(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 October 2014 (23.10.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/171963 A1

(51) International Patent Classification:
G06F 9/44 (2006.01) *G06F 3/048* (2013.01)

(21) International Application Number:
PCT/US2013/060247

(22) International Filing Date:
18 September 2013 (18.09.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/863,369 15 April 2013 (15.04.2013) US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: VRANJES, Miron; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). SATTERFIELD, Jesse Clay; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). WORLEY, Matthew Isaac; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). SAREEN, Chaitanya; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). SUNDELIN, Nils Anders; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). MALANI, Abhishek; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). STEINGLASS, Alice; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). JARRETT, Robert James; c/o Microsoft Corporation, LCA - International 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, 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, SA, 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, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

[Continued on next page]

(54) Title: APPLICATION WINDOW DIVIDER CONTROL FOR WINDOW LAYOUT MANAGEMENT

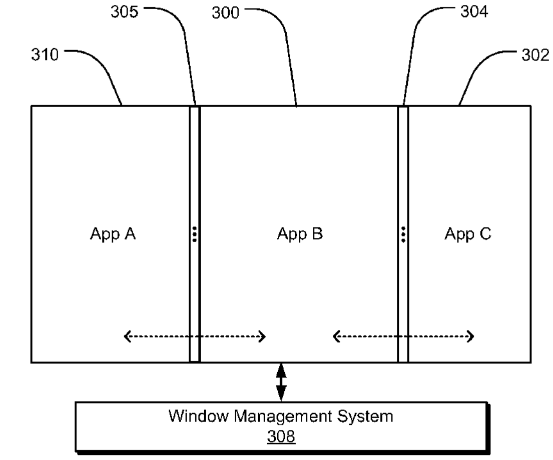


FIG. 3

(57) Abstract: An application window divider control is shared by a first application window and a second application window in a user interface. Based on a received directional instruction that moves the application window divider control along an axis of the user interface, placement of the first application window and the second application window may be positioned across a range of consistently spaced points along an axis of the user interface. Placement adjustment can also impact other application windows in the user interface. The application window divider control may also snap to provide "magnetic" points along the axis when certain conditions are satisfied.

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

APPLICATION WINDOW DIVIDER CONTROL FOR WINDOW LAYOUT MANAGEMENT

BACKGROUND

[0001] A computer operating system, including its corresponding user interface, allows a user to adjust an application window for presentation to the user through the user interface. However, such user-directed adjustments come with inconvenient limitations, particularly as user interfaces become more modern and flexible.

SUMMARY

[0002] Implementations described and claimed herein address the foregoing problems by providing an application window divider control that is shared by a first application window and a second application window in a user interface. Based on a received directional instruction that moves the application window divider control along an axis of the user interface, placement of the first application window and the second application window may be positioned across a range of consistently spaced points along an axis of the user interface. Placement adjustment can also impact other application windows in the user interface. The application window divider control may also snap to provide "magnetic" points along the axis when certain conditions are satisfied.

[0003] 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 to limit the scope of the claimed subject matter.

[0004] Other implementations are also described and recited herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 illustrates two application windows sharing an application window divider control in an example user interface screen.

[0006] FIG. 2 illustrates a schematic of two application windows sharing an application window divider control in an example user interface.

[0007] FIG. 3 illustrates a schematic of two application windows sharing an application window divider control and a third application window in an example user interface.

[0008] FIG. 4 illustrates an example sequence of user interface operations implementing an application window divider control.

[0009] FIG. 5 illustrates a schematic of two application windows sharing an application window divider control at a magnetic point in an example user interface.

[0010] FIG. 6 illustrates example operations for adjusting placement of two application windows using an application window divider control.

[0011] FIG. 7 illustrates example operations for operating an application window divider control relative to a magnetic point in an example user interface.

[0012] FIG. 8 illustrates an example system that may be useful in implementing the described technology.

DETAILED DESCRIPTION

[0013] Launching an application window includes without limitation initially executing the application and switching to a new or hidden application window of an already executing application. In addition, an application window may include without limitation a window of the operating system components, an operating system utility, and a special purpose application program (e.g., a Web browser program, a word processing program, a spreadsheet program).

[0014] FIG. 1 illustrates two application windows 100 and 102 sharing an application window divider control 110 in an example user interface screen 104. A window management system 106 is typically a component of an operating system or user interface environment but could also be an independent application. The window management system 106 manages the display, placement, layout, appearance, and other aspects of the application windows 100 and 102, as well as other windows and user interface characteristics and operations.

[0015] The application window 100 presents a search result window displaying search results from a Web search engine. The individual search results (such as a search result 108) includes representative text and/or one or more images along with a browser navigable link, which can be selected by the user for navigation to a Web site identified by the browser navigable link. The application window 102 presents results from a weather application or Web site, including a 4-day forecast of weather in Denver, CO. A window boundary control 110 divides the displayed area of the application window 100 and displayed area of the application window 102, in the illustrated case, separating the application windows. In some implementations, the window boundary control 110 may be manipulated by the user to change the size and/or location of one or both windows.

[0016] The application windows 100 and 102 are shown in FIG. 1 as being non-overlapping windows. However, the described technology is not limited to non-overlapping windowing environments. For example, in such environments, each application window in an overlapping windowing environment may be bounded by a window boundary control, which can be manipulated by the user to change the location, size, and front-to-back ordering (collectively, "placement") of the potentially overlapping windows (e.g., the z-ordering of the windows). The window management system 106 receives input through the application windows 100 and 102 and through other user interface components (e.g., the keyboard interface, a touch screen interface, a voice interface, and pointing device) and displays the applications windows 100 and 102, the window boundary control 110, and other application windows and controls through the user interface.

[0017] The window boundary control 110 represents a divider between the two distinct application windows 100 and 102 or between an application window and an empty space in the user interface, and therefore may also be referred to as an "application window divider control." The application window divider control may be manipulated through the user interface to communicate changes in the size of an application window (with respect to an empty space) or in the relative sizes of the application windows 100 and 102 to the window management system 106. Accordingly, functionality of the application window divider control represents a window management system control that, rather than being a component of one application window or another, provides an external boundary of one or more application windows and, in some configuration, may be shared by the two or more application windows as part of the user interface and window management system 106.

[0018] In one implementation, two or more application window divider controls may merge together to form a single application window divider control. For example, two application windows may be presented along a horizontal axis of a user interface, separated by an empty space. In this scenario, each application is displayed with an application window divider control between the application window and the empty space. If one of the application window divider controls is dragged to the other application window divider control, the empty space is filled with the expanded application window and the two application window divider controls merge into a single application window divider control within the user interface. In addition, the two application window divider controls can "snap" into a single application window divider control when the empty space between then narrows to a minimum empty space width.

[0019] Although FIG. 1 illustrates a non-overlapping windowing environment, overlapping windowing environments may also employ an application window divider, whether as a window management system control dividing one application window from another or dividing/bounding an application window relative to an empty space in the user interface or another application window partially occluded by the application window.

[0020] A user can select the application window divider control (e.g., via a touch screen, a pointing device, keyboard input) and drag it right and left along a wide range of relatively continuous locations along the horizontal extent of the user interface. For example, a user can give the application window divider control focus (e.g., by selecting it), and then the user can move the application window divider control by depressing or otherwise activating a keyboard input (e.g., an arrow key) until the application window divider control is at a desired location along the axis. In one implementation, the relatively continuous locations comprise tightly and consistently spaced points along the horizontal axis (e.g., each point separated from an adjacent point by a small number of pixels or some other visually continuous spacing). It should be understood that alternative implementations may provide relatively continuous window divider control along the vertical extent of the user interface or along some other axis. A user interface may also include multiple application window divider controls, some of which may intersect on different axes.

[0021] In one implementation, responsive to the user-induced movement of the application window divider control in a horizontal direction, the user interface provides a visual cue suggesting the continuous relative sizing of the application windows 100 and 102. For example, the application window divider control itself moves, at least one of the application windows appears to resize dynamically, or some other user-friendly suggestion of resizing is animated in the user interface. In one implementation, the application windows may be represented during resizing as snapshots of each window captured at or near the initiation of the directional instruction. In another implementation, the application windows may be represented during resizing as live application windows or other window placeholder representations.

[0022] When the user commits to the resizing (e.g., removes his or her touch, deselects the pointing device button, etc.), then both application windows snap into place at the relative sizes designated by the last location of the application window divider control. In another implementation, responsive to the user-induced movement of the application window divider control in a horizontal direction, the adjacent sides of the application windows 100 and 102 move in visual conformity with the application window divider control. For example, the application window divider moves continuous and both application windows 100 and 102 resize dynamically. When the user commits to the resizing (e.g., removes his or her touch, deselects the pointing device button, etc.), then both application windows remain at the relative sizes designated by the last location of the application window divider control.

[0023] FIG. 2 illustrates a schematic of two application windows (e.g., application window 200 for App A and application window 202 for App B) sharing an application window divider control 204 in an example user interface 206. The application window divider control 204 is shared by the application windows 200 and 202 but is not a component of either application window. The application window divider control 204 is a control of a window management system 208, which manages the display, placement, layout, appearance, and other aspects of the application windows 200 and 202, as well as other windows and user interface characteristics and operations. The window management system 208 manages the user manipulation of the application window divider control 204, including the receipt of user input (e.g., a directional instruction, such as a drag), the mutual resizing of the application windows 200 and 202, and the presentation of the application windows 200 and 202 within the user interface 206.

[0024] As shown in FIG. 2, the application window divider control 204 can be moved along the horizontal axis of the user interface 206 in response to a user-provided directional instruction. Upon completion of the directional instruction (e.g., the user commits to the dragging of the application window divider control by lifting a finger from the touch screen), the application windows 200 and 202 adjust their placement in according to the end point of the direction instruction (e.g., the location of the application window divider control 204 along the axis when the directional instruction is completed). In FIG. 2, this placement adjustment would result in the complimentary resizing of the application windows 200 and 202 based on the final location of the application window divider control 204.

[0025] FIG. 3 illustrates a schematic of two application windows (e.g., application window 300 for App B and application window 302 for App C) sharing an application window divider control 304 and a third application window (e.g., application window 310 for App A) in an example user interface 306. The application window divider control 304 is shared by the application windows 300 and 302 but is not a component of either application window. Another application window divider control 305 is also presented in the user interface 306 and is shared by the application windows 310 and 300. The application window divider controls 304 and 305 are controls of a window management system 308, which manages the display, placement, layout, appearance, and other aspects of the application windows 300, 302, and 310, as well as other windows and user interface characteristics and operations. The window management system 308 manages the user manipulation of the application window divider controls 304 and 305, including the receipt of user input (e.g., a directional instruction, such as a drag), the mutual resizing of the application windows 300 and 302 based on the application window divider control 304, the mutual resizing of the application windows 300 and 310 based on the application window divider control 305, and the presentation of the application windows 300, 302, and 310 within the user interface 306.

[0026] As shown in FIG. 3, the application window divider controls 304 and 305 can be moved along the horizontal axis of the user interface 306 in response to a user-provided directional instruction. Upon completion of the directional instruction (e.g., the user commits to the dragging of the application window divider control by lifting a finger from the touch screen) relating to the application window divider control 304, the window management system 308 adjusts the placements of the application windows 300 and 302 in accordance with the end point of the direction instruction (e.g., the location of the application window divider control 304 along the axis when the directional instruction is completed). In FIG. 3, this placement adjustment would result in the complimentary resizing of the application windows 300 and 302 based on the final location of the application window divider control 304. A similar behavior for application windows 300 and 310 would occur responsive to a directional instruction provided to the application window divider control 305. In addition, each application window divider control 304 and 305 may also impact placement of application windows that it does not share (e.g., divide). For example, in some circumstances, a directional instruction to the application window divider control 304 can cause a placement adjustment to the application window 310, as described below.

[0027] FIG. 4 illustrates an example sequence of user interface operations implementing an application window divider control 400. At 402, a user interface presents application windows for App A, App B, and App C. A directional instruction is provided to the application window divider control 400, moving it in the left direction toward App B and App A and adjusting the placement (e.g., the sizing) of the application window for App B (e.g., decreasing its width) and adjusting the placement (e.g., the sizing) of the application window for App C (e.g., increasing its width). At 404, the placement of the application window for App B has satisfied a minimum size condition 412 – once the width of the application window for App B decreases to a given width threshold, which may be statically or dynamically determined, the width of the application window for App B does not continue to decrease.

[0028] Instead, as shown at 406, the continuing directional instruction causes the application window for App A to adjust its placement (e.g., decreasing its width), while the width of the application window for App B remains at its minimum size condition 412. When the placement of the application window for App A has satisfied its minimum size condition 414, the width of the application window for App A also stops decreasing.

[0029] Accordingly, the continuation of the directional instruction has decreased the widths of the applications windows for App A and App B to their minimum size conditions 414 and 412 respectively. Therefore, at 408, the continuation of the directional instruction causes the application window for App A to be removed from the user interface. In one implementation, the application window for App A is animated to appear to be pushed out of the left edge of the user interface (e.g., in the direction of the directional instruction).

[0030] At 416, the directional instruction continues to cause the application window for App B to be removed from the user interface. In one implementation, the application window for App B is animated to appear to be pushed out of the left edge of the user interface (e.g., in the direction of the directional instruction).

[0031] At each stage in the operational flow in FIG. 4, the directional instruction can be completed (e.g., by the user lifting a finger off of the touch screen, by the user removing a finger from the depressed button of a pointing device, by the user providing an appropriate keyboard input to complete the directional instruction). At any point along the axis in the user interface that an application window divider control is located when the directional instruction is completed, the window management system presents the placement of the application windows in accordance with the placement of the corresponding application window divider controls.

[0032] In one implementation, the window management system records previous window sizes and placements to allow a user to undo one or more resizing operations.

[0033] FIG. 5 illustrates a schematic of two application windows 500 and 502 sharing an application window divider control 504 at a magnetic point 506 in an example user interface 508. The application window divider control 504 is shared by the application windows 500 and 502 but is not a component of either application window. The application window divider control 504 is a control of a window management system 512, which manages the display, placement, layout, appearance, and other aspects of the application windows 500 and 502, as well as other windows and user interface characteristics and operations. The window management system 512 manages the user manipulation of the application window divider control 504, including the receipt of user input (e.g., a directional instruction, such as a drag), the mutual resizing of the application windows 500 and 502, and the presentation of the application windows 500 and 502 within the user interface 508.

[0034] As shown in FIG. 5, the application window divider control 504 can be moved along the horizontal axis of the user interface 508 in response to a user-provided directional instruction. Based a completion of the directional instruction (e.g., the user commits to the dragging of the application window divider control by lifting a finger from the touch screen), the application windows 500 and 502 adjust their placement in according to the end point of the direction instruction (e.g., the location of the application window divider control 504 along the axis when the directional instruction is completed). In FIG. 5, this placement adjustment would result in the complimentary resizing of the application windows 500 and 502 based on the final location of the application window divider control 504.

[0035] In addition, FIG. 5 shows three “magnetic” points along the horizontal axis of the user interface 508, although any number of magnetic points is contemplated. Each magnetic point represents a position on the horizontal axis of the user interface 508 to which the application window divider control 504 is pulled when, during a directional instruction, the movement of the application window divider control 504 satisfies a speed condition (e.g., movement slows below a speed threshold) and/or a distance condition (e.g., movement brings the application window divider 504 within a given distance threshold 510 from the magnetic point). For example, as the application window divider control 504 is moved from the left to the right across the horizontal axis of the user interface 508, the user can slow the movement near the magnetic point 506, which can cause the application window divider control 504 to snap to its location shown as 504'.

