animations that cross different windows. These cross window animations may include interwindow animations and/or window transition animations. Interwindow animations are used to animate elements across windows and/or window transitions are used to move/resize/show/hide windows simultaneously. An interwindow animation is generally used when a visual element or layer is animated from one window to another. To perform the interwindow animation, a transparent overlay window may be located over the windows that the visual element moves between. The visual element is then animated in the transparent overlay across the two windows. A window transition is generally used when you want to use animation to move, resize, or show/hide a window. To perform the window transition animation, an overlay window may be located on top of the underlying windows to be animated and the content of these windows is redirected into the overlay window in order to animate their content. Layers may be used to represent images or visual elements that can be moved as part of an animation. These layers may be placed in the animation overlay window to move visual elements across the underlying windows. Supported underlying windows are directed to draw content directly to the overlay window during a time of the animation. A picture of the underlying window is drawn to the animation overlay window when the underlying window is not-supported such that the underlying window is included within the animation. When the animation is complete, the animation overlay window may be removed. Additional details regarding the operation of animation manager 26 will be provided below.

[0024] FIGURE 2 illustrates an exemplary system for performing cross window animations. As illustrated, system 200 includes service 210, animation manager 240, store 245, touch screen input device/display 250 (e.g. slate) and smart phone 230.

[0025] As illustrated, service 210 is a cloud based and/or enterprise based service that may be configured to provide productivity services (e.g. MICROSOFT OFFICE 365, MICROSOFT WEB APPS, MICROSOFT POWERPOINT). Functionality of one or more of the services/applications provided by service 210 may also be configured as a client based application. For example, a client device may include an application that performs

animations that may cross different windows. Although system 200 shows a productivity service, other services/applications may be configured to perform cross window animations.

5 **[0026]** As illustrated, service 210 is a multi-tenant service that provides resources 215 and services to any number of tenants (e.g. Tenants 1-N). According to an embodiment, multi-tenant service 210 is a cloud based service that provides resources/services 215 to tenants subscribed to the service and maintains each tenant's data separately and protected from other tenant data.

10 **[0027]** System 200 as illustrated comprises a touch screen input device/display 250 (e.g. a slate/tablet device) and mobile phone 230 that detects when a touch input has been received (e.g. a finger touching or nearly touching the touch screen). Any type of touch screen may be utilized that detects a user's touch input. For example, the touch screen may include one or more layers of capacitive material that detects the touch input. Other sensors may be used in addition to or in place of the capacitive material. For example,
15 Infrared (IR) sensors may be used. According to an embodiment, the touch screen is configured to detect objects that in contact with or above a touchable surface. Although the term "above" is used in this description, it should be understood that the orientation of the touch panel system is irrelevant. The term "above" is intended to be applicable to all such orientations. The touch screen may be configured to determine locations of where
20 touch input is received (e.g. a starting point, intermediate points and an ending point). Actual contact between the touchable surface and the object may be detected by any suitable means, including, for example, by a vibration sensor or microphone coupled to the touch panel. A non-exhaustive list of examples for sensors to detect contact includes
25 capacitive sensors, resistive sensors, inductive sensors, laser vibrometers, and LED vibrometers.

30 **[0028]** As illustrated, touch screen input device/display 250 and mobile device 230 shows an exemplary display 252/232 of a presentation slide. Mobile phone 230 shows display 232 before a display of user interface element 256. Animation manager 240 is configured to perform cross window animations that may include interwindow animations and/or window transition animations. Interwindow animations animate one or more visual elements over different windows (e.g. opening up a menu, moving an object across different windows, and the like). A window transition is generally used when you want to use animation to move, resize, or show/hide a window. In the current example, a user 264

has selected the file menu option in user interface window 254 to display the user interface elements in window 256. In response to selecting the file menu option, a window transition animation is performed that appears to smoothly resize the user interface window from original size 254 to expanded size 256 while also shrinking the document display window 258 to a smaller document display window 252. The underlying windows may position/resize the windows at any time during/before the animation.

[0029] Animation manager 240 creates an overlay window that covers the portion of the different windows where the animation is to occur (the animation area). The animation area may be configured to include a portion of different windows or configured to cover an entire display area. For example, the animation area may be configured to include application windows that are associated with one or more applications as well as a desktop area and other user interface elements (e.g. a start bar). Generally, the animation area is set to a size such that the animation when performed is contained within the animation area. Some windows may use a rendering method that supports drawing content to the overlay window (supported window). Other windows may not be able to draw content directly to the overlay window (non-supported window). Supported windows that are within the animation area are directed to draw content directly to the overlay window during the animation. In this way, any effects/animations that are being performed in the underlying window show up in the animation (e.g. cursor blinking, text effect being applied to underlying content ...). A picture of any non-supported windows may be drawn to the animation overlay such that the non-supported window is included within the animation. When the animation is complete, the overlay window may be removed and the underlying windows are viewable by the user.

[0030] FIGURE 3 shows exemplary displays showing an interwindow animation and a windows transition animation.

[0031] Display 310 shows an interwindow animation that crosses two different underlying windows. As illustrated, display 310 includes a user interface window 302, an application window 304, a document window 306, an application window 308, and an overlay window 320. A cross window animation may include one or more interwindow animations and/or one or more window transition images.

[0032] In the current example illustrated in display 310, an animation of element 312 appears to start in application window 304 and continues to animate across document window 306. The animation may cross one or more applications (e.g. moving an element to a trash can that is displayed on a desktop from an application). The animation may also

include performing other effects (e.g. resizing, fading in/out, changing one or more display characteristics and the like).

[0033] Overlay window 320 is created to cover the animation area that crosses the different windows. Overlay window 320 is sized such that the animation is contained within the size of the overlay window 320. One or more layers may be placed over the overlay window to move visual elements (e.g. element 312) across the underlying windows. As discussed above, supported underlying windows are directed to draw content directly to the overlay window during the time of the animation. According to an embodiment, the parent of the layers from the underlying windows are changed to the overlay window during the time of the animation. A picture of any non-supported windows may be drawn to the animation overlay such that the non-supported window is included within the animation. According to an embodiment, when the animation is complete, overlay window 320 is removed. More details regarding interwindow animation are provided in FIGURE 3 and below.

[0034] Display 350 shows an exemplary window transition animations. As illustrated, display 350 includes user interface window 352/352', document window 356/356', overlay window 360, application window 368, UI element 362 and user interface window 364. Overlay window 360 is sized to cover the animation area that includes user interface window 352, document window 356 and application window 368. The current example shows an animation where user interface window 352 is resized to a larger size (window 352'), document window 356 is resized smaller and moved to a different location (window 356') and a user interface window 364 is displayed. According to an embodiment, the windows that are affected by the display of the animation are registered with the animation manager. Registering the windows is directed at improving the performance of the animation and system by excluding the windows that are not affected by the animation in the various animation operations. A process/application may provide hints to the animation manager to assist in setting animation parameters (e.g. initial/final window size/location/visibility, clipping regions, content static, and the like). As discussed above, the underlying windows may be directed to draw directly to overlay window 360 when supported. According to an embodiment, while the animation is playing the window is blocked from receiving input from a user. More details regarding interwindow animation are provided in FIGURE 4 and below.

[0035] FIGURES 4-5 show illustrative processes for cross window animation. When reading the discussion of the routines presented herein, it should be appreciated that the

logical operations of various embodiments are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system.

The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated and making up the embodiments described herein are referred to variously as operations, structural devices, acts or modules. These operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof. While the operations are shown in a particular order, the ordering of the operations may change and be performed in other orderings.