[0036] The location of the magnetic points may be determined statically for a set of visible application windows or dynamically during a directional instruction. For example, a statically determined magnetic point may be located at the center of the horizontal axis when two application windows are visible or magnetic points may be placed at thirds along the horizontal axis when three application windows are visible. In contrast, the location of some magnetic points may be determined dynamically during a directional instruction. For example, responsive to initiation of the directional instruction, the window management system can calculate the locations of any magnetic points along the axis, and in some implementations, the locations may be dynamically updated as the directional instruction continues to be executed, subject to application window placement preferences and other constraints. Eventually, when the directional instruction is committed, the application window divider controls will snap to the dynamically calculated locations of the magnetic points.

[0037] The term “snap” describes moving the application window divider control 504 to align with the magnetic point 506 along the axis when the application window divider control 504 is not initially aligned with the magnetic point 506 but the movement of the application window divider control 504 satisfies a speed condition and a distance condition relative to the magnetic point 506. Visually, in one implementation, this snapping operation appears as a relatively rapid movement or jump by the application window divider control 504 to align with the magnetic point 506. Although, in alternative implementations, the snapping operation can be presented through the user interface 508 in other ways (e.g., the snapping of window boundaries, the rapid movement of ghost images of application windows, etc.).

[0038] It should be understood that various types of magnetic points may be employed. In one example, magnetic points may be positioned to maintain consistent horizontal and/or sizing of application windows (e.g., each of three windows have the same width). In another example, minimum and maximum windows sizes, user-defined size preferences, and system defined window preferences may also be considered in the placement of magnetic points along an axis within the user interface.

[0039] In one implementation, one or more application window divider controls may be automatically snapped to appropriate magnetic points via a double-click, a double-tap, or a key combination on or in relation to one of the application window divider controls. Such user input can instruct one or more application window divider controls to move in an appropriate direction to snap to the nearest magnetic point, subject to application window placement preferences and other constraints. In one implementation, repeating the user input on the same application window divider control causes the application window divider controls of the visible application windows within the user interface to move in appropriate directions to snap to the nearest magnetic points, subject to application window placement preferences, and other constraints.

[0040] FIG. 6 illustrates example operations 600 for adjusting placement of two application windows using an application window divider control. A presenting operation 602 presents an application window divider control shared by two application windows in a user interface. In one implementation, the application window divider control separates or acts as a boundary between the two application windows. An instruction operation 604 receives a directional instruction (e.g., a drag gesture) in association with the application window divider control. An adjusting operation 606 adjusts the relative placement of the two application windows based on the directional instruction applied to the application window divider control. For example, one application window gets narrower and the other application window gets wider within the user interface as the application window divider control moves or after its movement is completed.

[0041] A decision operation 608 determines whether one of the application windows has been sized to satisfy a minimum size condition (note: different application windows can have different minimum size conditions). If not, the adjusting operation 606 continues processing the directional instruction. If the decision operation 608 determines whether one of the application windows has satisfied a minimum size condition, that application window stops getting narrower, and another decision operation 610 determines whether another application window is located in the direction of the directional instruction within the user interface. If not, the application window satisfying the minimum size condition is removed from the user interface in a removal operation 612. Otherwise, another adjusting operation 610 adjusts the placement of the other application window based on the directional instruction. In one implementation, the visual effect of the adjusting operation 610 resembles the first minimum window freezing at its minimum width and starting to push the side of the next window so that the next window gets narrower.

[0042] It should be understood that the next window may also reach a width that satisfies its minimum width condition. Accordingly, if the directional instruction continues in the same direction, then the next minimum window may be removed from the user interface in a manner similar to that discussed with regard to the removal operation 612. Further, subsequent to removal of the next minimum window and subject to a continued directional instruction in the same direction, the first minimum window may be removed in a manner similar to that discussed with regard to the removal operation 612.

[0043] FIG. 7 illustrates example operations 700 for operating an application window divider control relative to a magnetic point in an example user interface. A presenting operation 702 presents an application window divider control shared by two application windows in a user interface. In one implementation, the application window divider control separates or acts as a boundary between the two application windows. An instruction operation 704 receives a directional instruction (e.g., a drag gesture) in association with the application window divider control. An adjusting operation 706 adjusts the relative placement of the two application windows based on the directional instruction applied to the application window divider control.

[0044] A decision operation 708 determines whether movement of the application window divider control satisfies a speed condition and/or a distance condition relative to a magnetic point located along an axis of the user interface. For example, the decision operation 708 may determine that the speed of the application window divider control movement is below a given speed threshold. In another example, the decision operation 708 may determine that the distance between the application window divider control and the magnetic point is below a given distance threshold. In yet another example, the decision operation 708 tests both the speed condition and the distance condition to proceed to a snap operation 710. If the appropriate conditions are not satisfied in the decision operation 708, the adjusting operation 706 continues to adjust the relative placement of the two application windows.

[0045] If the appropriate condition or conditions are satisfied in the decision operation 708, the snap operation 710 snaps the application window divider control to the magnetic point in the user interface. In one implementation, the snap operation 710 presents a rapid movement by the application window divider control along the axis from a location that is not aligned with the magnetic point to a location that is aligned with the magnetic point.

[0046] It should be understood that such directional instructions, magnetic points, and application window divider controls may be performed, manipulated, or positioned along any axis of the user interface, including without limitation a horizontal axis or a vertical axis.

[0047] FIG. 8 illustrates an example system that may be useful in implementing the described technology. The example hardware and operating environment of FIG. 8 for implementing the described technology includes a computing device, such as general purpose computing device in the form of a gaming console or computer 20, a mobile telephone, a personal data assistant (PDA), a set top box, or other type of computing device. In the implementation of FIG. 8, for example, the computer 20 includes a processing unit 21, a system memory 22, and a system bus 23 that operatively couples various system components including the system memory to the processing unit 21. There may be only one or there may be more than one processing unit 21, such that the processor of computer 20 comprises a single central-processing unit (CPU), or a plurality of processing units, commonly referred to as a parallel processing environment. The computer 20 may be a conventional computer, a distributed computer, or any other type of computer; the implementations are not so limited.

[0048] The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, a switched fabric, point-to-point connections, and a local bus using any of a variety of bus architectures. The system memory may also be referred to as simply the memory, and includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the computer 20, such as during start-up, is stored in ROM 24. The computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM, DVD, or other optical media.

[0049] The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their associated tangible computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for the computer 20. It should be appreciated by those skilled in the art that any type of tangible computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROMs), and the like, may be used in the example operating environment.

[0050] A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone (e.g., for voice input), a camera (e.g., for a natural user interface (NUI)), a joystick, a game pad, a satellite dish, a scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or a universal serial bus (USB). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, computers typically include other peripheral output devices (not shown), such as speakers and printers.

[0051] The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computer 49. These logical connections are achieved by a communication device coupled to or a part of the computer 20; the implementations are not limited to a particular type of communications device. The remote computer 49 may be another computer, a server, a router, a network PC, a client, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20, although only a memory storage device 50 has been illustrated in FIG. 8. The logical connections depicted in FIG. 8 include a local-area network (LAN) 51 and a wide-area network (WAN) 52. Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the Internet, which are all types of networks.

[0052] When used in a LAN-networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53, which is one type of communications device. When used in a WAN-networking environment, the computer 20 typically includes a modem 54, a network adapter, a type of communications device, or any other type of communications device for establishing communications over the wide area network 52. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program engines depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It is appreciated that the network connections shown are example and other means of and communications devices for establishing a communications link between the computers may be used.

[0053] In an example implementation, software or firmware instructions and data for providing a window management system, a magnetic point, an application window divider control, and other instructions and data may be stored in memory 22 and/or storage devices 29 or 31 and processed by the processing unit 21. The user interface data, speed thresholds, distance thresholds, and other data may be stored in memory 22 and/or storage devices 29 or 31 as persistent datastores.

[0054] Some embodiments may comprise an article of manufacture. An article of manufacture may comprise a tangible storage medium to store logic. Examples of a storage medium may include one or more types of computer-readable storage media capable of storing electronic data, including volatile memory or non-volatile memory, removable or non-removable memory, erasable or non-erasable memory, writeable or re-writeable memory, and so forth. Examples of the logic may include various software elements, such as software components, programs, applications, computer programs, application programs, system programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, application program interfaces (API), instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof. In one embodiment, for example, an article of manufacture may store executable computer program instructions that, when executed by a computer, cause the computer to perform methods and/or operations in accordance with the described embodiments. The executable computer program instructions may include any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, and the like. The executable computer program instructions may be implemented according to a predefined computer language, manner or syntax, for instructing a computer to perform a certain function. The instructions may be implemented using any suitable high-level, low-level, object-oriented, visual, compiled and/or interpreted programming language.

[0055] The implementations described herein are implemented as logical steps in one or more computer systems. The logical operations may be implemented (1) as a sequence of processor-implemented steps executing in one or more computer systems and (2) as interconnected machine or circuit modules within one or more computer systems. The implementation is a matter of choice, dependent on the performance requirements of the computer system being utilized. Accordingly, the logical operations making up the implementations described herein are referred to variously as operations, steps, objects, or modules. Furthermore, it should be understood that logical operations may be performed in any order, unless explicitly claimed otherwise or a specific order is inherently necessitated by the claim language.

[0056] The above specification, examples, and data provide a complete description of the structure and use of exemplary implementations. Since many implementations can be made without departing from the spirit and scope of the claimed invention, the claims hereinafter appended define the invention. Furthermore, structural features of the different examples may be combined in yet another implementation without departing from the recited claims.

CLAIMS

1. A method comprising:

presenting an application window divider control shared by a first application window and a second application window in a user interface; and

adjusting placement of the first application window and the second application window across a range of consistently spaced points along an axis of the user interface, responsive to a received directional instruction that moves the application window divider control along the axis.

2. The method of claim 1 wherein a minimum size condition is associated with the first application window and further comprising:

removing the first application window from the user interface when the adjusting operation decreases the placement of the first application window to satisfy the minimum size condition.

3. The method of claim 2 wherein the first application window is removed from the user interface with an animation that pushes the first application window out of the user interface in the direction of the directional instruction.

4. The method of claim 1 wherein the presenting operation further presents a third application window that does not share the application window divider control, the third application window being placed between the first window and an edge of the user interface in the direction of the directional instruction, and the adjusting operation further comprises:

adjusting the placement of the first application window to satisfy a minimum size condition of the first application window, responsive to the directional instruction; and

adjusting placement of the third application window after the placement of the first application window satisfies the minimum size condition of the first application window, responsive to the directional instruction.

5. The method of claim 4 wherein a minimum size condition is associated with the third application window and further comprising:

removing the third application window from the user interface when the operation of adjusting the third application window decreases the placement of the third application window to satisfy the minimum size condition of the third application window.

6. The method of claim 5 wherein the third application window is removed from the user interface with an animation that pushes the third application window out of the user interface in the direction of the directional instruction.

7. The method of claim 1 further comprising:

snapping the application window divider control to a point along the axis of the user interface if movement of the application window divider control satisfies a speed condition or a distance condition relative to the point.

8. The method of claim 7 wherein the snapping operation comprises:

moving the application window divider control to align with the point along the axis when the application window divider control is not aligned with the point but the movement of the application window divider control satisfies the speed condition and the distance condition relative to the point.

9. One or more tangible computer-readable storage media encoding computer-executable instructions for executing on a computer system a computer process, the computer process comprising:

presenting an application window divider control shared by a first application window and a second application window in a user interface; and

adjusting placement of the first application window and the second application window across a range of consistently spaced points along an axis of the user interface, responsive to a received directional instruction that moves the application window divider control along the axis.

10. A system comprising:

a computing device presenting a user interface;

a window management system configured to adjust placement of a first application window and the second application window across a range of consistently spaced points along an axis of a user interface, responsive to a received directional instruction that moves a visible application window divider control shared by the first application window and the second application window along the axis.

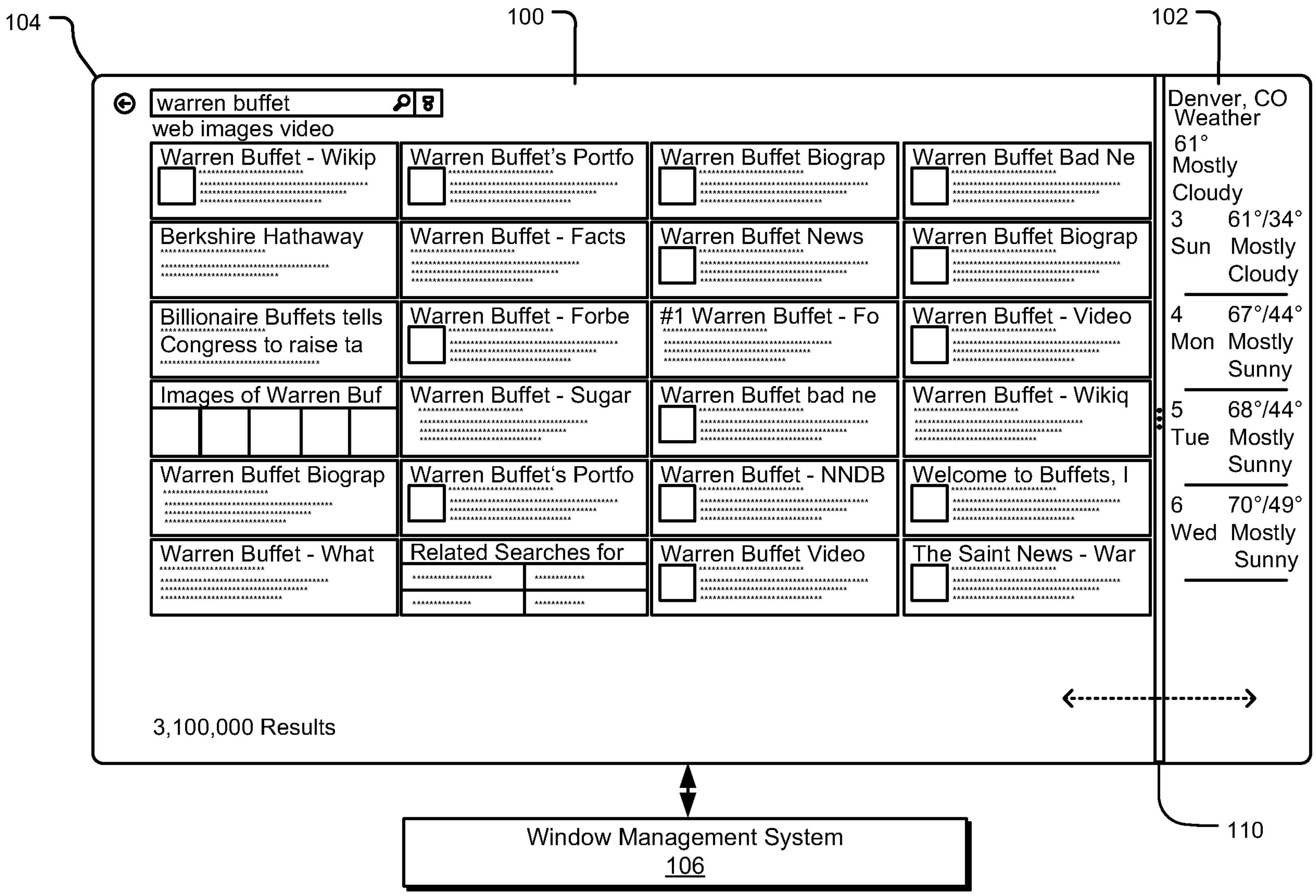

104
100
102
warren buffet
web images video
Warren Buffet - Wikip
Warren Buffet's Portfo
Warren Buffet Biograp
Warren Buffet Bad Ne
Berkshire Hathaway
Warren Buffet - Facts
Warren Buffet News
Warren Buffet Biograp
Billionaire Buffets tells Congress to raise ta
Warren Buffet - Forbe
#1 Warren Buffet - Fo
Warren Buffet - Video
Images of Warren Buf
Warren Buffet - Sugar
Warren Buffet bad ne
Warren Buffet - Wikiq
Warren Buffet Biograp
Warren Buffet's Portfo
Warren Buffet - NNDB
Welcome to Buffets, I
Warren Buffet - What
Related Searches for
Warren Buffet Video
The Saint News - War
3,100,000 Results
Denver, CO Weather
61° Mostly Cloudy
3 Sun 61°/34° Mostly Cloudy
4 Mon 67°/44° Mostly Sunny
5 Tue 68°/44° Mostly Sunny
6 Wed 70°/49° Mostly Sunny
110
Window Management System
106

FIG. 1

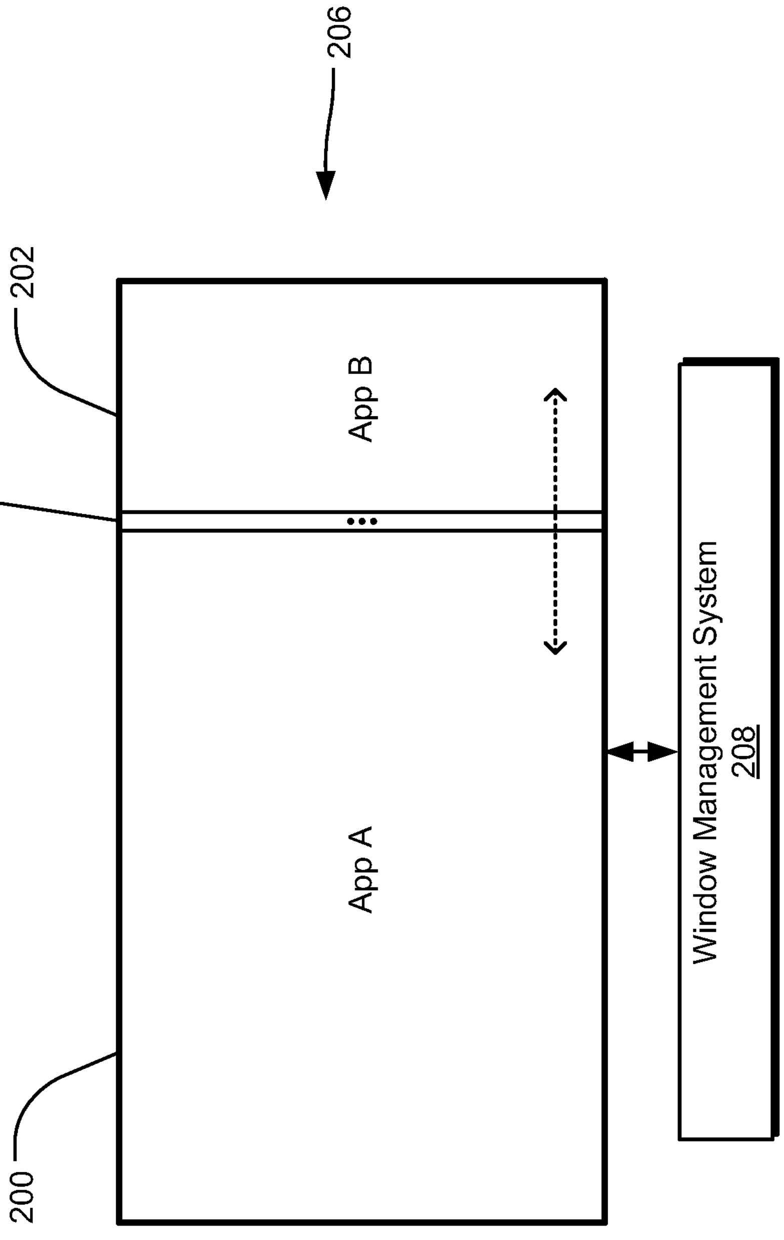


FIG. 2

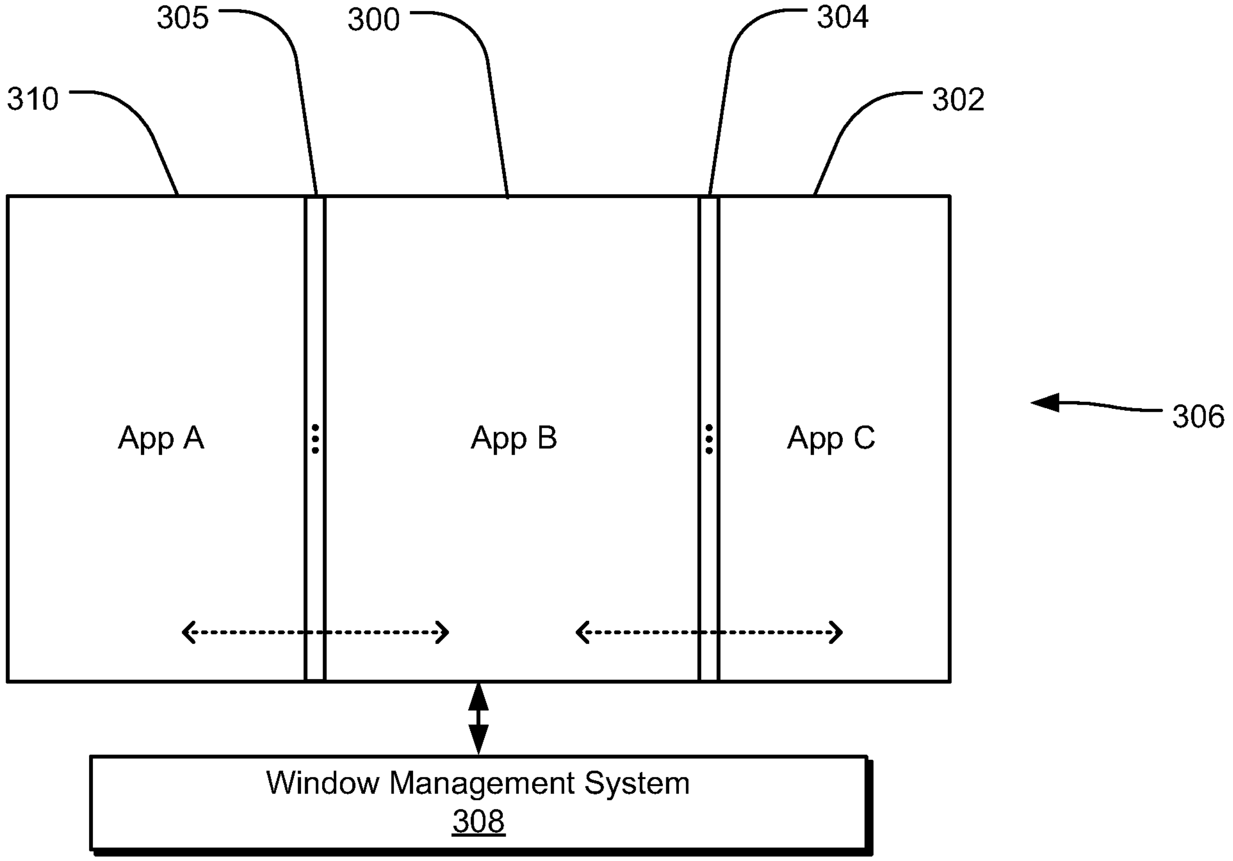

305
300
304
310
302
App A
App B
App C
306
Window Management System
308

FIG. 3

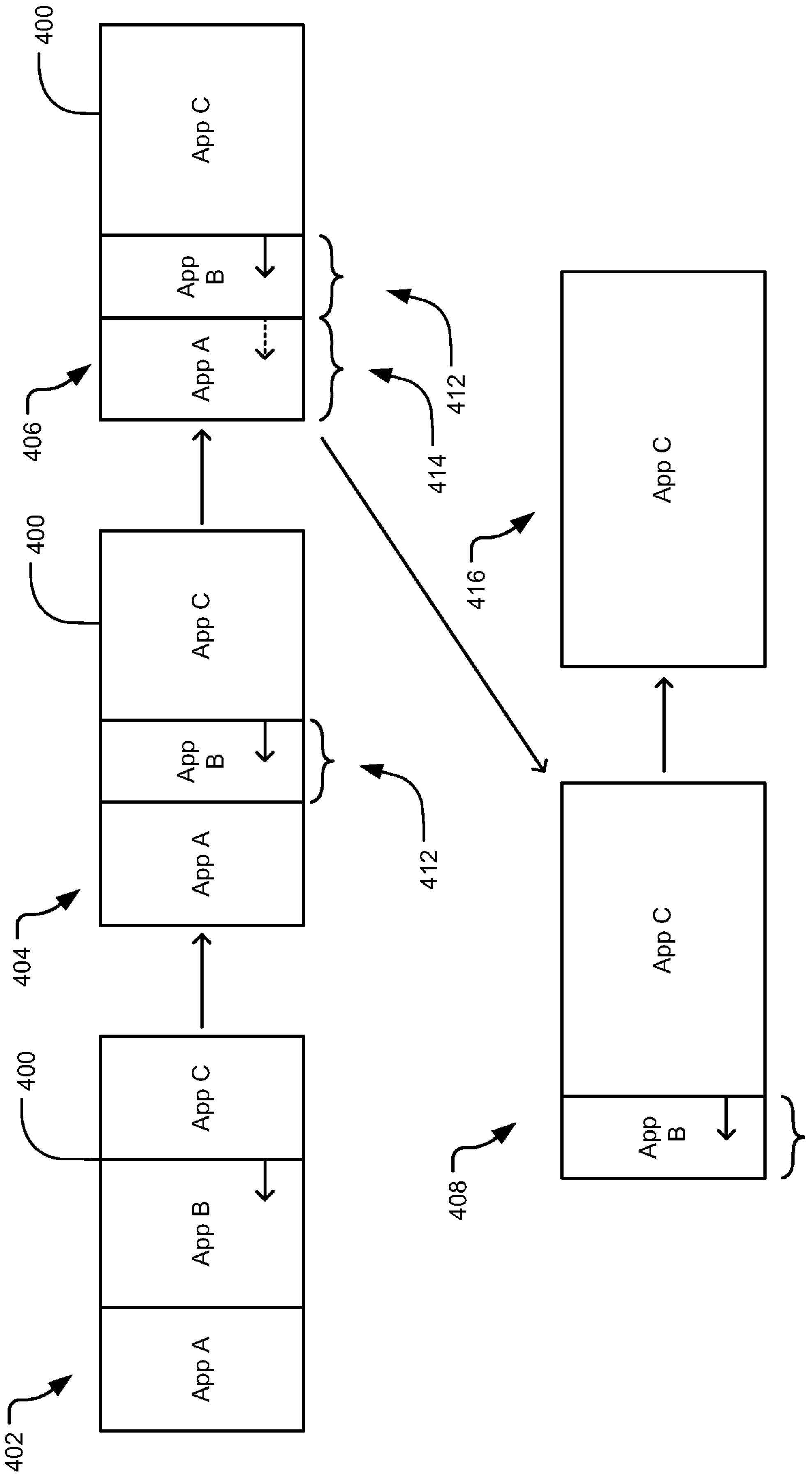

402
App A
App B
App C
400
404
App A
App B
App C
412
400
406
App A
App B
App C
414
412
400
408
App B
App C
416
App C

FIG. 4

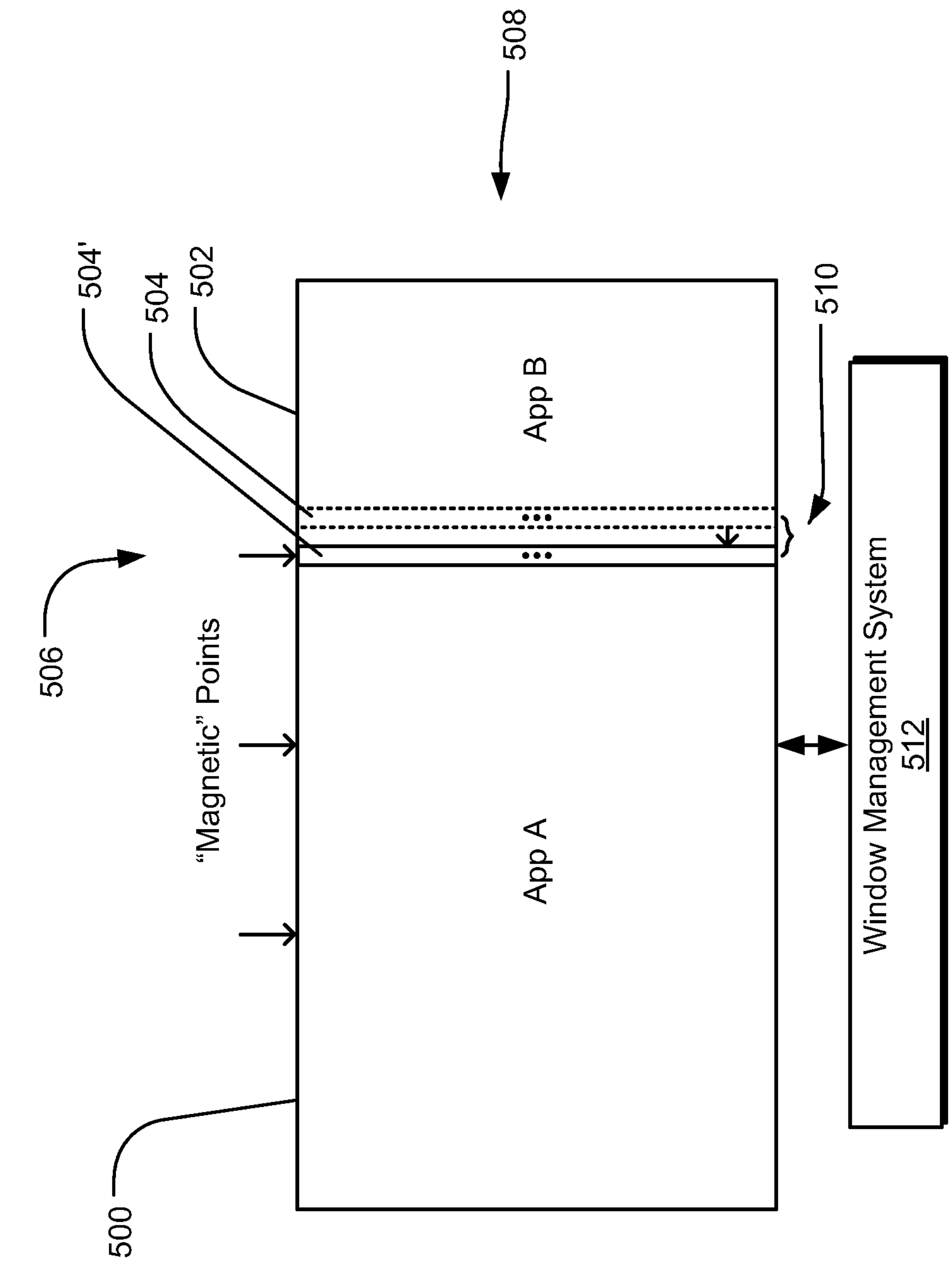

508
504'
504
502
510
App B
506
"Magnetic" Points
App A
Window Management System
512
500