[0036] FIGURE 4 shows a process for cross window animation using an overlay window to animate a layer between different windows.

[0037] After a start operation, the process flows to operation 410, where the animation area is determined. The animation area is sized to cover the portions of the windows that are used during the cross window animation. The animation area may include windows from one or more applications/processes. For example, the animation are may cross two or more windows from a same application and/or from different applications. The animation area may include all/portion of a display.

[0038] Moving to operation 420, the overlay window is created and sized based on the determined animation area. According to an embodiment, the overlay window includes a transparent background that is displayed above the windows in the animation area.

[0039] Flowing to operation 430, the parent for the layers of the underlying windows is changed to the overlay window. The parent is changed for the duration of the performance of the animation.

[0040] Transitioning to operation 440, the animation(s) to be performed are scheduled and performed. According to an embodiment, a layer is moved into the overlay window, moved across the overlay window such that it crosses the windows that are beneath it, and then the layer is placed back (e.g. change the parent to the underlying window) into the underlying window where the animation ends.

[0041] Moving to operation 450, the process waits for the animations to complete. According to an embodiment, the animations may be made to be modal and/or an animation completed notification handler is configured to be used to provide a notification when the animations in the overlay have completed. During the animation time, the user

interface thread for the underlying windows is blocked. Any received instructions that occur during this time may be stored and executed once the animation is complete.

According to an embodiment, messages from an operating system continue to be processed during the animation. Timing parameters may be used to help ensure a

5 responsive user interface such that a user is not stuck waiting on an animation to complete. For example, a computing device may have a graphics card that is not capable of quickly displaying animations. According to an embodiment, when it is estimated that an animation is going to take longer than a predetermined time to complete (e.g. 200, 300, 400, 500 milliseconds) the cross window animation is canceled.

10 [0042] Flowing to operation 460, the parent of the layers of the underlying windows are changed back to the underlying windows from the overlay window.

[0043] Transitioning to operation 470, the overlay window is removed from the display.

15 [0044] The process then moves to an end operation and returns to processing other actions.

[0045] FIGURE 5 shows a process for cross window animation including moving/resizing different windows using an overlay window.

20 [0046] After a start operation, the process flows to operation 510, where the animation area is determined. The animation area is sized to cover the portions of the windows that are used during the cross window animation. The animation area may include windows from one or more applications/processes. For example, the animation are may cross two or more windows from a same application and/or from different applications. The animation area may include all/portion of a display.

25 [0047] Moving to operation 520, the overlay window is created and sized based on the determined animation area. The overlay window may include an opaque background or a transparent background that is displayed above the windows in the animation area.

30 [0048] Flowing to operation 530, the windows that are affected by the cross window animation may be registered. According to an embodiment, regardless of whether a window is registered, windows that are visible during the animation are automatically redirected into the overlay window to help ensure that the overlay depicts an accurate picture of the underlying window tree. Registering specific windows allows clients to specify which windows are to be included in the animation. This helps to save processing time by reducing the number of windows processed during operation. For each registered window it takes time to redirect windows into the overlay window and/or to capture

pictures of them to bring into the overlay window. Windows may be registered before and/or during the cross window animation. For example, windows may be registered during the animation when they are about to become visible.

[0049] Transitioning to operation 540, hints may be provided/received. Hints may be used to assist in correcting any behaviors that are not being performed correctly. Hinting may also be used to assist in avoiding changes to existing window layout code. For example, to achieve a slide in/out effect when a window is shown or hidden, the client provide hints that define an initial location and a final location for a window. Hinting may help to avoid unnecessary window event triggering and may remove interdependencies between window transition logic and the existing layout code. Many different type of hints may be provided to the animation manager. According to an embodiment, the hints include an initial location of the window, a final location of the window, an initial size of the window, a final size of the window, an initial visibility state of the window, a final visibility state of the window, a hint that specifies that the windows contents are a solid color, a hint that specifies that a window's contents will not change between initial and final states (that doesn't specify that the size of the window will not change, but instead that the contents painted into the client area are the same where they intersect), a hint that specifies that a window is a control (e.g. an ActiveX control), a hint that the window will not be seen (this helps to ensure that the window's image is not redirected during the animation, thereby avoiding a potentially expensive amount of work), a resize hint that adds extra area to the overlay region (e.g. a client may want to add more area to account for rotation or other transforms that will be applied manually to the redirected layers), and a clipping hint that is used to achieve a clipping effect as windows are transitioned.

[0050] Moving to operation 550, the layout of the windows and the animation are performed. The layers and properties on the layers are set to perform the animation.

[0051] Flowing to operation 530, extra layers may be added to the overlay window and/or existing layers may be modified. For example, an extra layer may be added to provide an additional animation effect. For example, a picture may be displayed at a certain point during the animation, a light source added, a sparkling effect, and the like may be added. An existing layer may be modified to create/change another desired animation effect.

[0052] Transitioning to operation 570, the cross window animations are performed using the overlay window.

[0053] Moving to operation 580, the overlay window is removed.

[0054] The process then moves to an end operation and returns to processing other actions.

[0055] FIGURE 6 illustrates a system architecture used in cross window animations, as described herein. Content used and displayed by the application (e.g. application 1020) and the animation manager 26 may be stored at different locations. For example, application 1020 may use/store data using directory services 1022, web portals 1024, mailbox services 1026, instant messaging stores 1028 and social networking sites 1030. The application 1020 may use any of these types of systems or the like. A server 1032 may be used to access sources and to prepare and display cross window animations. For example, server 1032 may generate a cross window animation for application 1020 to display at a client (e.g. a browser or some other window). As one example, server 1032 may be a web server configured to provide productivity services (e.g. presentation, spreadsheet, word-processing, messaging, document collaboration, and the like) to one or more users. Server 1032 may use the web to interact with clients through a network 1008. Server 1032 may also comprise an application program (e.g. a presentation application, an application that uses animations, ...). Examples of clients that may interact with server 1032 and a presentation application include computing device 1002, which may include any general purpose personal computer, a tablet computing device 1004 and/or mobile computing device 1006 which may include smart phones. Any of these devices may obtain content from the store 1016.

[0056] FIGURES 7-10 show exemplary displays illustrating cross window animations. The examples shown herein are for illustration purposes and not intended to be limiting.

[0057] FIGURE 7 shows an exemplary landscape slate display a cross window animation.

[0058] Display 710 shows a slate that displays a presentation slide 712. While display 710 appears to be displayed within a single window, many windows are used in the display. For example, window 720 may be used to display user interface elements, window 714 may be used to display thumbnails of the slides of a presentation and window 716 is used to display slide 712. When any animation appears to move from one window to another window, that animation is a cross window animation. In the current example, a cross window animation is performed when the user interface 760 is displayed. Display 710 shows an initial state before the animation and display 750 shows the final state after the cross window animation is performed.

[0059] If a cross window animation is not performed, the position changes of the different elements may appear to be jerky to a user. For example, instead of seeing a smooth resizing of the user interface window and the document window, the windows would appear jerky to a user when displayed at the determined size. As discussed above, the cross window animation is performed in an overlay window. Beneath the overlay window, the windows may have already changed in size/position, but the animation in the overlay window shows those changes in position/size smoothly. The overlay window smoothly animates the reposition/resizing. As the animation in the overlay window is occurring, changes that occur in the registered windows may also be redirected in real time, such that changes that occur in the registered underlying windows, the user still sees the animations of the content inside the underlying window.

[0060] FIGURE 8 shows an exemplary landscape slate display a cross window animation.