FIG. 5

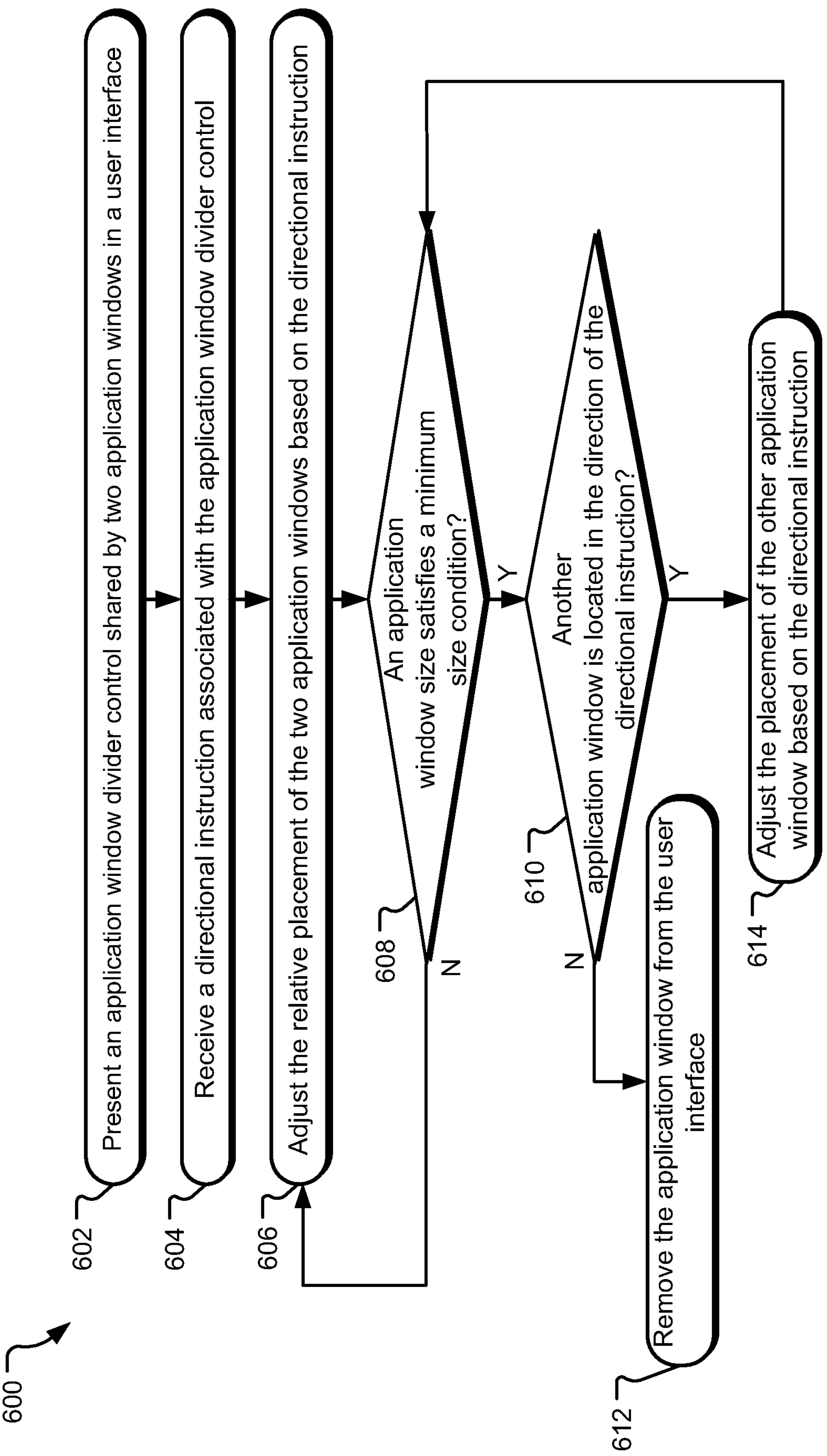

600
602
Present an application window divider control shared by two application windows in a user interface
604
Receive a directional instruction associated with the application window divider control
606
Adjust the relative placement of the two application windows based on the directional instruction
608
An application window size satisfies a minimum size condition?
Y
N
610
Another application window is located in the direction of the directional instruction?
Y
N
612
Remove the application window from the user interface
614
Adjust the placement of the other application window based on the directional instruction

FIG. 6

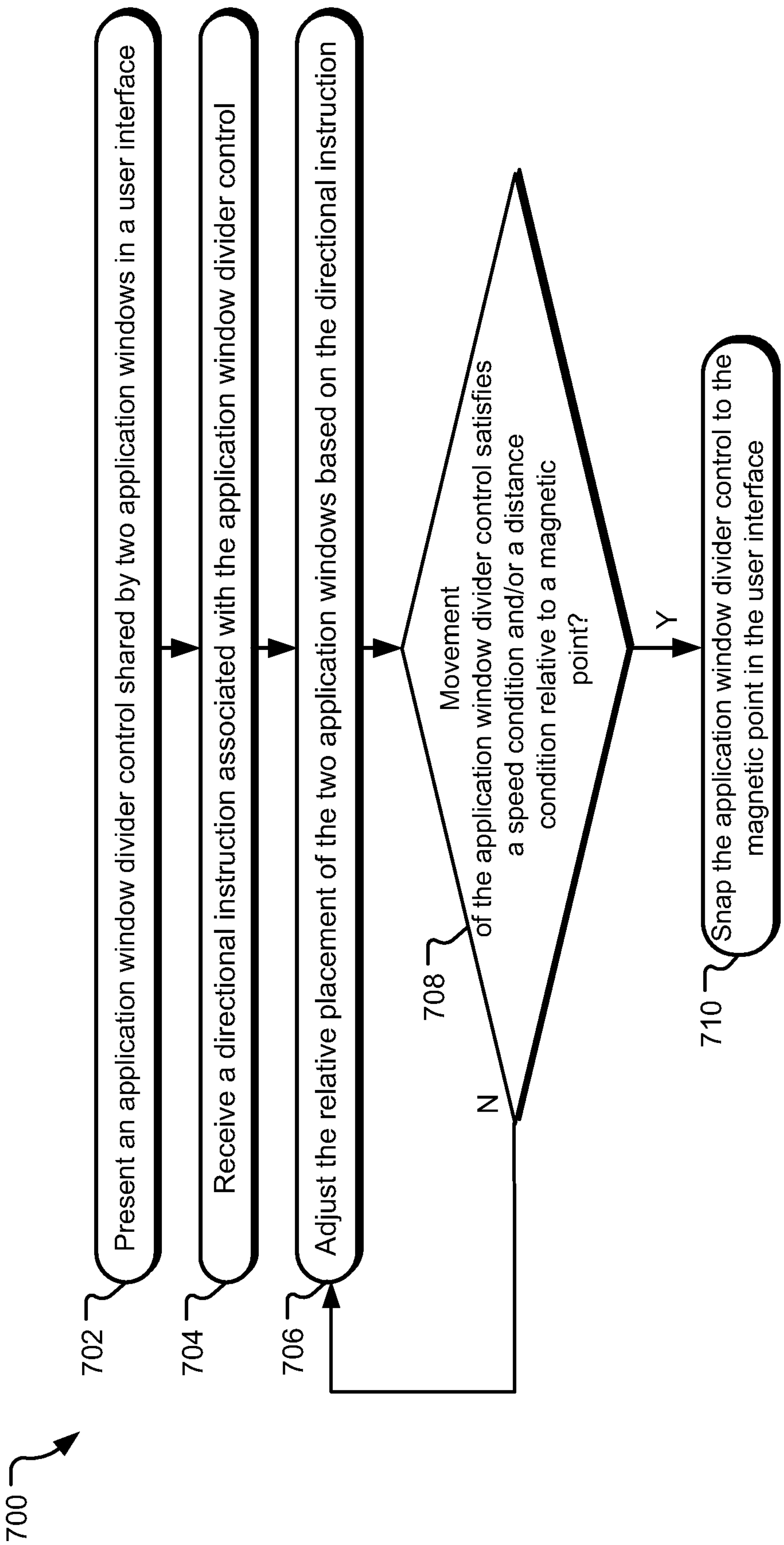

700
702
Present an application window divider control shared by two application windows in a user interface
704
Receive a directional instruction associated with the application window divider control
706
Adjust the relative placement of the two application windows based on the directional instruction
708
Movement of the application window divider control satisfies a speed condition and/or a distance condition relative to a magnetic point?
N
Y
710
Snap the application window divider control to the magnetic point in the user interface

FIG. 7

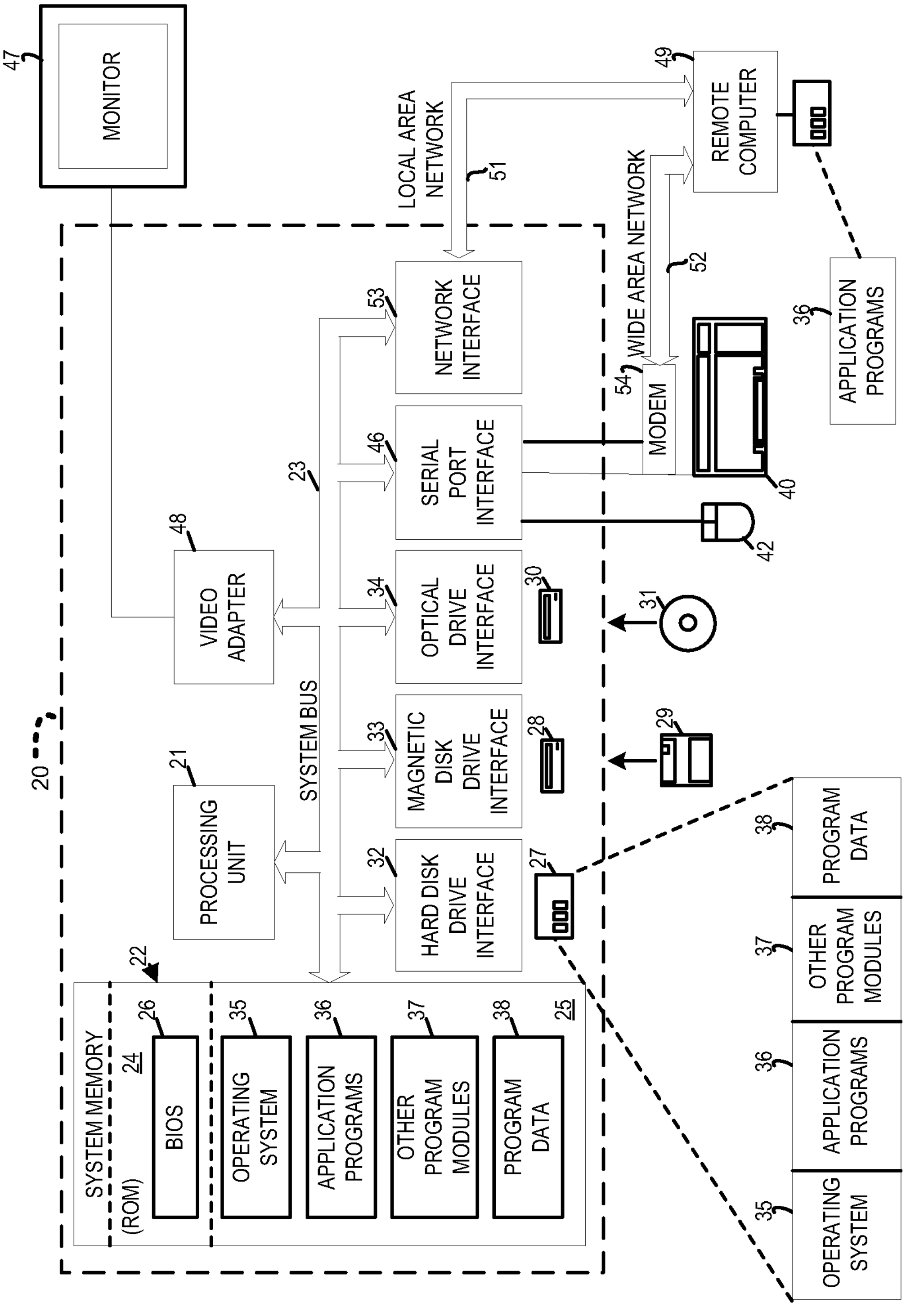

20
SYSTEM MEMORY
(ROM)
24
26
BIOS
22
35
OPERATING SYSTEM
36
APPLICATION PROGRAMS
37
OTHER PROGRAM MODULES
38
PROGRAM DATA
25
21
PROCESSING UNIT
SYSTEM BUS
23
48
VIDEO ADAPTER
47
MONITOR
32
HARD DISK DRIVE INTERFACE
33
MAGNETIC DISK DRIVE INTERFACE
34
OPTICAL DRIVE INTERFACE
46
SERIAL PORT INTERFACE
53
NETWORK INTERFACE
27
28
29
30
31
40
42
54
MODEM
LOCAL AREA NETWORK
51
WIDE AREA NETWORK
52
49
REMOTE COMPUTER
36
APPLICATION PROGRAMS
35
OPERATING SYSTEM
36
APPLICATION PROGRAMS
37
OTHER PROGRAM MODULES
38
PROGRAM DATA

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/060247

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44 G06F3/048
ADD.

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

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

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

EPO-Internal, WPI Data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/166980 A1 (YOSEF NADAV HAR EL [IL] ET AL) 28 June 2012 (2012-06-28)	1-10
Y	paragraph [0001] - paragraph [0004] paragraph [0020]	7,8
X	US 7 890 882 B1 (NELSON JOHN [US]) 15 February 2011 (2011-02-15)	1-10
Y	column 2, line 47 - column 5, line 35; figure 1B	7,8
Y	US 2002/191028 A1 (SENECHALLE DAVID A [US] ET AL) 19 December 2002 (2002-12-19) paragraph [0009]	7,8

□ 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 another 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 2 December 2013	Date of mailing of the international search report 10/12/2013
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Lo Turco, Salvatore

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/060247

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012166980 A1	28-06-2012	NONE	
US 7890882 B1	15-02-2011	NONE	
US 2002191028 A1	19-12-2002	NONE	

(19) 中华人民共和国国家知识产权局

(12) 发明专利申请

(10) 申请公布号 CN 105190544 A
(43) 申请公布日 2015.12.23

(21) 申请号 201380075620.5

(22) 申请日 2013.09.18

(30) 优先权数据
13/863369 2013.04.15 US

(85) PCT国际申请进入国家阶段日
2015.10.15

(86) PCT国际申请的申请数据
PCT/US2013/060247 2013.09.18

(87) PCT国际申请的公布数据
WO2014/171963 EN 2014.10.23

(71) 申请人 微软技术许可有限责任公司
地址 美国华盛顿州

(72) 发明人 M. 弗兰耶斯 J.C. 萨特菲尔德
M.I. 沃利 C. 萨里恩 N.A. 森德林
A. 马拉尼 A. 施泰因格拉斯
R.J. 贾雷特

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

代理人 陈慧 景军平

(51) Int.Cl.
G06F 9/44(2006.01)
G06F 3/048(2013.01)

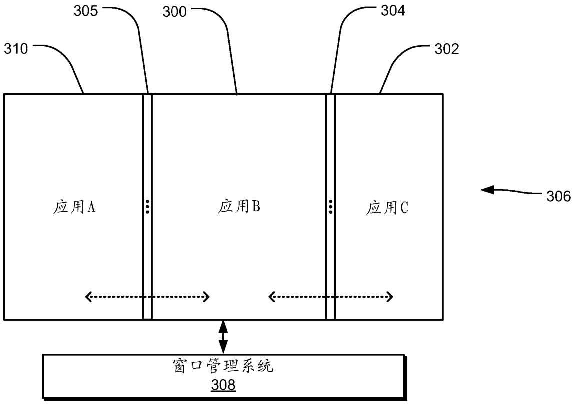
权利要求书1页 说明书8页 附图8页

(54) 发明名称

用于窗口布局管理的应用窗口分隔器控件

(57) 摘要

应用窗口分隔器控件由用户接口中的第一应用窗口和第二应用窗口所共享。基于沿着用户接口的轴线移动应用窗口分隔器控件的所接收的定向指令，可以横跨沿着用户接口的轴线的一系列一致间隔的点来定位第一应用窗口和第二应用窗口的放置。放置调整还可能影响用户接口中的其它应用窗口。应用窗口分隔器控件还可以在满足某些条件时对齐以提供沿着轴线的“磁力”点。