[0061] Display 810 shows a slate that displays a presentation slide 820. While display 810 appears to be displayed within a single window, many windows may be used in the display. For example, a window may be used to display user interface elements, a window may be used to display other elements, a window may be used to display a document, and the like. In the current example, a cross window animation is performed in response to a user interaction. Display 810 shows an initial state before the animation and display 850 shows the final state after the cross window animation is performed.

[0062] As can be seen, the slide 820 has been shrunk and moved to appear within a different window/location and the contents of the document window have been replaced with content 860. The animation that is performed from initial display 810 to final display 850 may include different animation effects. For example, slide 820 may appear to smoothly shrink and move to slide position 820' as shown in display 850. Slide 820 may also appear to fade while content 860 appears within display 850. Generally, any animation effects may be applied during the transitions.

[0063] FIGURE 9 shows an exemplary landscape slate display a cross window animation.

[0064] Display 910 shows a slate that displays content 915. While display 910 appears to be displayed within a single window, many windows may be used in the display. For example, a window 912 may be used to display user interface elements, a window 920 may be used to display tasks, a window 930 may be used to display a discussion, a window 940 may be used to display a social connector, and the like. In the

current example, a cross window animation is performed in response to a user interaction. Display 910 shows an initial state before the animation and display 950 shows the final state after the cross window animation is performed.

5 [0065] As illustrated, in response to a user selecting a tasks UI element in window 912, window 960 is displayed, window 930 and 940 are repositioned and window 920 is removed from the display. One or more animations may be applied during the cross window animation.

 [0066] FIGURE 10 shows an exemplary landscape slate display a cross window animation.

10 [0067] Display 950 shows a slate as illustrated in FIGURE 9. While display 950 appears to be displayed within a single window, many windows may be used in the display. For example, a window 912 may be used to display user interface elements, a window 930 may be used to display a discussion, a window 940 may be used to display a social connector, a window 960 may be displayed to show options, and the like. In the
15 current example, a cross window animation is performed in response to a user interaction with an element shown in window 960. Display 950 shows an initial state before the animation and display 1000 shows the final state after the cross window animation is performed.

 [0068] As illustrated, in response to a user selecting a fly out element 962 in window
20 960, window 1060 is displayed, window 930 and 940 are repositioned and window 960 is removed from the display. One or more animations may be applied during the cross window animation.

 [0069] While certain embodiments of the invention have been described, other embodiments may exist. Furthermore, although embodiments of the present invention
25 have been described as being associated with data stored in memory and other storage mediums, data can also be stored on or read from other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or a CD-ROM, a carrier wave from the Internet, or other forms of RAM or ROM. Further, the disclosed methods' stages may be modified in any manner, including by reordering stages and/or inserting or
30 deleting stages, without departing from the invention.

 [0070] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for cross window animation, comprising:
determining an animation area that covers a portion of different windows;
5 creating an overlay window that is sized to cover the animation area;
displaying the overlay window over the portion of the different windows;
performing an animation using the overlay window;
displaying the animation in the overlay window such that the animation appears to
cross the different windows that are under the display of the overlay window; and
10 removing the overlay window when the animation is complete.
2. The method of Claim 1, wherein the animation is at least one of: an
interwindow animation that animates a visual element over the different windows and a
windows transition animation that comprises at least one of: changing a size of at least one
of the different windows and moving at least one of the different windows.
- 15 3. The method of Claim 1, further comprising changing a parent of layers of
the different windows to the overlay window and changing the parent of layers back from
the overlay window to the different windows after the animation is complete.
4. The method of Claim 1, further comprising registering the different
windows that are affected by the animation such that content from the registered windows
20 is rendered to the overlay window during the animation.
5. The method of Claim 2, further comprising receiving hints that specify
information about the different windows that are affected in the windows transition
animation that comprises an initial window location, an initial window size, a final
window location, a final window size, an initial window visibility, a final window
25 visibility and a clipping region in the overlay window.
6. The method of Claim 1, further comprising blocking a user interface thread
until at least one of: determining the animation is complete and determining when the
animation exceeds a predetermined time period.
7. A computer-readable medium having computer-executable instructions for
30 cross window animation, comprising:
determining an animation area that covers a portion of different windows;
creating an overlay window that is sized to cover the animation area;
displaying the overlay window over the portion of the different windows;

performing an animation using the overlay window, wherein the animation is at least one of: an interwindow animation that animates a visual element over the different windows and a windows transition animation that comprises at least one of: changing a size of at least one of the different windows and moving at least one of the different

5 windows; and

displaying the animation in the overlay window such that the animation appears to cross the different windows that are under the display of the overlay window.

8. A system for cross window animation, comprising:

a display;

10 a network connection that is coupled to tenants of the multi-tenant service;

a processor and a computer-readable medium;

an operating environment stored on the computer-readable medium and executing on the processor; and

a process operating under the control of the operating environment and

15 operative to perform actions, comprising:

determining an animation area that covers a portion of different windows;

creating an overlay window that is sized to cover the animation area;

20 displaying the overlay window over the portion of the different windows;

performing an animation using the overlay window, wherein the

animation is at least one of: an interwindow animation that animates a

visual element over the different windows and a windows transition

25 animation that comprises at least one of: changing a size of at least one of the different windows and moving at least one of the different windows;

displaying the animation in the overlay window such that the animation appears to cross the different windows that are under the display of the overlay window; and

30 removing the overlay window when the animation is complete.

9. The system of Claim 8, further comprising changing a parent of layers of the different windows to the overlay window before performing the animation and changing the parent of layers back from the overlay window to the different windows after the animation is complete.

10. The system of Claim 8, further comprising blocking a user interface thread until at least one of: determining the animation is complete and determining when the animation exceeds a predetermined time period.