1. 一种方法，包括：

呈现由用户接口中的第一应用窗口和第二应用窗口所共享的应用窗口分隔器控件；以及

响应于沿着用户接口的轴线移动应用窗口分隔器控件的所接收的定向指令，横跨沿着所述轴线的一系列一致间隔的点来调整第一应用窗口和第二应用窗口的放置。

2. 权利要求 1 的方法，其中最小尺寸条件与第一应用窗口相关联，并且还包括：

在调整操作降低第一应用窗口的放置以满足最小尺寸条件时从用户接口移除第一应用窗口。

3. 权利要求 2 的方法，其中利用在定向指令的方向上将第一应用窗口从用户接口推出的动画来从用户接口移除第一应用窗口。

4. 权利要求 1 的方法，其中呈现操作还呈现不共享应用窗口分隔器控件的第三应用窗口，第三应用窗口放置在定向指令的方向上的用户接口的边缘与第一窗口之间，并且调整操作还包括：

响应于定向指令，调整第一应用窗口的放置以满足第一应用窗口的最小尺寸条件；以及

响应于定向指令，在第一应用窗口的放置满足第一应用窗口的最小尺寸条件之后，调整第三应用窗口的放置。

5. 权利要求 4 的方法，其中最小尺寸条件与第三应用窗口相关联，并且还包括：

在调整第三应用窗口的操作降低第三应用窗口的放置以满足第三应用窗口的最小尺寸条件时，从用户接口移除第三应用窗口。

6. 权利要求 5 的方法，其中利用在定向指令的方向上将第三应用窗口从用户接口推出的动画来从用户接口移除第三应用窗口。

7. 权利要求 1 的方法，还包括：

如果应用窗口分隔器控件的运动满足相对于沿着用户接口的轴线的点的速度条件或距离条件，则使应用窗口分隔器控件对齐到所述点。

8. 权利要求 7 的方法，其中对齐操作包括：

在应用窗口分隔器控件不与沿着轴线的点对准、但是应用窗口分隔器控件的运动满足相对于所述点的速度条件和距离条件时，移动应用窗口分隔器控件以与所述点对准。

9. 编码计算机可执行指令以用于在计算机系统上执行计算机进程的一个或多个有形计算机可读存储介质，所述计算机进程包括：

呈现由用户接口中的第一应用窗口和第二应用窗口所共享的应用窗口分隔器控件；以及

响应于沿着用户接口的轴线移动应用窗口分隔器控件的所接收的定向指令，横跨沿着所述轴线的一系列一致间隔的点来调整第一应用窗口和第二应用窗口的放置。

10. 一种系统，包括：

呈现用户接口的计算设备；

窗口管理系统，其被配置成响应于沿着用户接口的轴线移动第一应用窗口和第二应用窗口所共享的可见应用窗口分隔器控件的所接收的定向指令，横跨沿着所述轴线的一系列一致间隔的点来调整第一应用窗口和第二应用窗口的放置。

用于窗口布局管理的应用窗口分隔器控件

背景技术

[0001] 包括其对应用用户接口的计算机操作系统允许用户调整应用窗口以用于通过用户接口呈现给用户。然而，这样的用户引导的调整具有不便利的限制，特别是随着用户接口变得更为先进和灵活。

发明内容

[0002] 本文所描述和要求保护的实现方案通过提供由用户接口中的第一应用窗口和第二应用窗口所共享的应用窗口分隔器控件来解决前述问题。基于沿着用户接口的轴线移动应用窗口分隔器控件的所接收的定向指令，第一应用窗口和第二应用窗口的放置可以定位成横跨沿着用户接口的轴线的一系列一致间隔点。放置调整还可能影响用户接口中的其它应用窗口。应用窗口分隔器控件还可以对齐以在满足某些条件时提供沿着轴线的“磁力”点。

[0003] 提供本发明内容来以简化形式引入以下在具体实施方式中进一步描述的概念的选择。本发明内容不旨在标识所要求保护的主题的关键特征或必要特征，也不旨在用于限制所要求保护的主题的范围。

[0004] 本文还描述和陈述了其它实现方案。

附图说明

[0005] 图1图示了共享示例用户接口屏幕中的应用窗口分隔器控件的两个应用窗口。

[0006] 图2图示了共享示例用户接口中的应用窗口分隔器控件的两个应用窗口的示意图。

[0007] 图3图示了共享示例用户接口中的应用窗口分隔器控件和第三应用窗口的两个应用窗口的示意图。

[0008] 图4图示了实现应用窗口分隔器控件的用户接口操作的示例序列。

[0009] 图5图示了共享示例用户接口中磁力点处的应用窗口分隔器控件的两个应用窗口的示意图。

[0010] 图6图示了用于使用应用窗口分隔器控件调整两个应用窗口的放置的示例操作。

[0011] 图7图示了用于相对于示例用户接口中的磁力点操作应用窗口分隔器控件的示例操作。

[0012] 图8图示了可能有助于实现所描述的技术的示例系统。

具体实施方式

[0013] 启动应用窗口包括但不限于最初执行应用并且切换到已经执行的应用的新的或隐藏的应用窗口。此外，应用窗口可以包括但不限于操作系统组件、操作系统实体设施和专用应用程序（例如网络浏览器程序、文字处理程序、工作表程序）的窗口。

[0014] 图1图示了共享示例用户接口屏幕104中的应用窗口分隔器控件110的两个应用

窗口100和102。窗口管理系统106典型地是操作系统或用户接口环境的组件，但是还可以是独立的应用。窗口管理系统106管理应用窗口100和102的显示、放置、布局、外观和其它方面，以及其它窗口和用户接口特性和操作。

[0015] 应用窗口100呈现显示来自网络搜索引擎的搜索结果的搜索结果窗口。各个搜索结果（诸如搜索结果108）包括代表性文本和/或一个或多个图像连同浏览器可导航链接，其可以由用户选择以用于导航到由浏览器可导航链接所标识的网站。应用窗口102呈现来自天气应用或网站的结果，包括Denver, CO的4天天气预报。窗口边界控件110分隔应用窗口100的显示区域和应用窗口102的显示区域，在所图示的情形中，分离应用窗口。在一些实现方案中，窗口边界控件110可以由用户操纵以改变一个或两个窗口的尺寸和/或位置。

[0016] 在图1中将应用窗口100和102示出为非重叠窗口。然而，所描述的技术不限于非重叠窗口环境。例如，在这样的环境中，重叠窗口环境中的每一个应用窗口可以通过窗口边界控件来定界，该控件可以由用户操纵以改变潜在重叠窗口的位置、尺寸和从前到后次序（笼统地，“放置”）（例如，窗口的z排序）。窗口管理系统106通过应用窗口100和102并且通过其它用户接口组件（例如键盘接口、触摸屏接口、语音接口和定点设备）接收输入，并且通过用户接口显示应用窗口100和102、窗口边界控件110和其它应用窗口和控件。

[0017] 窗口边界控件110表示用户接口中两个不同应用窗口110和102之间或者应用窗口与空白空间之间的分隔器，并且因此还可以称为“应用窗口分隔器控件”。应用窗口分隔器控件可以通过用户接口来操纵以将应用窗口的尺寸（相对于空白空间）的改变或应用窗口100和102的相对尺寸的改变传达给窗口管理系统106。因此，应用窗口分隔器控件的功能性表示提供一个或多个应用窗口的外部边界的窗口管理系统控件，而不是作为一个应用窗口或另一个的组件，并且在一些配置中可以作为用户接口和窗口管理系统106的一部分而由两个或更多应用窗口所共享。

[0018] 在一个实现方案中，两个或更多应用窗口分隔器控件可以合并在一起以形成单个应用窗口分隔器控件。例如，两个应用窗口可以沿着用户接口的水平轴线呈现，通过空白空间分离。在该情况下，利用应用窗口和空白空间之间的应用窗口分隔器控件显示每一个应用。如果一个应用窗口分隔器控件被拖拽到另一应用窗口分隔器控件，则空白空间填充有扩展的应用窗口，并且两个应用窗口分隔器控件合并成用户接口内的单个应用窗口分隔器控件。此外，，两个应用窗口分隔器控件在它们之间的空白空间变窄成最小空白空间宽度时、可以“对齐”成单个应用窗口分隔器控件。

[0019] 尽管图1图示了非重叠窗口环境，但是重叠窗口环境也可以采用应用窗口分隔器，不管是窗口管理系统控件将一个应用窗口与另一个分隔、还是相对于用户接口中的空白空间或由应用窗口部分阻挡的另一应用窗口对应用窗口进行分隔/定界。

[0020] 用户可以选择应用窗口分隔器控件（例如经由触摸屏、定点设备、键盘输入）并且顺着沿用户接口的水平范围的一大系列相对连续位置左右拖拽它。例如，用户可以关注应用窗口分隔器控件（例如通过选择它），并且然后用户可以通过压下或以其它方式激活键盘输入（例如箭头按键）来移动应用窗口分隔器控件，直到应用窗口分隔器控件处于沿轴线的期望位置处。在一个实现方案中，相对连续位置包括沿着水平轴线的紧密且一致间隔的点（例如，每一个点通过少量像素或某种其它视觉上连续的间距与相邻点分离）。应当理解到，

可替换的实现方案可以提供沿着用户接口的竖直范围或者沿着某一其它轴线的相对连续窗口分隔器控件。用户接口还可以包括多个应用窗口分隔器控件，其中一些可以在不同轴线上相交。

[0021] 在一些实现方案中，响应于应用窗口分隔器控件在水平方向上的用户引发的运动，用户接口提供建议应用窗口 100 和 102 的连续相对尺寸的视觉提示。例如，应用窗口分隔器控件自身移动，至少一个应用窗口看起来动态地重新定尺寸，或者在用户接口中动画化重新定尺寸的某种其它用户友好的建议。在一个实现方案中，应用窗口可以在重新定尺寸期间表示为在定向指令的发起处或附近所捕获的每一个窗口的截屏。在另一实现方案中，应用窗口可以在重新定尺寸期间表示为活动的应用窗口或其它窗口占位符表示。

[0022] 当用户致力于重新定尺寸时（例如移除他或她的触碰、去选择（deselect）定点设备按钮等），然后两个应用窗口对齐到由应用窗口分隔器控件的最后位置所指定的相对尺寸处的位置。在另一实现方案中，响应于应用窗口分隔器控件在水平方向上的用户引发的运动，应用窗口 100 和 102 的相邻侧在视觉上与应用窗口分隔器控件一致地运动。例如，应用窗口分隔器连续运动，并且两个应用窗口 100 和 102 动态地重新定尺寸。当用户致力于重新定尺寸时（例如移除他或她的触碰、去选择定点设备按钮等），然后两个应用窗口保留在由应用窗口分隔器控件的最后位置所指定的相对尺寸处。

[0023] 图 2 图示了共享示例用户接口 206 中的应用窗口分隔器控件 204 的两个应用窗口（例如用于应用 A 的应用窗口 200 和用于应用 B 的应用窗口 202）的示意图。应用窗口分隔器控件 204 被应用窗口 200 和 202 所共享，但不是任一个应用窗口的组件。应用窗口分隔器控件 204 是窗口管理系统 208 的控件，其管理应用窗口 200 和 202 的显示、放置、布局、外观和其它方面，以及其它窗口和用户接口特性和操作。窗口管理系统 208 管理应用窗口分隔器控件 204 的用户操纵，包括用户输入（例如定向指令，诸如拖拽）的接收、应用窗口 200 和 202 的相互重新定尺寸、以及用户接口 206 内的应用窗口 200 和 202 的呈现。

[0024] 如图 2 中所示，应用窗口分隔器控件 204 可以响应于用户提供的定向指令而沿着用户接口 206 的水平轴线移动。当完成定向指令（例如用户致力于通过从触摸屏提起手指而拖拽应用窗口分隔器控件）之后，应用窗口 200 和 202 根据定向指令的终点调整其放置（例如在定向指令完成时，应用窗口分隔器控件 204 沿着轴线的位置）。在图 2 中，该放置调整将导致应用窗口 200 和 202 基于应用窗口分隔器控件 204 的最终位置的互补重新定尺寸。

[0025] 图 3 图示了共享示例用户接口 306 中的应用窗口分隔器控件 304 和第三应用窗口（例如用于应用 A 的应用窗口 310）的两个应用窗口（例如用于应用 B 的应用窗口 300 和用于应用 C 的应用窗口 302）的示意图。应用窗口分隔器控件 304 被应用窗口 300 和 302 所共享，但不是任一个应用窗口的组件。另一应用窗口分隔器控件 305 也呈现在用户接口 306 中并且被应用窗口 310 和 300 所共享。应用窗口分隔器控件 304 和 305 是窗口管理系统 308 的控件，其管理应用窗口 300,302 和 310 的显示、放置、布局、外观和其它方面，以及其它窗口和用户接口特性和操作。窗口管理系统 308 管理应用窗口分隔器控件 304 和 305 的用户操纵，包括用户输入（例如定向指令，诸如拖拽）的接收、应用窗口 300 和 302 基于应用窗口分隔器控件 304 的相互重新定尺寸、应用窗口 300 和 310 基于应用窗口分隔器控件 305 的相互重新定尺寸、以及应用窗口 300,302 和 310 在用户接口 306 内的呈现。

[0026] 如图3中所示，应用窗口分隔器控件304和305可以响应于用户提供的定向指令而沿着用户接口306的水平轴线运动。在完成涉及应用窗口分隔器控件304的定向指令(例如用户致力于通过从触摸屏提起手指而拖拽应用窗口分隔器控件)之后，窗口管理系统308依照定向指令的终点调整应用窗口300和302的放置(例如在定向指令完成时应用窗口分隔器控件304沿着轴线的位置)。在图3中，该放置调整将导致应用窗口300和302基于应用窗口分隔器控件304的最终位置的互补重新定尺寸。针对应用窗口300和310的类似行为将响应于提供给应用窗口分隔器控件305的定向指令而发生。此外，每一个应用窗口分隔器控件304和305还可能影响其不共享(例如分隔)的应用窗口的放置。例如，在一些情况下，到应用窗口分隔器控件304的定向指令可以引起对应用窗口310的放置调整，如下文描述的。

[0027] 图4图示了实现应用窗口分隔器控件400的用户接口操作的示例序列。在402处，用户接口呈现用于应用A、应用B和应用C的应用窗口。将定向指令提供给应用窗口分隔器控件400，从而将其在左方向上朝向应用B和应用A移动，并且调整用于应用B的应用窗口的放置(例如定尺寸)(例如减小其宽度)以及调整用于应用C的应用窗口的放置(例如定尺寸)(例如增加其宽度)。在404处，用于应用B的应用窗口的放置已经满足最小尺寸条件412——一旦用于应用B的应用窗口的宽度减小到给定宽度阈值的话，这可以静态地或动态地确定，那么用于应用B的应用窗口的宽度不会连续减小。

[0028] 相反，如在406处所示，继续定向指令使得用于应用A的应用窗口调整其放置(例如减小其宽度)，而同时用于应用B的应用窗口的宽度保持在其最小尺寸条件412处。当用于应用A的应用窗口的放置已经满足其最小尺寸条件414时，用于应用A的应用窗口的宽度也停止减小。

[0029] 因此，定向指令的继续已经分别将用于应用A和应用B的应用窗口的宽度减小到其最小尺寸条件414和412。因此，在408处，定向指令的继续使得用于应用A的应用窗口从用户接口移除。在一个实现方案中，用于应用A的应用窗口被动画化成看起来被推出用户接口的左边缘(例如在定向指令的方向上)。

[0030] 在416处，定向指令继续使得用于应用B的应用窗口从用户接口移除。在一个实现方案中，用于应用B的应用窗口被动画化成看起来被推出用户接口的左边缘(例如在定向指令的方向上)。

[0031] 在图4中的操作流中的每一个阶段处，定向指令可以完成(例如，通过用户从触摸屏提起手指、通过用户从定点设备的压下按钮移除手指、通过用户提供适当的键盘输入以完成定向指令)。在定向指令完成时应用窗口分隔器控件所位于的用户接口中沿轴线的任何点处，窗口管理系统依照对应应用窗口分隔器控件的放置呈现应用窗口的放置。