1/10

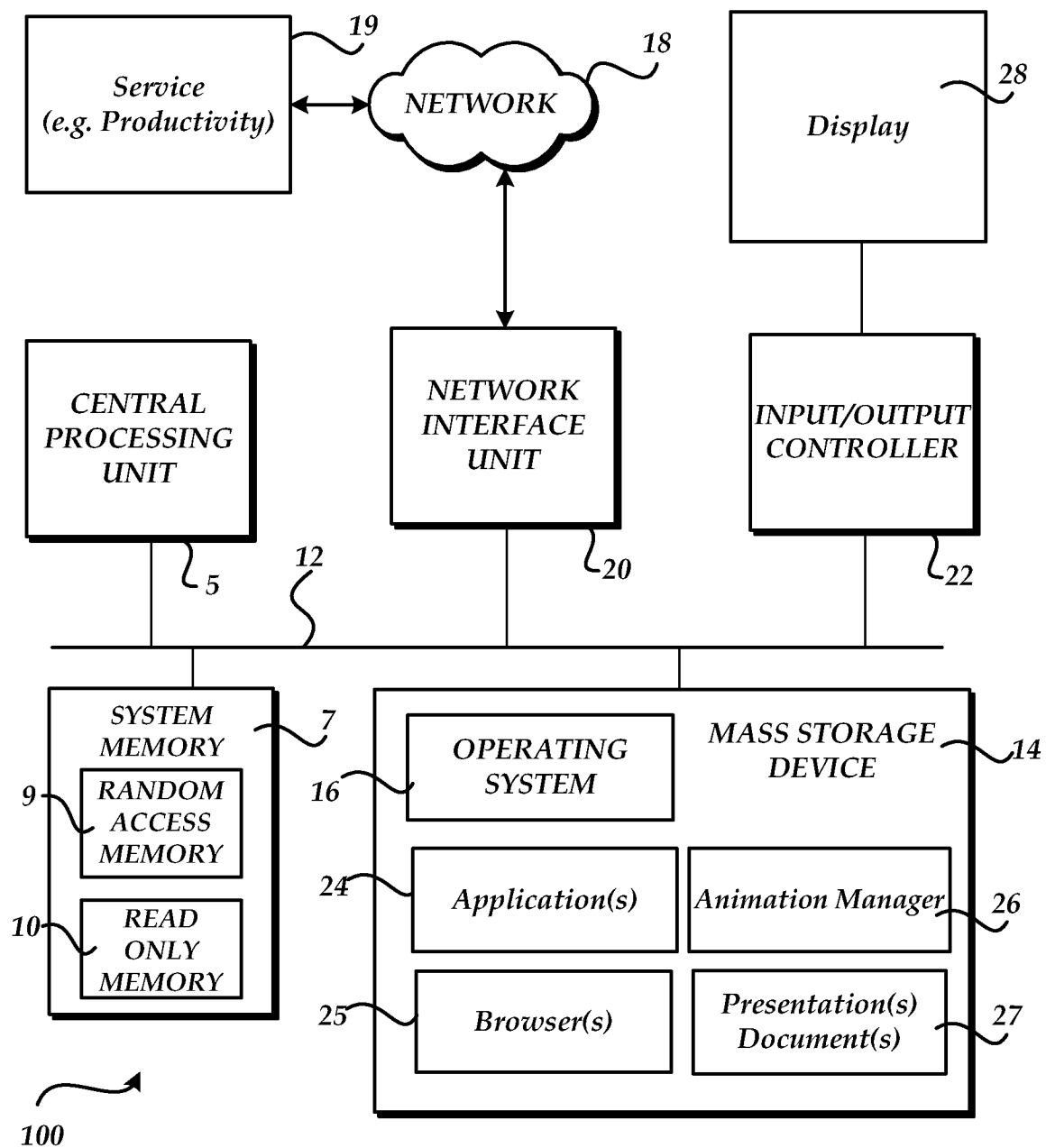


Fig. 1

2/10

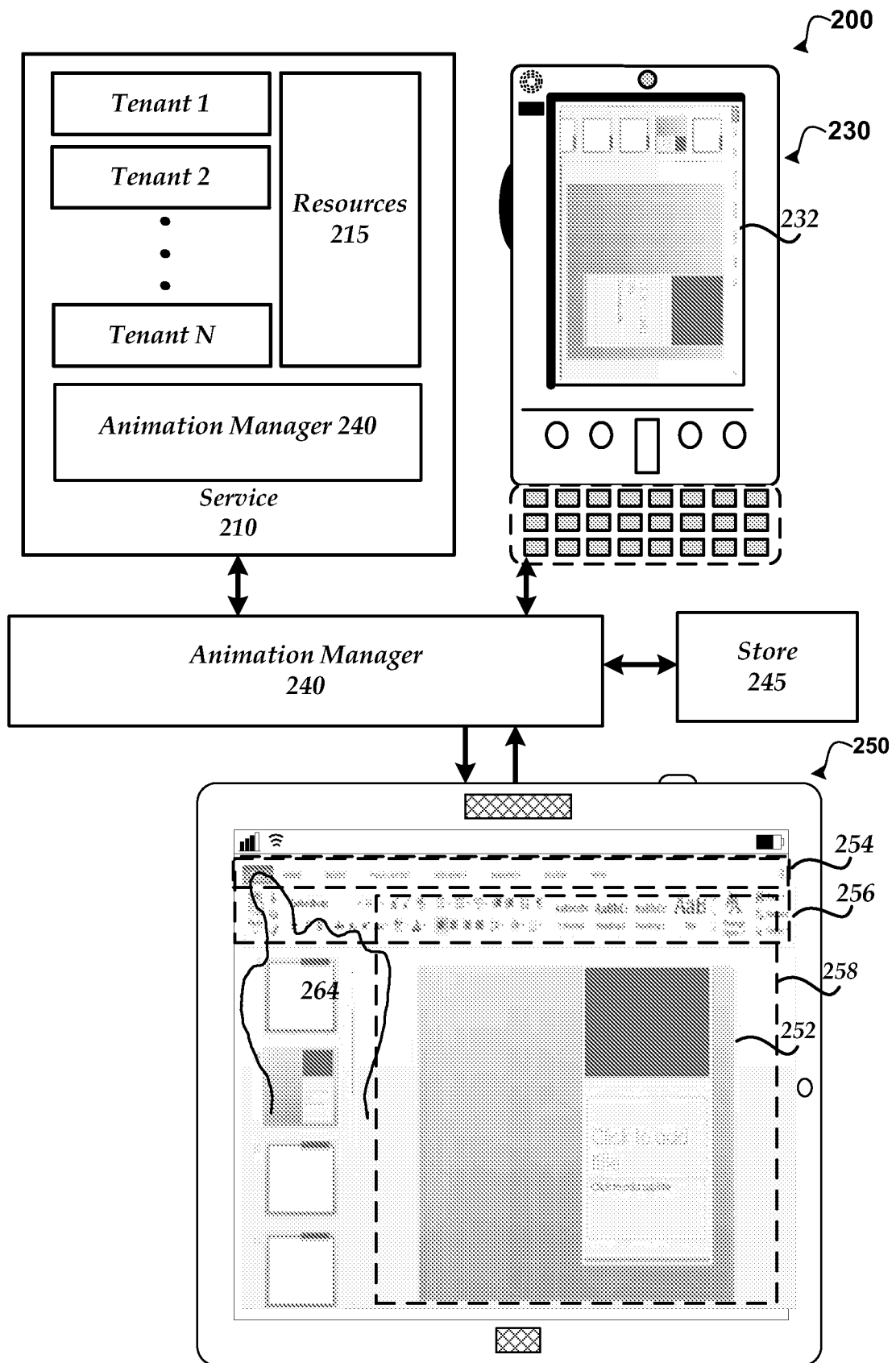


Fig. 2

3/10

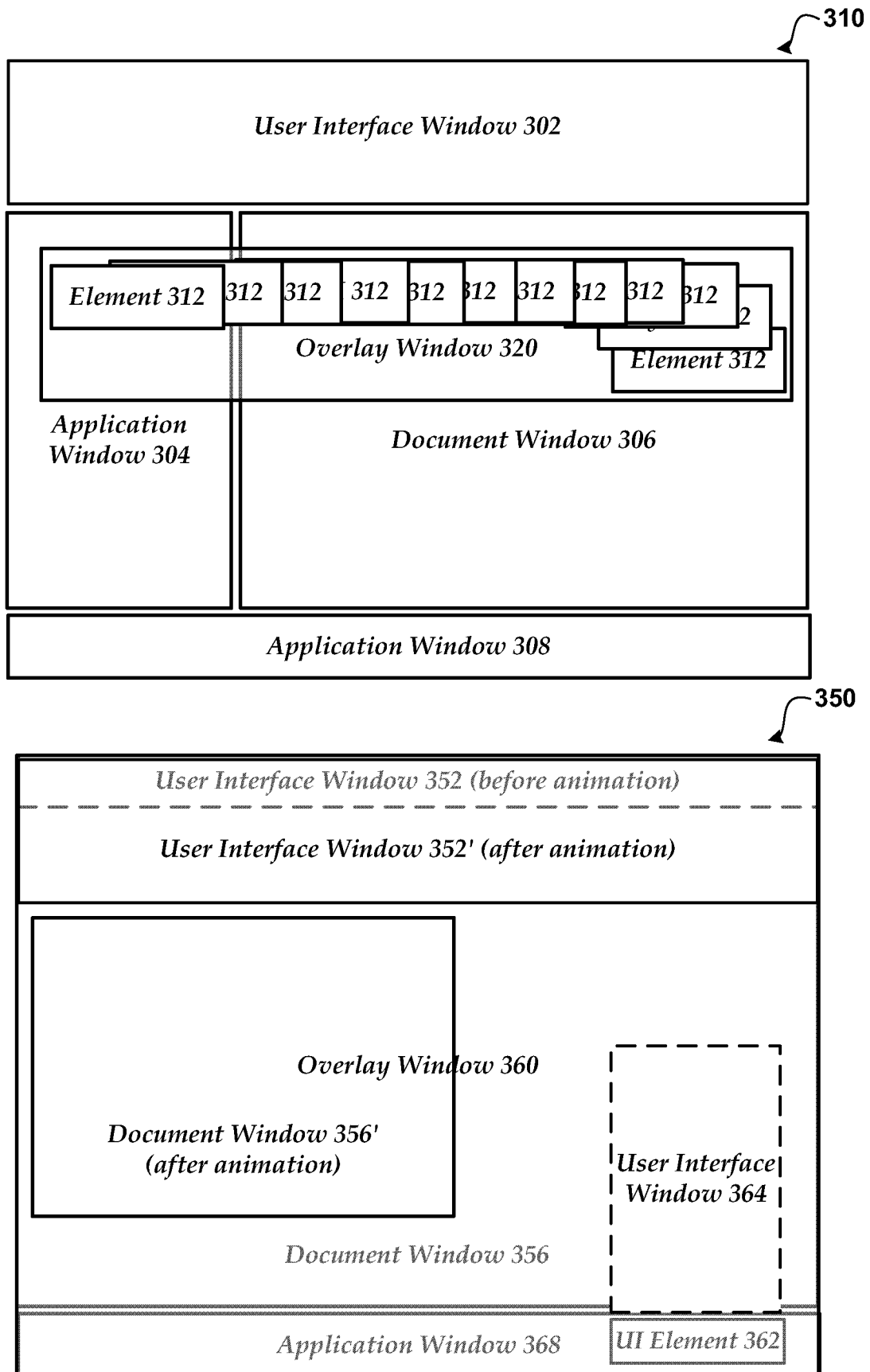


Fig. 3

4/10

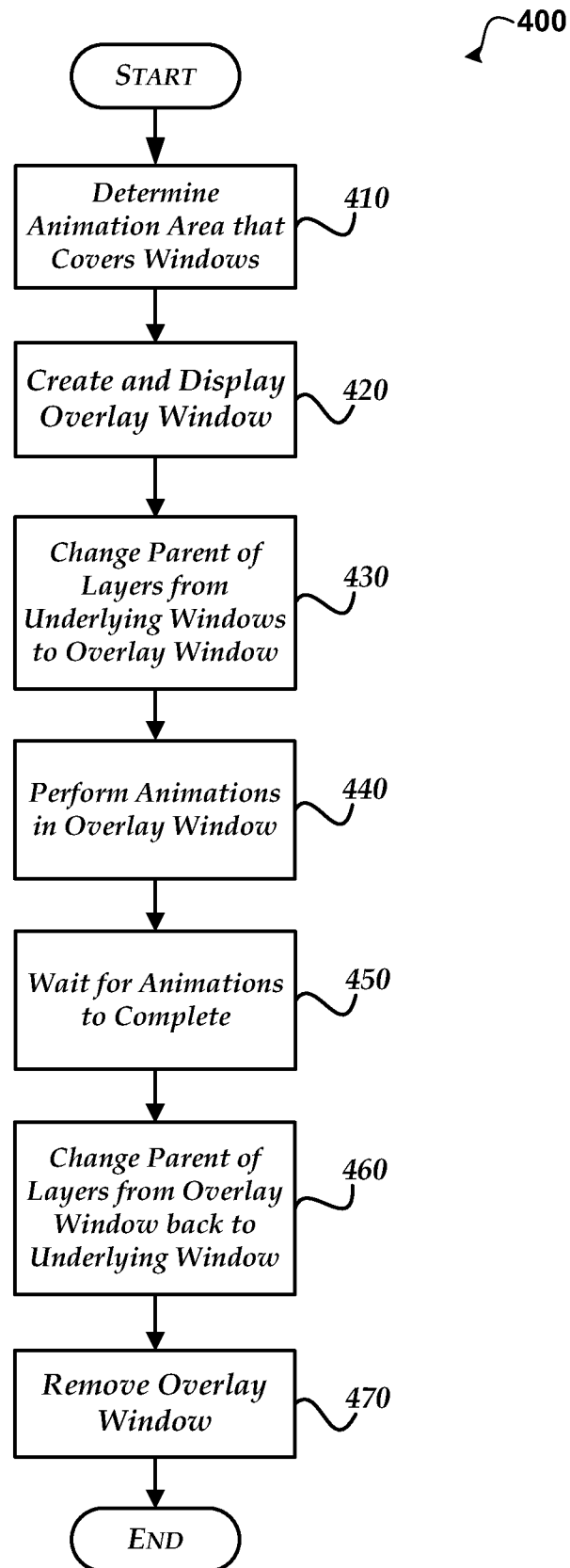


Fig. 4

5/10

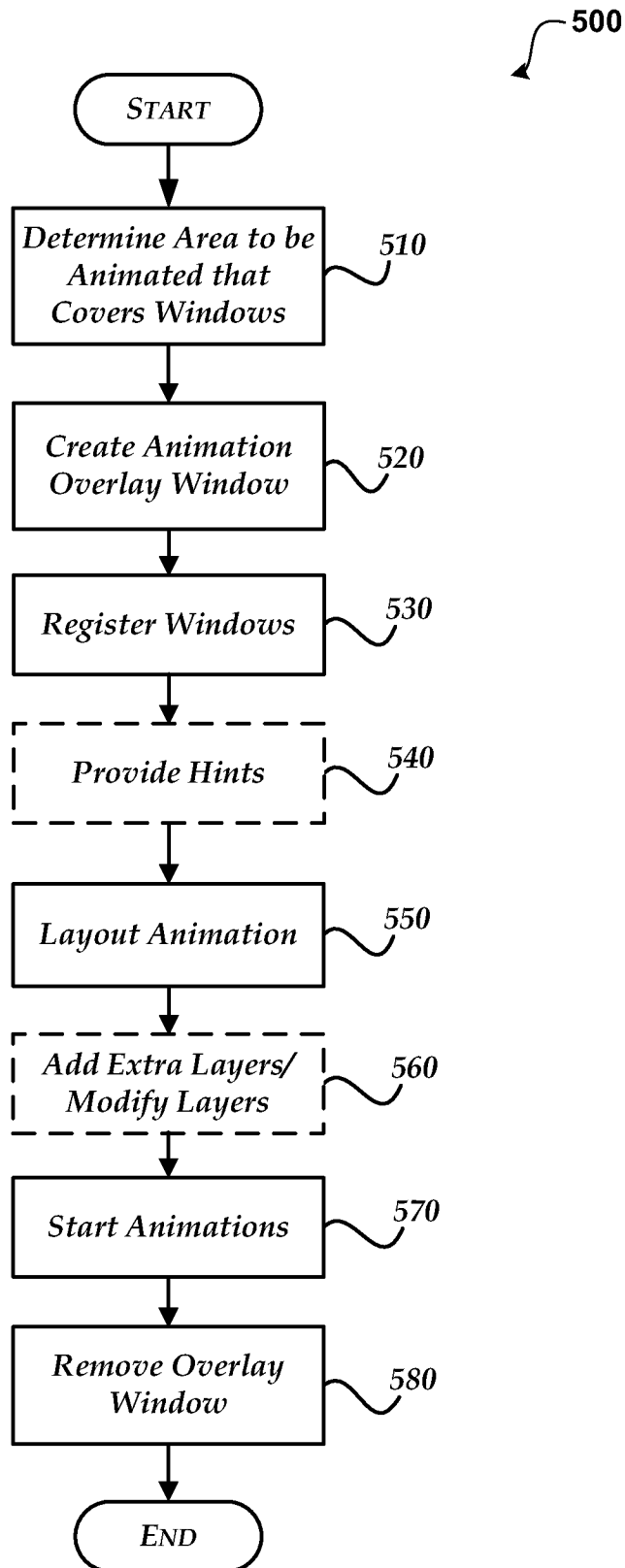


Fig. 5

6/10

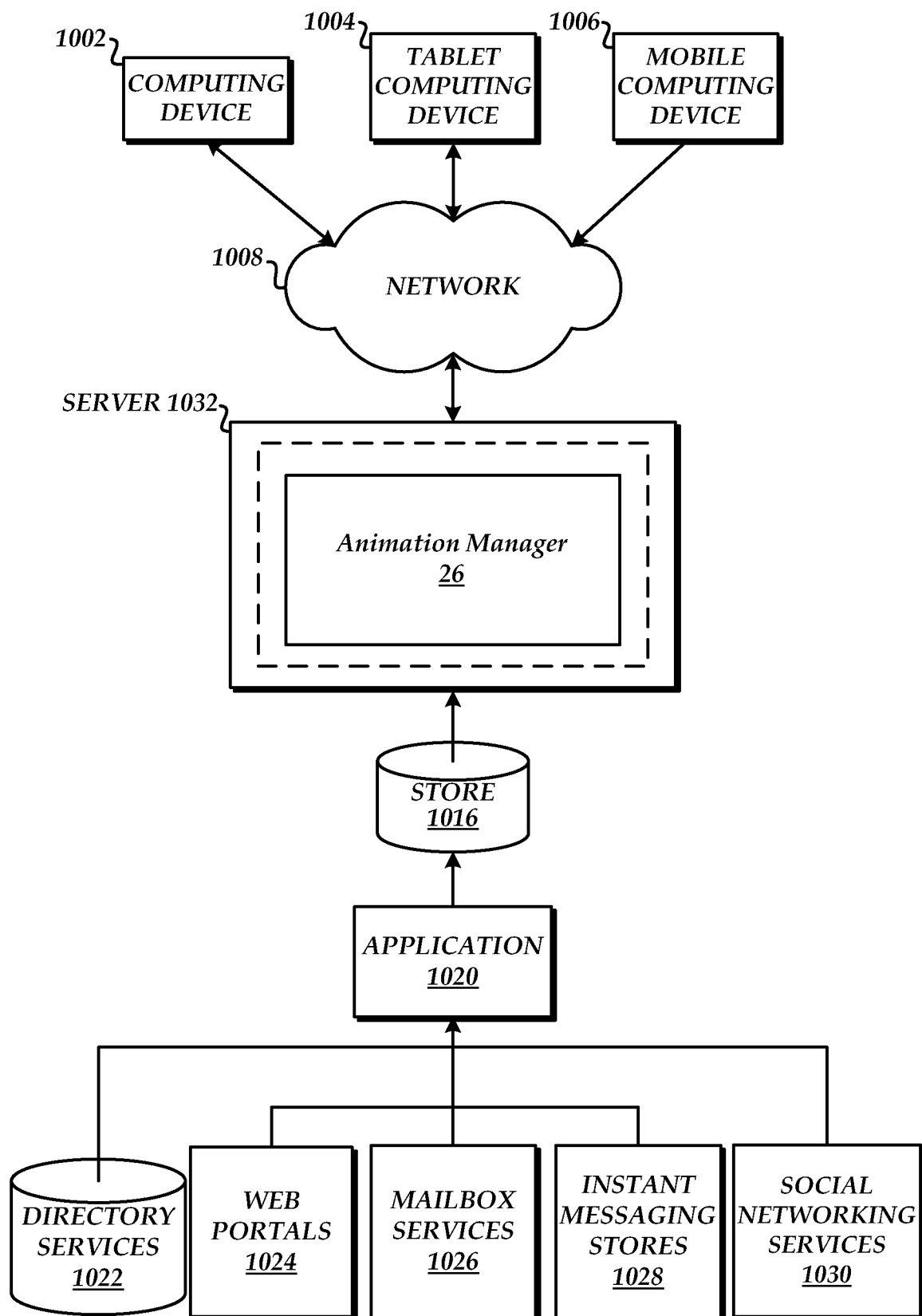


Fig. 6

7/10

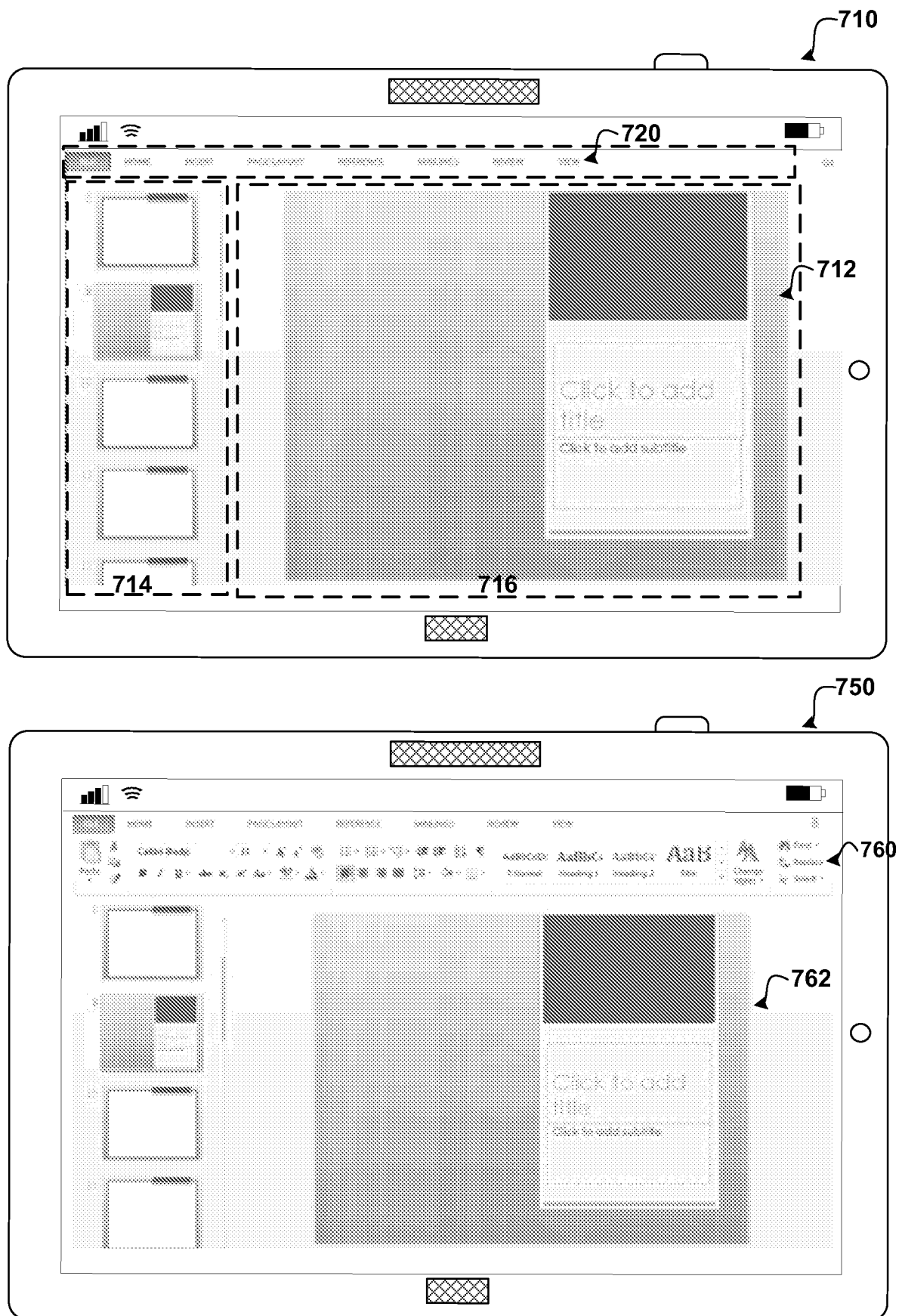


Fig. 7

8/10

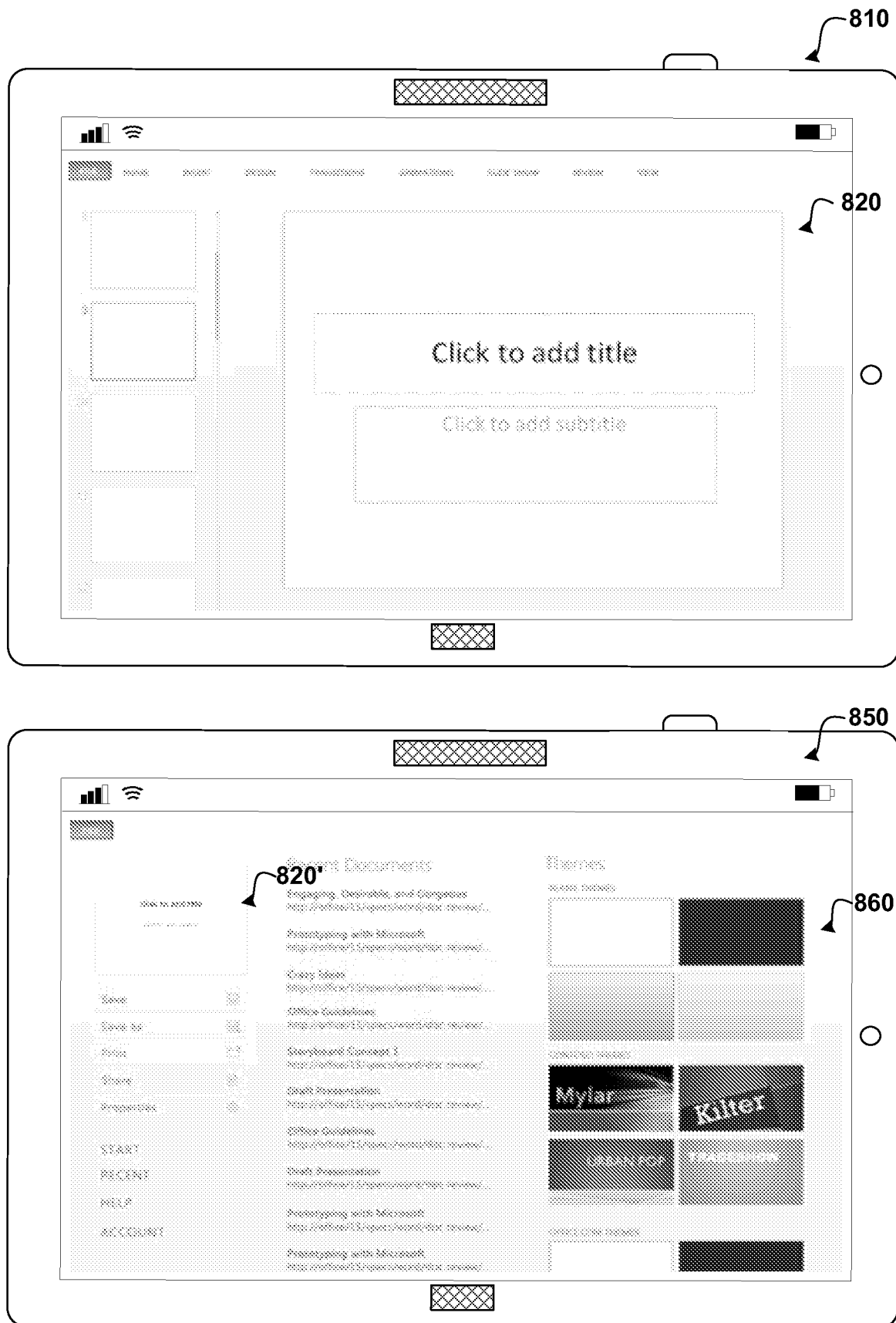


Fig. 8

9/10

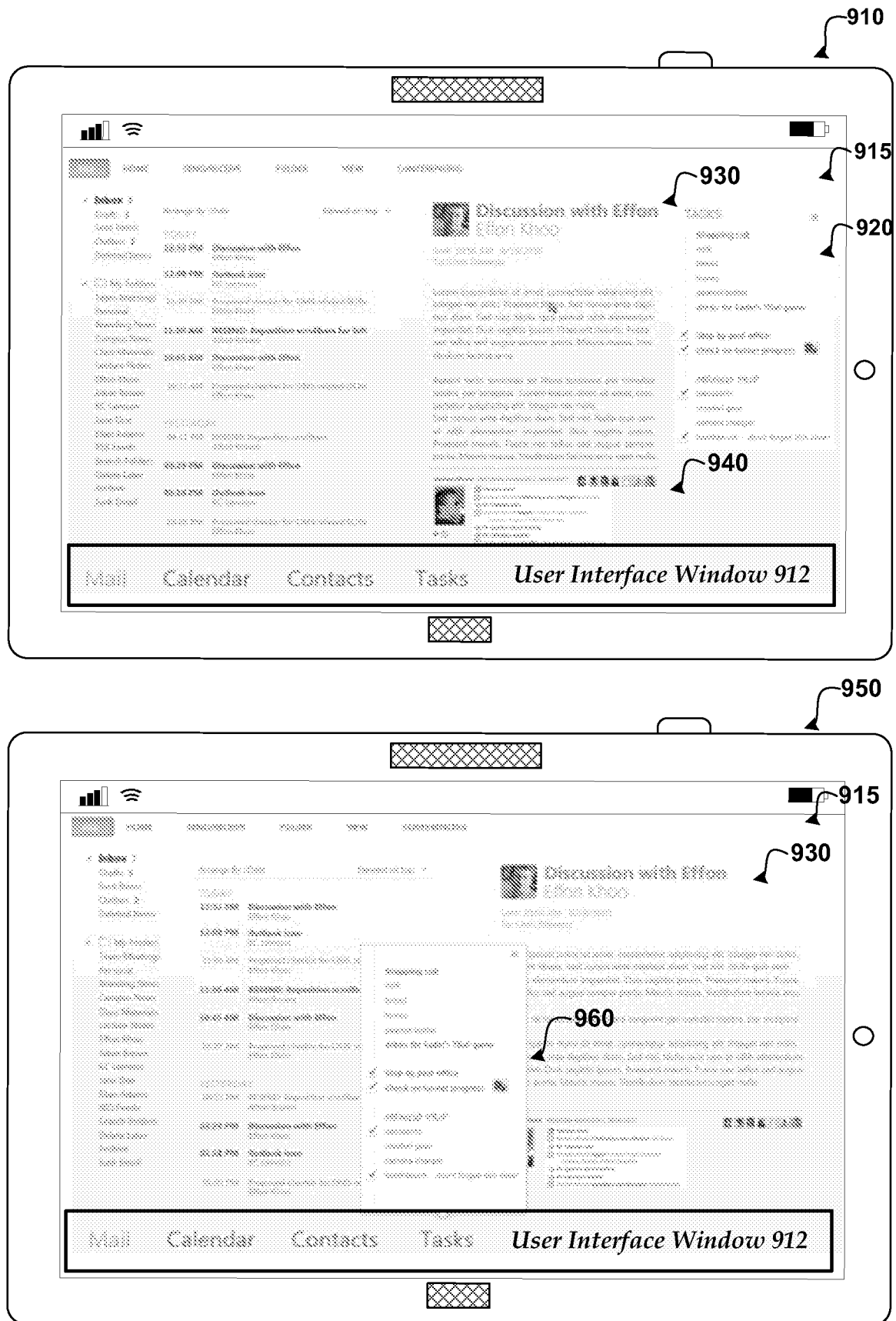


Fig. 9

10/10

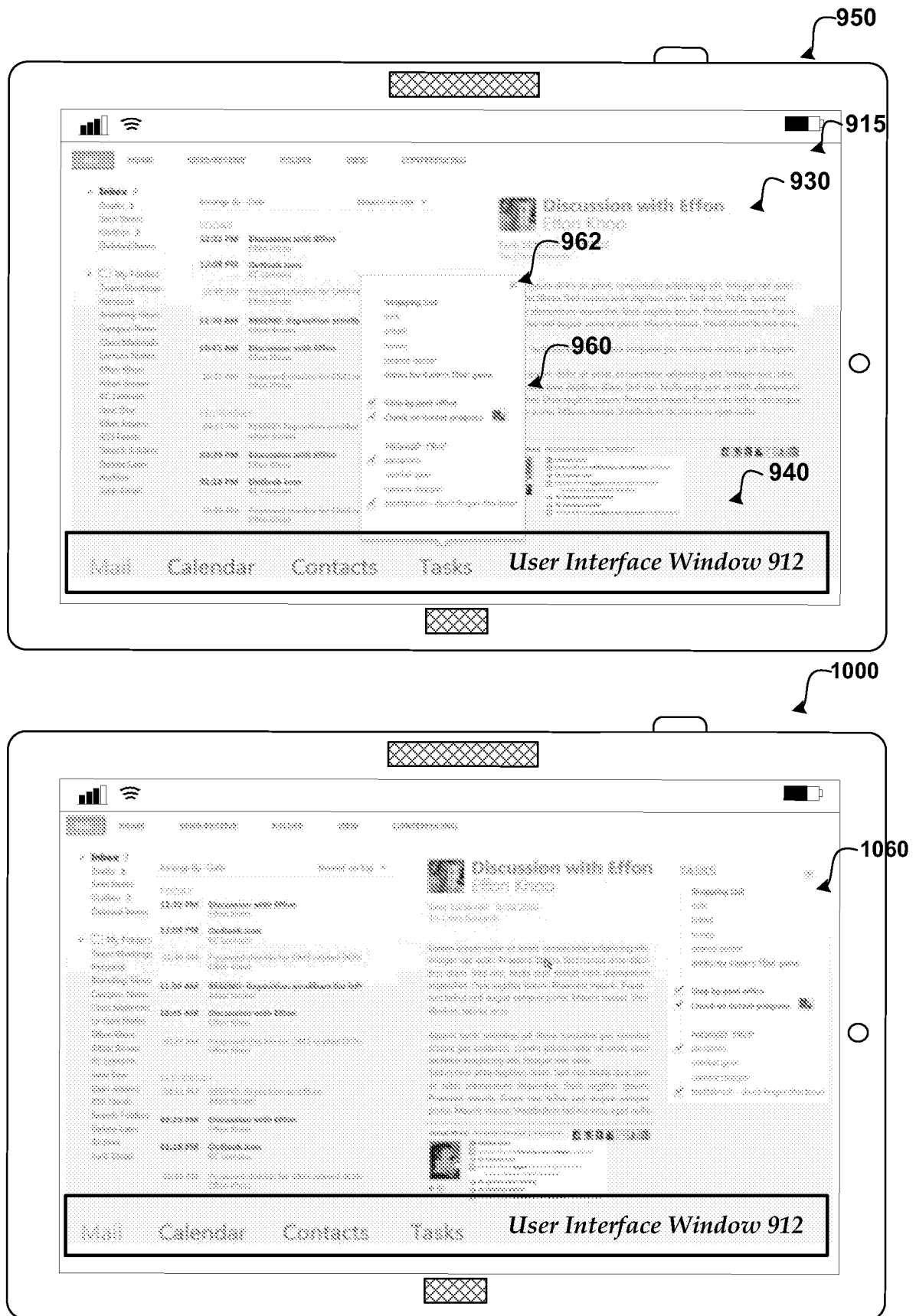