[0032] 在一个实现方案中，窗口管理系统记录之前的窗口尺寸和放置以允许用户撤消一个或多个重新定尺寸操作。

[0033] 图5图示了共享示例用户接口508中的磁力点506处的应用窗口分隔器控件504的两个应用窗口500和502的示意图。应用窗口分隔器控件504被应用窗口500和502所共享，但不是任一个应用窗口的组件。应用窗口分隔器控件504是窗口管理系统512的控件，其管理应用窗口500和502的显示、放置、布局、外观和其它方面，以及其它窗口和用户接口特性和操作。窗口管理系统512管理应用窗口分隔器控件504的用户操纵，包括用户

输入(例如定向指令,诸如拖拽)的接收、应用窗口 500 和 502 的相互重新定尺寸、以及应用窗口 500 和 502 在用户接口 508 内的呈现。

[0034] 如图 5 中所示,应用窗口分隔器控件 504 可以响应于用户提供的定向指令而沿着用户接口 508 的水平轴线运动。基于定向指令的完成(例如用户致力于通过从触摸屏提起手指而拖拽应用窗口分隔器控件),应用窗口 500 和 502 根据定向指令的终点调整其放置(例如在完成定向指令时,应用窗口分隔器控件 504 沿着轴线的位置)。在图 5 中,该放置调整将导致应用窗口 500 和 502 基于应用窗口分隔器控件 504 的最终位置的互补重新定尺寸。

[0035] 此外,图 5 示出了沿着用户接口 508 的水平轴线的三个“磁力”点,当然预期到任何数目的磁力点。每一个磁力点表示在定向指令期间当应用窗口分隔器控件 504 的运动满足速度条件(例如运动减慢到速度阈值以下)和/或距离条件(例如运动使应用窗口分隔器 504 处于距磁力点的给定距离阈值 510 内)时应用窗口分隔器控件 504 被拉到的用户接口 508 的水平轴线上的位置。例如,当应用窗口分隔器控件 504 横跨用户接口 508 的水平轴线从左向右运动时,用户可以在磁力点 506 附近减慢运动,其可以使得应用窗口分隔器控件 504 对齐到其如 504’所示的位置。

[0036] 可以针对一组可变应用窗口静态地或者在定向指令期间动态地确定磁力点的位置。例如,静态确定的磁力点可以在两个应用窗口可见时位于水平轴线的中心处,或者磁力点可以在三个应用窗口可见时放置于沿着水平轴线的第三个处。相比而言,一些磁力点的位置可以在定向指令期间动态地确定。例如,响应于定向指令的发起,窗口管理系统可以计算沿着轴线的任何磁力点的位置,并且在一些实现方案中,位置可以随着定向指令继续执行而动态地更新,经受应用窗口放置偏好和其它约束。最终,当定向指令被提交时,应用窗口分隔器控件将对齐到磁力点的动态计算的位置。

[0037] 术语“对齐”描述移动应用窗口分隔器控件 504 以在应用窗口分隔器控件 504 最初不与磁力点 506 对准但是应用窗口分隔器控件 504 的移动满足相对于磁力点 506 的速度条件和距离条件时与沿着轴线的磁力点 506 对准。视觉上,在一个实现方案中,该对齐操作看起来是应用窗口分隔器控件 504 的相对快速的运动或跳动以与磁力点 506 对准。当然,在可替换实现方案中,对齐操作可以通过用户接口 508 以其它方式呈现(例如窗口边界的对齐、应用窗口的鬼像的快速运动等)。

[0038] 应当理解到,可以采用各种类型的磁力点。在一个示例中,磁力点可以定位成维持应用窗口的一致水平和/或尺寸(例如三个窗口中的每一个具有相同宽度)。在另一示例中,在磁力点沿着用户接口内的轴线的放置方面还可以考虑到最小和最大窗口尺寸、用户限定的尺寸偏好以及系统限定的窗口偏好。

[0039] 在一个实现方案中,一个或多个应用窗口分隔器控件可以经由双击、双拍(double-tap)、或者一个应用窗口分隔器控件上或关于其的按键组合而自动对齐到适当的磁力点。这样的用户输入可以指令一个或多个应用窗口分隔器控件在适当方向上运动以对齐到最近磁力点,经受应用窗口放置偏好和其它约束。在一个实现方案中,重复相同应用窗口分隔器控件上的用户输入使得用户接口内的可见应用窗口的应用窗口分隔器控件在适当方向上运动以对齐到最近磁力点,经受应用窗口放置偏好和其它约束。

[0040] 图 6 图示了用于使用应用窗口分隔器控件调整两个应用窗口的放置的示例操作

600。呈现操作 602 呈现由用户接口中的两个应用窗口所共享的应用窗口分隔器控件。在一个实现方案中，应用窗口分隔器控件分离两个应用窗口或者用作两个应用窗口之间的边界。指令操作 604 与应用窗口分隔器控件相关联地接收定向指令(例如拖拽手势)。调整操作 606 基于应用于应用窗口分隔器控件的定向指令而调整两个应用窗口的相对放置。例如，当应用窗口分隔器控件移动时或者在其移动完成之后，在用户接口内，一个应用窗口变得更窄并且另一个应用窗口变得更宽。

[0041] 决定操作 608 确定一个应用窗口的尺寸是否已经满足最小尺寸条件(注意：不同应用窗口可以具有不同最小尺寸条件)。如果否，则调整操作 606 继续处理定向指令。如果决定操作 608 确定一个应用窗口是否已经满足最小尺寸条件，则该应用窗口停止变得更窄，并且另一决定操作 610 确定另一应用窗口是否位于用户接口内的定向指令的方向上。如果否，则满足最小尺寸条件的应用窗口在移除操作 612 中从用户接口移除。否则，另一调整操作 610 基于定向指令调整另一应用窗口的放置。在一个实现方案中，调整操作 610 的视觉效果类似于第一最小窗口冻结在其最小宽度处并且开始推动下一窗口的侧边使得下一窗口变得更窄。

[0042] 应当理解到，下一窗口也可以达到满足其最小宽度条件的宽度。因此，如果定向指令在相同方向上继续，则下一最小窗口可以以类似于关于移除操作 612 所讨论的方式从用户接口移除。另外，紧接着下一最小窗口的移除并且经受相同方向上的继续的定向指令，第一最小窗口可以以类似于关于移除操作 612 所讨论的方式移除。

[0043] 图 7 图示了用于相对于示例用户接口中的磁力点来操作应用窗口分隔器控件的示例操作 700。呈现操作 702 呈现由用户接口中的两个应用窗口所共享的应用窗口分隔器控件。在一个实现方案中，应用窗口分隔器控件分离两个应用窗口或者用作两个应用窗口之间的边界。指令操作 704 与应用窗口分隔器控件相关联地接收定向指令(例如拖拽手势)。调整操作 706 基于应用于应用窗口分隔器控件的定向指令而调整两个应用窗口的相对放置。

[0044] 决定操作 708 确定应用窗口分隔器控件的运动是否满足相对于沿着用户接口的轴线定位的磁力点的速度条件和 / 或距离条件。例如，决定操作 708 可以确定应用窗口分隔器控件运动的速度在给定速度阈值以下。在另一示例中，决定操作 708 可以确定应用窗口分隔器控件和磁力点之间的距离在给定距离阈值以下。在又一示例中，决定操作 708 测试速度条件和距离条件二者以进行到对齐操作 710。如果在决定操作 708 中没有满足适当的条件，则调整操作 706 继续调整两个应用窗口的相对放置。

[0045] 如果在决定操作 708 中满足一个或多个适当条件，则对齐操作 710 使应用窗口分隔器控件对齐到用户接口中的磁力点。在一个实现方案中，对齐操作 710 呈现应用窗口分隔器控件沿着轴线从不与磁力点对准的位置到与磁力点对准的位置的快速运动。

[0046] 应当理解到，这样的定向指令、磁力点和应用窗口分隔器控件可以沿着用户接口的任何轴线执行、操纵或定位，包括但不限于水平轴线或竖直轴线。

[0047] 图 8 图示了可以有助于实现所描述的技术的示例系统。用于实现所描述的技术的图 8 的示例硬件和操作环境包括计算设备，诸如以游戏控制台或计算机 20 的形式的通用计算设备、移动电话、个人数字助理(PDA)、机顶盒或其它类型的计算设备。在图 8 的实现方案中，例如，计算机 20 包括处理单元 21、系统存储器 22、以及将包括系统存储器的各种系统总

线操作性地耦合到处理单元21的系统总线23。可能仅存在一个处理单元21或者可能存在不止一个处理单元21，使得计算机20的处理器包括单个中央处理单元（CPU）或多个处理单元，其共同地被称为并行处理环境。计算机20可以是常规计算机、分布式计算机、或者任何其它类型的计算机；实现方案不限于此。

[0048] 系统总线23可以是任何若干类型的总线结构，包括存储器总线或存储器控制器、外围总线、开关构造、点对点连接、以及使用任何各种总线架构的局部总线。系统存储器还可以简单地被称为存储器，并且包括只读存储器（ROM）24和随机存取存储器（RAM）25。包含诸如在启动期间帮助在计算机20内的元件之间传递信息的基本例程的基本输入／输出系统（BIOS）26存储在ROM 24中。计算机20还包括用于从硬盘（未示出）读取并向其写入的硬盘驱动器27、用于从可移除磁盘29读取或向其写入的磁盘驱动器28、以及用于从可移除光盘31读取或向其写入的光盘驱动器30，所述光盘31诸如CD ROM、DVD或其它光学介质。

[0049] 硬盘驱动器27、磁盘驱动器28和光盘驱动器30分别通过硬盘驱动器接口32、磁盘驱动器接口33和光盘驱动器接口34连接到系统总线23。驱动器及其相关联的有形计算机可读介质为计算机20提供计算机可读指令、数据结构、程序模块和其它数据的非易失性存储。本领域技术人员应当领会到，可以存储数据的任何类型有形计算机可读介质是计算机可访问的，诸如盒式磁带、闪速存储器卡、数字视频盘、随机存取存储器（RAM）、只读存储器（ROM）等，其可以用在示例操作环境中。

[0050] 数个程序模块可以存储在硬盘、磁盘29、光盘31、ROM 24或RAM 25上，包括操作系统35、一个或多个应用程序36、其它程序模块37和程序数据38。用户可以通过诸如键盘40和定点设备42之类的输入设备将命令和信息录入到个人计算机20中。其它输入设备（未示出）可以包括麦克风（例如用于语音输入）、相机（例如用于自然用户界面（NUI））、游戏手柄、游戏垫、圆盘式天线、扫描仪等。这些和其它输入设备常常通过耦合到系统总线的串行端口接口46连接至处理单元21，但是可以通过诸如并行端口、游戏端口或通用串行总线（USB）之类的其它接口进行连接。监视器47或其它类型显示设备也经由诸如视频适配器48之类的接口连接到系统总线23。除监视器之外，计算机典型地包括其它外围输出设备（未示出），诸如扬声器和打印机。

[0051] 计算机20可以使用到一个或多个远程计算机（诸如远程计算机49）的逻辑连接而操作于联网环境中。这些逻辑连接通过耦合到计算机20或者其一部分的通信设备来实现；实现方案不限于特定类型的通信设备。远程计算机49可以是另一计算机、服务器、路由器、网络PC、客户端、对等设备或其它常见网络节点，并且典型地包括以上相对于计算机20所描述的许多或全部元件，尽管在图8中已经图示了仅存储器存储设备50。图8中所描绘的逻辑连接包括局域网（LAN）51和广域网（WAN）52。这样的联网环境在作为所有类型网络的办公室网络、企业范围的计算机网络、以太网和因特网中是无处不在的。

[0052] 当在LAN联网环境中使用时，计算机20通过作为一种类型通信设备的适配器53或网络接口连接到本地网络51。当在WAN联网环境中使用时，计算机20典型地包括调制解调器54、网络适配器、某种类型通信设备、或者用于通过广域网52建立通信的任何其它类型通信设备。可以在内部或外部的调制解调器54经由串行端口接口46连接到系统总线23。在联网环境中，相对于个人计算机20所描绘的程序引擎或其部分可以存储在远程存储

器存储设备中。领会到，所示出的网络连接是示例，并且可以使用用于建立计算机之间的通信链接的其它措施和通信设备。

[0053] 在示例实现方案中，用于提供窗口管理系统、磁力点、应用窗口分隔器控件的软件或固件指令和数据，以及其它指令和数据可以存储在存储器 22 和 / 或存储设备 29 或 31 中并且由处理单元 21 处理。用户接口数据、速度阈值、距离阈值和其它数据可以存储在作为持久数据仓库的存储器 22 和 / 或存储设备 29 或 31 中。

[0054] 一些实施例可以包括制造品。制造品可以包括存储逻辑的有形存储介质。存储介质的示例可以包括能够存储电子数据的一个或多个类型的计算机可读存储介质，包括易失性存储器或非易失性存储器、可移除或不可移除存储器、可擦除或不可擦除存储器、可写入或可再写存储器等。逻辑的示例可以包括各种软件元件，诸如软件组件、程序、应用、计算机程序、应用程序、系统程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子例程、功能、方法、进程、软件接口、应用程序接口（API）、指令集、计算代码、计算机代码、代码片段、计算机代码片段、文字、值、符号、或者其任何组合。在一个实施例中，例如，制造品可以存储可执行计算机程序指令，其在由计算机运行时使计算机执行依照所描述的实施例的方法和 / 或操作。可执行计算机程序指令可以包括任何适当类型的代码，诸如源代码、编译代码、解译代码、可执行代码、静态代码、动态代码等。可执行计算机程序指令可以根据预限定的计算机语言、方式或句法来实现以用于指令计算机执行某种功能。指令可以使用任何适当的高级、低级、面向对象、视觉、编译和 / 或解译编程语言来实现。

[0055] 本文所描述的实现方案被实现为一个或多个计算机系统中的逻辑步骤。逻辑操作可以实现为（1）在一个或多个计算机系统中执行的处理器实现的步骤的序列和（2）一个或多个计算机系统内的互连机器或电路模块。实现方案是一个选择的问题，这取决于所利用的计算机系统的性能要求。因此，形成本文所描述的实现方案的逻辑操作被不同地称为操作、步骤、对象或模块。此外，应当理解到，逻辑操作可以以任何顺序执行，除非以其它方式明确要求保护或者通过权利要求语言使特定顺序固有地成为必需。

[0056] 以上说明书、示例和数据提供示例性实现方案的结构和使用的完整描述。由于在不脱离所要求保护的发明的精神和范围的情况下可以做出许多实现方案，在此之后随附的权利要求限定本发明。此外，不同示例的结构特征可以在又一实现方案中组合而不脱离所陈述的权利要求。

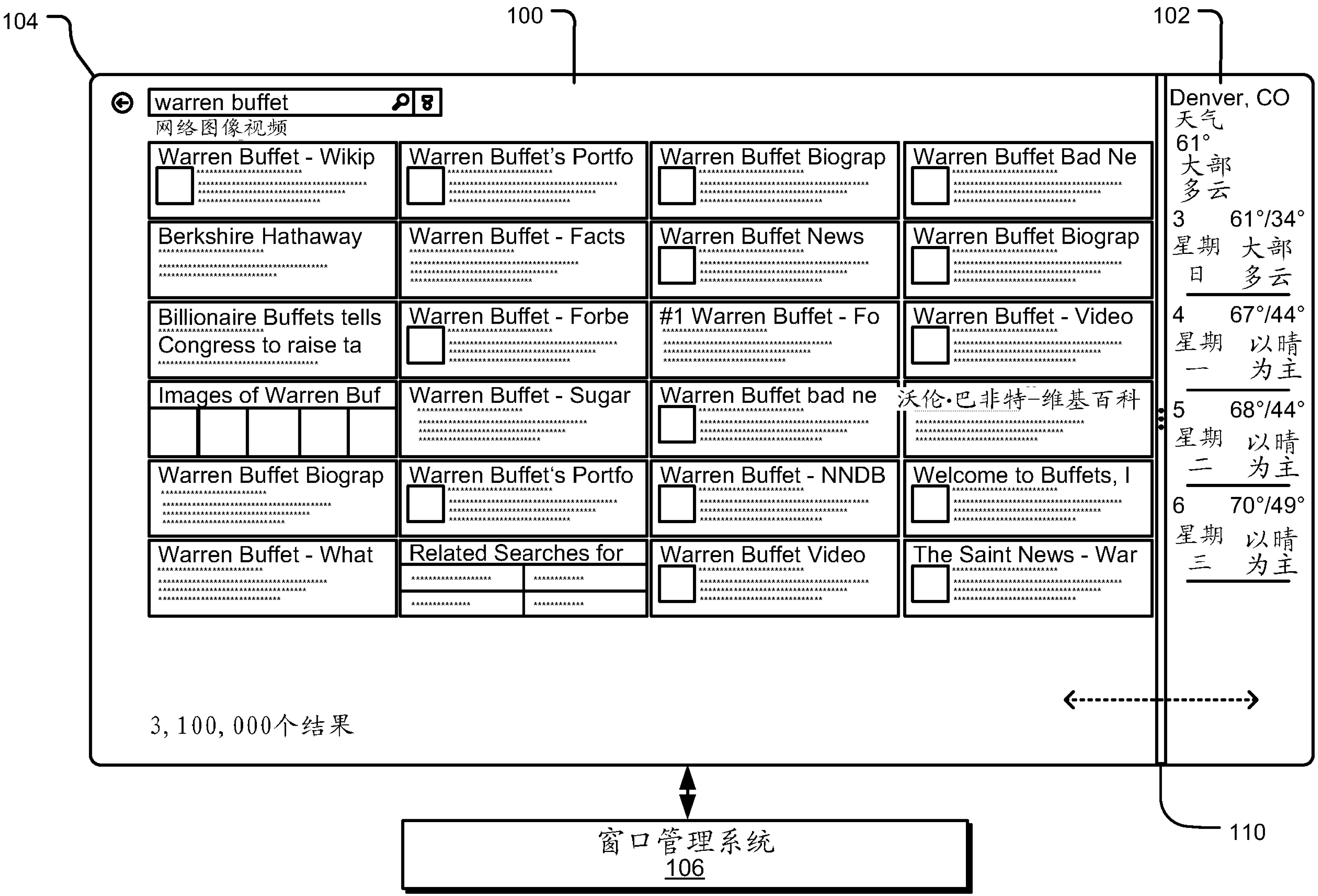

104
100
102
warren buffet
网络图像视频
Warren Buffet - Wikip
Warren Buffet's Portfo
Warren Buffet Biograp
Warren Buffet Bad Ne
Berkshire Hathaway
Warren Buffet - Facts
Warren Buffet News
Warren Buffet Biograp
Billionaire Buffets tells Congress to raise ta
Warren Buffet - Forbe
#1 Warren Buffet - Fo
Warren Buffet - Video
Images of Warren Buf
Warren Buffet - Sugar
Warren Buffet bad ne
沃伦·巴菲特–维基百科
Warren Buffet Biograp
Warren Buffet's Portfo
Warren Buffet - NNDB
Welcome to Buffets, I
Warren Buffet - What
Related Searches for
Warren Buffet Video
The Saint News - War
3,100,000个结果
Denver, CO
天气
61°
大部多云
3 61°/34°
星期日 大部多云
4 67°/44°
星期一 以晴为主
5 68°/44°
星期二 以晴为主
6 70°/49°
星期三 以晴为主
110
窗口管理系统
106

图 1

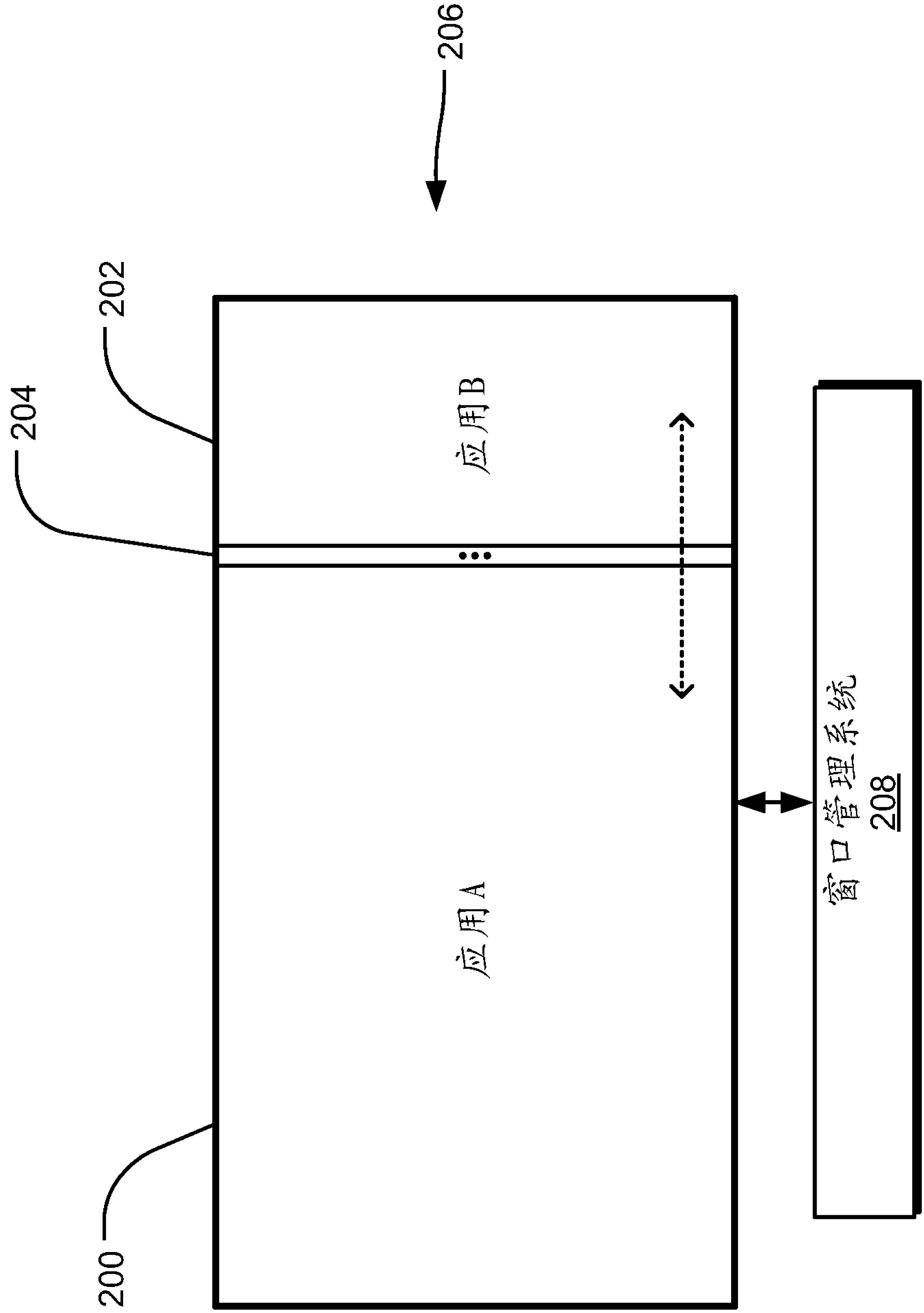


图 2

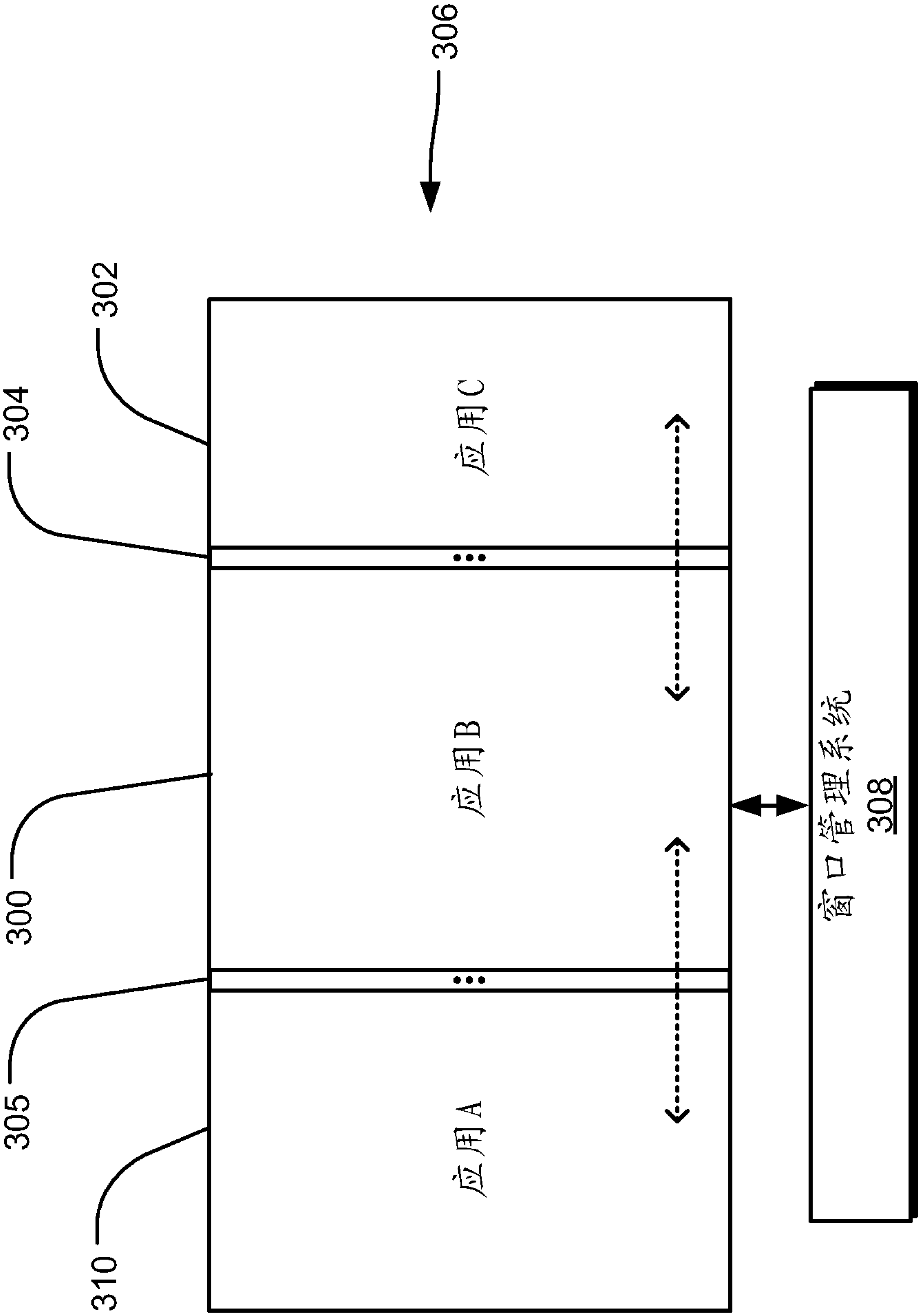


图 3

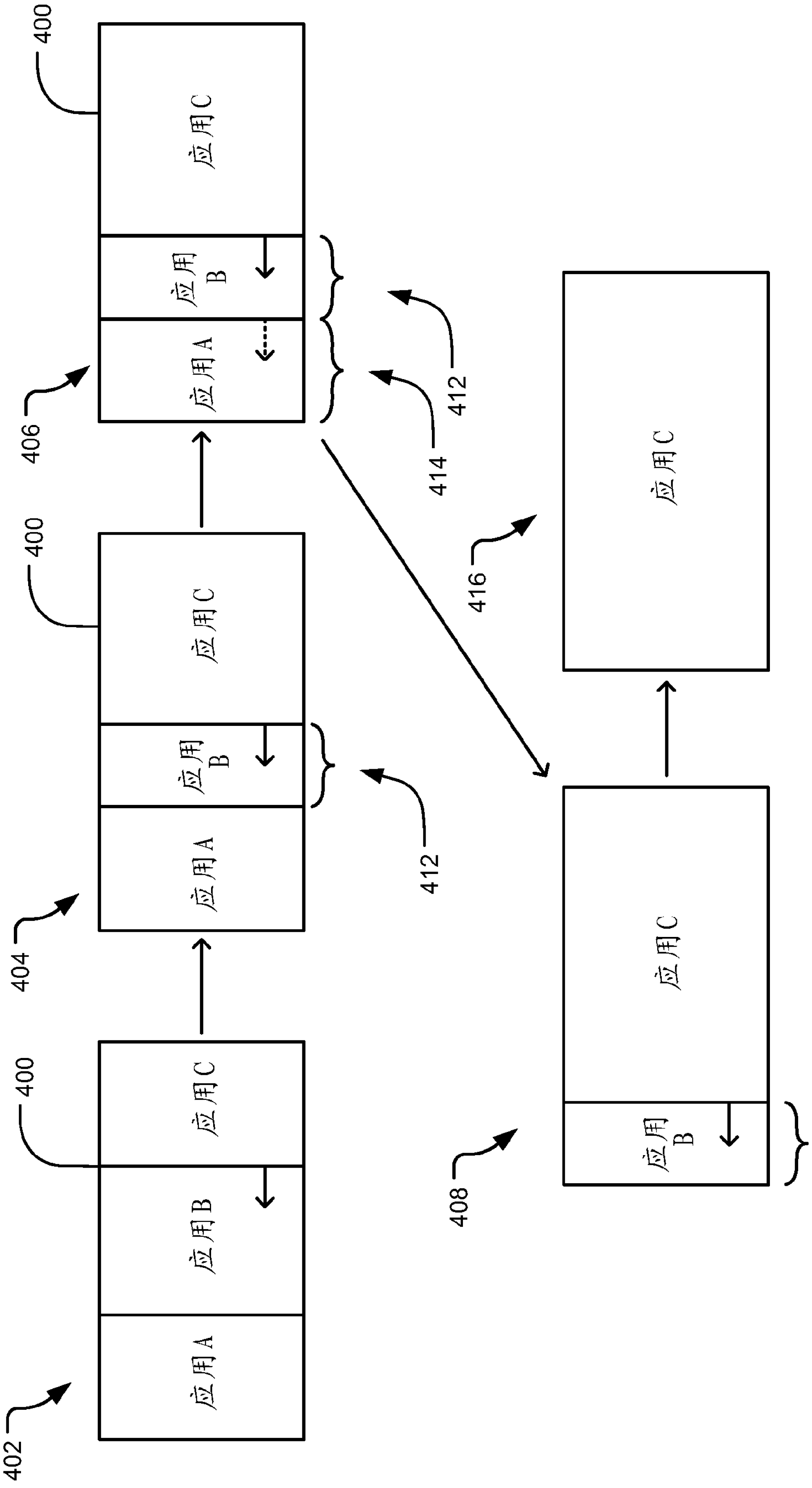

402
应用A
应用B
应用C
400
404
应用A
应用B
应用C
400
412
406
应用A
应用B
应用C
400
414
412
408
应用B
应用C
416
应用C