Fig. 10

A. CLASSIFICATION OF SUBJECT MATTER**G06F 3/048(2006.01)i, G06F 3/14(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 3/048; G06F 21/24; G06F 15/00; G06F 3/00; G06F 3/033; G09G 5/14; G00F 21/24; G09G 5/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: different window, animation and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 11-184596 A (ITO NAOKI) 09 July 1999 See paragraphs [0011]–[0012] and figures 1–4.	1–10
A	US 2004-0100501 A1 (JASON ROBERT DORNBACH) 27 May 2004 See paragraph [0016]; claim 1 and figures 1–2.	1–10
A	US 06008809 A (BROOKS, ARTHUR P.) 28 December 1999 See column 6, line 11 – column 7, line 31 and figures 6–7.	1–10
A	WO 2004-114113 A1 (APPLE COMPUTER, INC. et al.) 29 December 2004 See page 13, line 31 – page 14, line 31; claim 1 and figures 16, 23a–23d.	1–10
A	US 7316032 B2 (TAYEBI, AMAD et al.) 01 January 2008 See column 22, lines 37–61 and figures 5a–5f.	1–10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

20 MARCH 2013 (20.03.2013)

Date of mailing of the international search report

21 MARCH 2013 (21.03.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan
City, 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

LEE, Dong Yun

Telephone No. 82-42-481-8734



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/064259

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 11-184596 A	09.07.1999	None	
US 2004-0100501 A1	27.05.2004	None	
US 06008809 A	28.12.1999	JP 11-110099 A JP 3504502 B2	23.04.1999 08.03.2004
WO 2004-114113 A1	29.12.2004	CN 100474221 C CN 101488072 A CN 101488072 B CN 1826575 A CN 1826575 CO DE 202004009752 U1 EP 1491990 A2 EP 1491990 A3 EP 1491990 B1 EP 1644816 A1 EP 2182424 A1 EP 2284659 A2 EP 2284659 A3 US 2004-0261037 A1 US 2004-0261038 A1 US 2007-0022389 A1 US 2007-0288863 A1 US 7739617 B2 US 8127248 B2	01.04.2009 22.07.2009 21.03.2012 30.08.2006 30.08.2006 11.11.2004 29.12.2004 26.01.2005 17.02.2010 12.04.2006 05.05.2010 16.02.2011 09.11.2011 23.12.2004 23.12.2004 25.01.2007 13.12.2007 15.06.2010 28.02.2012
US 7316032 B2	01.01.2008	US 2003-0163724 A1 US 2008-0071685 A1 US 2008-0071686 A1 US 2009-0257591 A1 US 2009-0307078 A1 US 7562397 B1 US 8069489 B2	28.08.2003 20.03.2008 20.03.2008 15.10.2009 10.12.2009 14.07.2009 29.11.2011