图 4

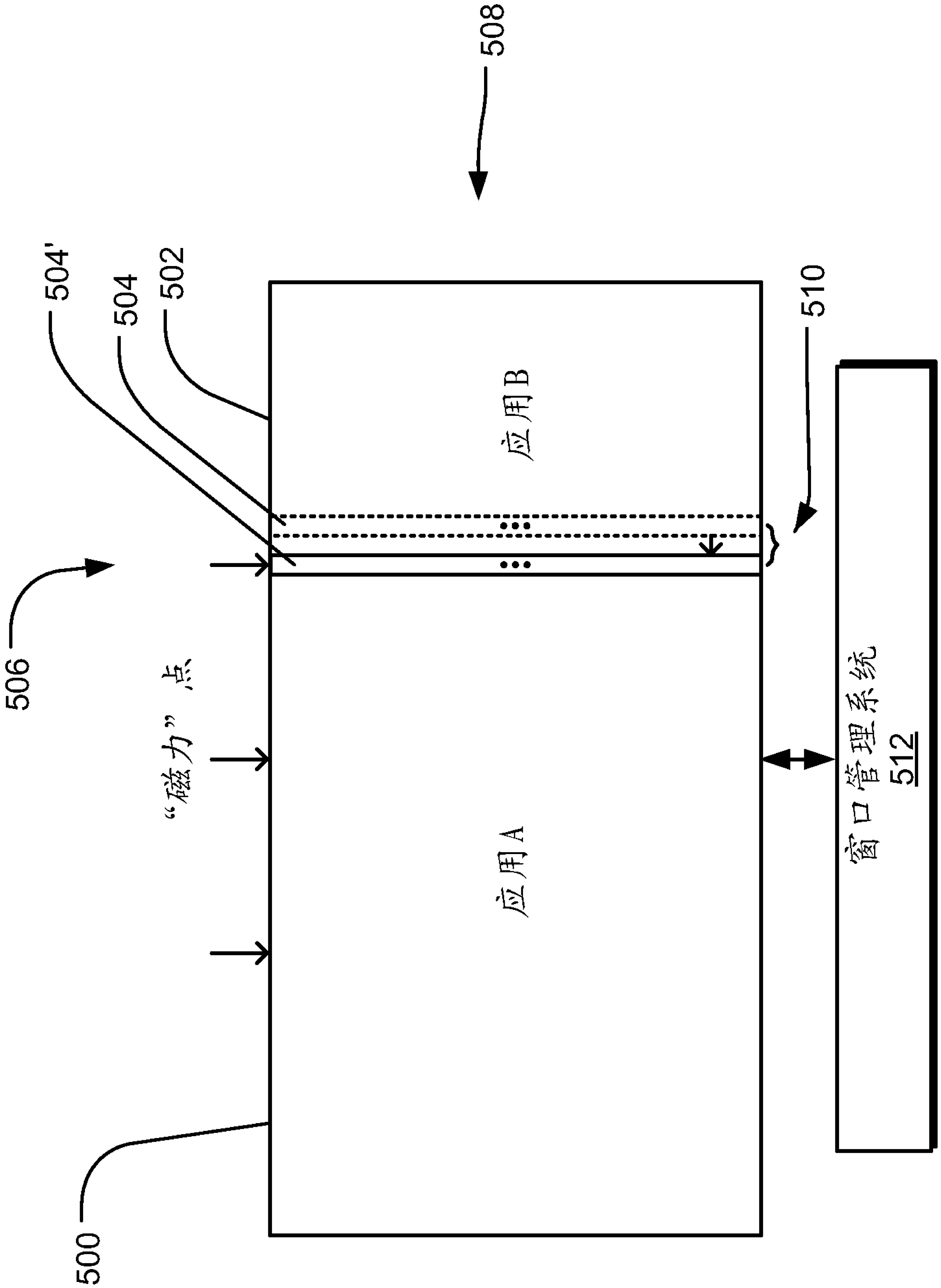


图 5

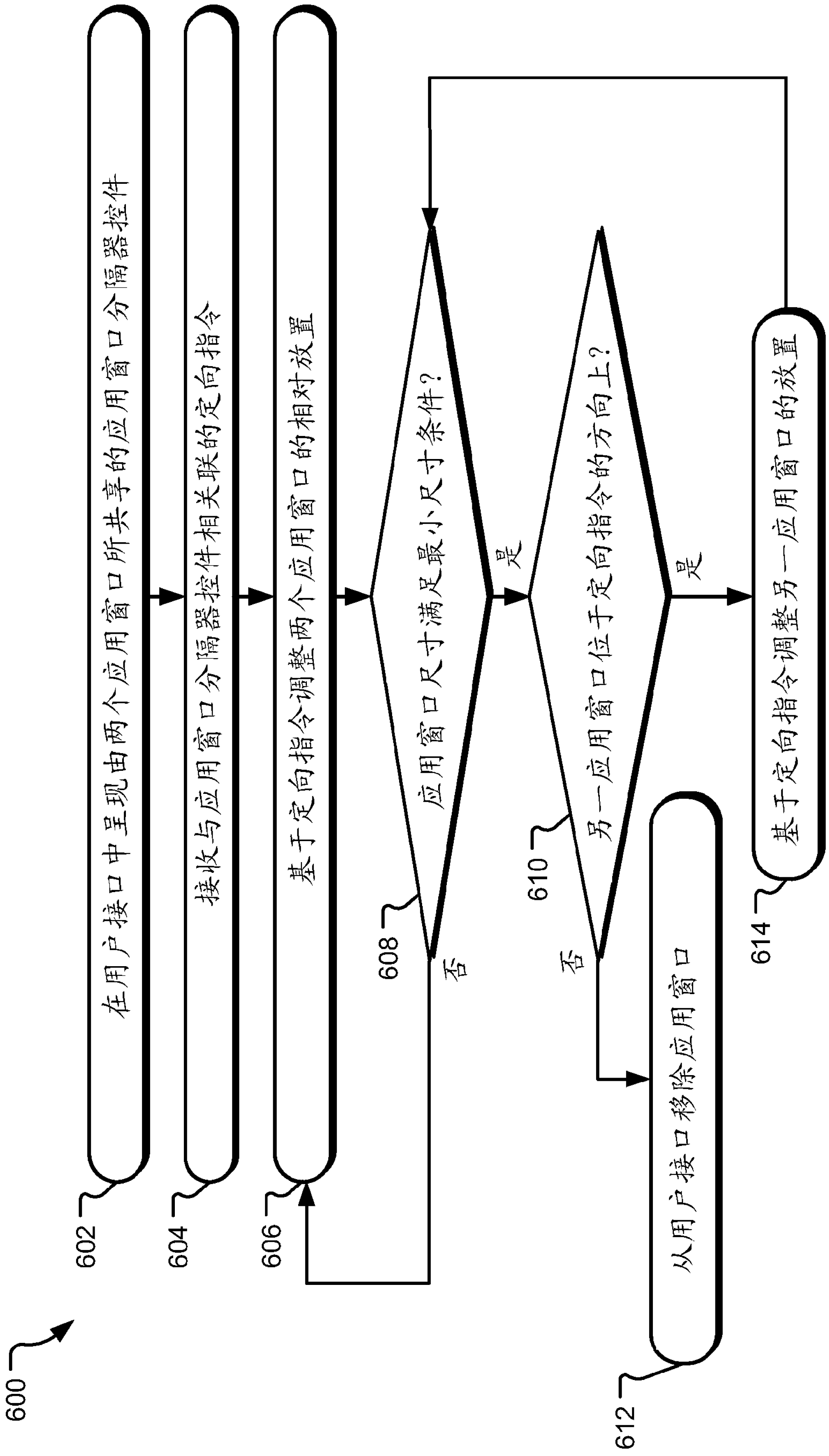

600
602
在用户接口中呈现由两个应用窗口所共享的应用窗口分隔器控件
604
接收与应用窗口分隔器控件相关联的定向指令
606
基于定向指令调整两个应用窗口的相对放置
608
应用窗口尺寸满足最小尺寸条件？
是
否
610
另一应用窗口位于定向指令的方向上？
是
否
612
从用户接口移除应用窗口
614
基于定向指令调整另一应用窗口的放置

图 6

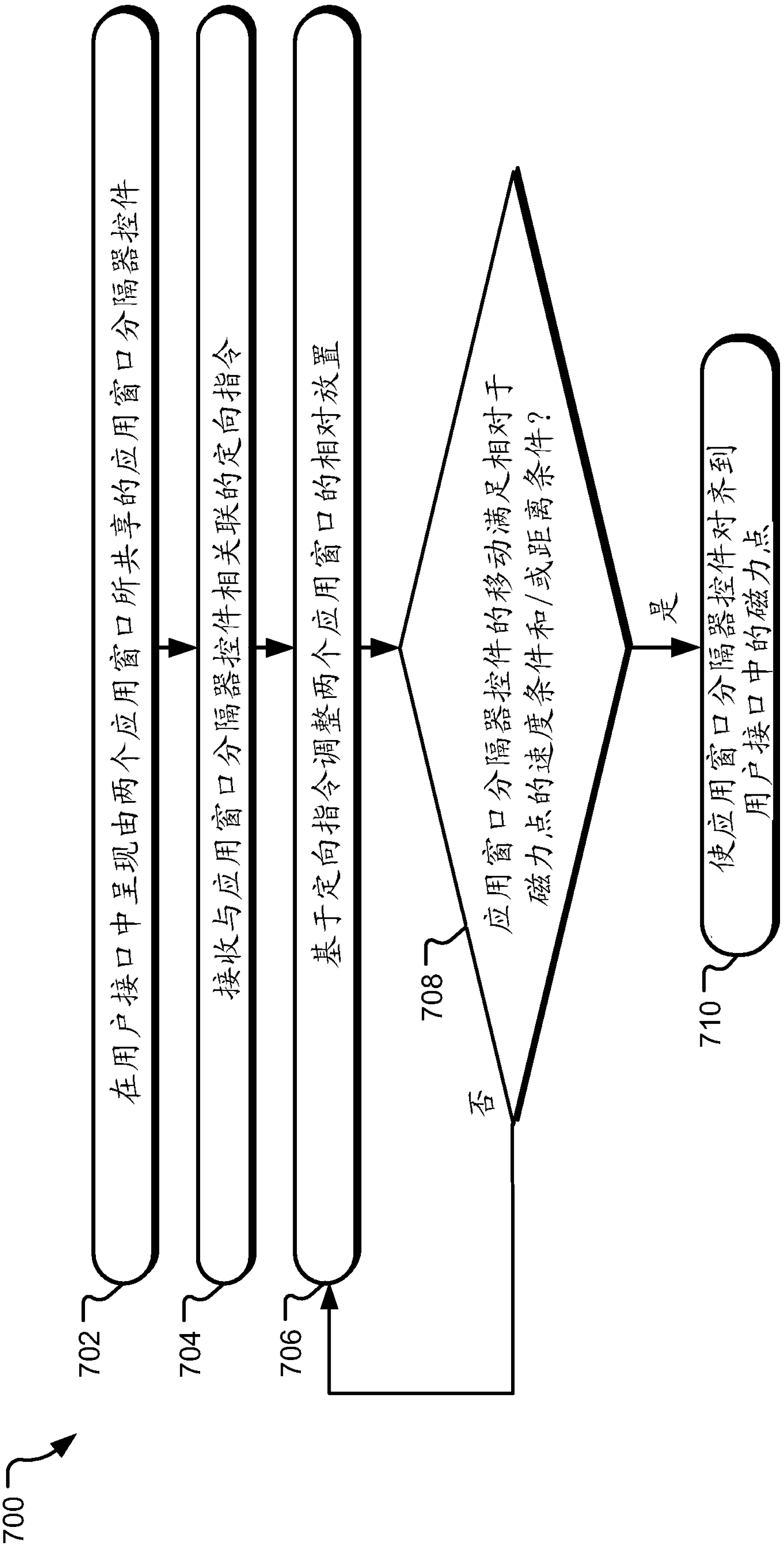

700
702
在用户接口中呈现由两个应用窗口所共享的应用窗口分隔器控件
704
接收与应用窗口分隔器控件相关联的定向指令
706
基于定向指令调整两个应用窗口的相对放置
708
应用窗口分隔器控件的移动满足相对于磁力点的速度条件和/或距离条件?
是
否
710
使应用窗口分隔器控件对齐到用户接口中的磁力点

图 7

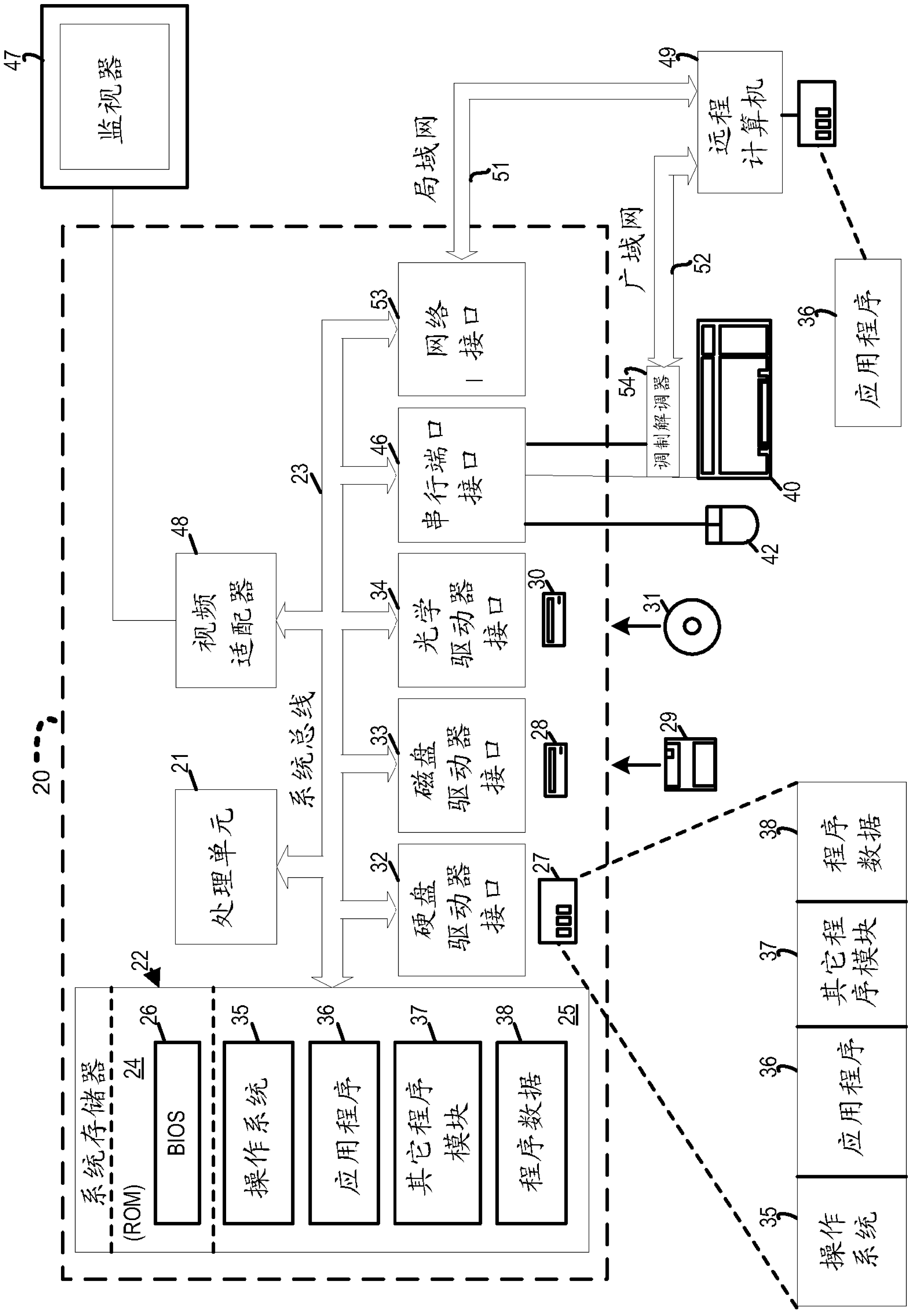

监视器
47
局域网
51
49
远程计算机
广域网
52
36
应用程序
20
53
网络接口
46
串行端口接口
54
调制解调器
40
42
23
48
视频适配器
34
光学驱动器接口
30
31
33
磁盘驱动器接口
28
29
21
处理单元
系统总线
32
硬盘驱动器接口
27
38
程序数据
37
其它程序模块
36
应用程序
35
操作系统
22
系统存储器
24
(ROM)
26
BIOS
35
操作系统
36
应用程序
37
其它程序模块
38
程序数据
25

图 8