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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 November 2014 (20.11.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/185946 A1

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/US2013/060752

(22) International Filing Date:
20 September 2013 (20.09.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/892,532 13 May 2013 (13.05.2013) 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**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WORLEY, Matthew**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SUNDELIN, Nils**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WONG, Tsz Yan**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ANDERSON, Bret**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **JARRETT, Robert**; 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). **STEINGLASS, Alice**; 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,

[Continued on next page]

(54) Title: SMART INSERTION OF APPLICATION WINDOWS INTO A MANAGED DISPLAY REGION

(57) Abstract: Embodiments described herein relate to maintaining a tiled arrangement of application windows as a user inserts application windows into a managed display region. A tiled arrangement is an arrangement where windows do not overlap each other and substantially maximize window occupation of the display region. Generally, when a user inserts an application window the tiled arrangement is maintained by resizing any previously present application windows. Where to insert a window, which displayed windows to resize and to what extent can be determined based on factors such as user input, any prior manual adjustment of the tiled arrangement, and/or minimum window sizes. Whether tile arranging is performed in a managed or curated fashion (e.g., by sharing space equally among windows) or in a semi-curated fashion may be determined by taking into account any of the aforementioned factors.

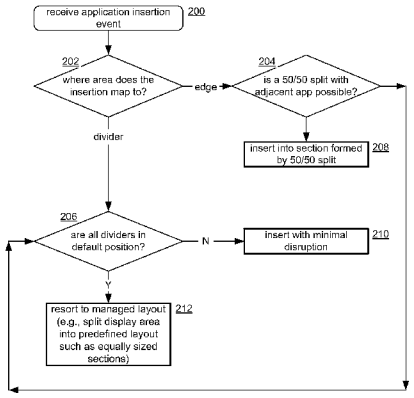

FIG. 4

WO 2014/185946 A1

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))*

— *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))*

SMART INSERTION OF APPLICATION WINDOWS INTO A MANAGED DISPLAY REGION

BACKGROUND

[0001] Computer windowing systems have long been in use. Such windowing systems can have many designs, for instance free-form window manipulation by users. Features of such windowing systems are well known and documented elsewhere. Typically, windowing systems have graphic windows for respective processes running on a computer. A user interacts with the windowing system to manipulate and manage the windows. Most windowing systems have functionality to allow applications (assumed to have respective windows) to be opened, closed, moved, hidden, displayed, resized, customized, and so forth.

[0002] While various schemes for windowing systems have been used, the increasing size of displays (including projectors and 3D displays) as well as the popularity of tablet computing devices and touch input has renewed interest in tiled windowing systems. In such windowing systems application windows may be manipulated by a user to maximize window sizes while keeping windows from occluding each other, even when application windows are added. In general, tiled windowing systems may benefit from improved techniques for automatically managing the insertion by a user of windows to a display where one or more other windows are already displayed. Such techniques and related approaches are discussed below.

SUMMARY

[0003] The following summary is included only to introduce some concepts discussed in the Detailed Description below. This summary is not comprehensive and is not intended to delineate the scope of the claimed subject matter, which is set forth by the claims presented at the end.

[0004] Embodiments described herein relate to maintaining a tiled arrangement of application windows as a user inserts application windows into a managed display region. A tiled arrangement is an arrangement where windows do not overlap each other and substantially maximize window occupation of the display region. Generally, when a user inserts an application window the tiled arrangement is maintained by resizing any previously present application windows. Where to insert a window, which displayed windows to resize and to what extent can be determined based on factors such as user input, any prior manual adjustment of the tiled arrangement, and/or minimum window sizes. Whether tile arranging is performed in a managed or curated fashion (e.g., by

sharing space equally among windows) or in a semi-curated fashion may be determined by taking into account any of the aforementioned factors.

[0005] Many of the attendant features will be explained below with reference to the following detailed description considered in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present description will be better understood from the following detailed description read in light of the accompanying drawings, wherein like reference numerals are used to designate like parts in the accompanying description.

[0007] Figure 1 shows a computing device.

[0008] Figure 2 shows examples of tiled window arrangements.

[0009] Figure 3 shows an example of predefined input regions that allow a user to insert or replace application windows.

[0010] Figure 4 shows an example of an algorithm for automatically maintaining a tiled window arrangement when an application window is inserted by a user.

[0011] Figure 5 shows examples of inserting an application window at an edge when an adjacent split is found to be possible.

[0012] Figure 6 shows an example where an edge insertion includes an application representation inserted at an edge.

[0013] Figure 7 shows an example of an insertion directed to an edge when a split of the adjacent application is not possible.

[0014] Figure 8 shows an example of an insertion at a divider or other interior point.

[0015] Figure 9 shows an example inserting a target application window at a divider or interior point that has been manually adjusted.

[0016] Figure 10 shows an example of a computing device on which embodiments described herein may be implemented.

DETAILED DESCRIPTION

[0017] Embodiments discussed below relate to managing layout and arrangement of windows when a window is interactively added to a display. Tiled windowing systems are discussed first. Interactive window placement is described next, followed by discussion of algorithms for automatically managing interactive window placement or insertion using various signals available at the time of insertion. Example insertion scenarios are also discussed.

[0018] Figure 1 shows a computing device 100 having an operating system 102 managing application processes 104 and displaying corresponding windows 106 on a

display 108. The computing device 100 may be any of a variety of types which are described later with reference to Figure 10. The computing device 100 may be assumed to have at least storage and a processor for executing the operating system 102 and the application processes 104. Although there may be windowing systems with individual windows that represent multiple processes, and also processes without windows, for ease of discussion application processes 104 are assumed herein to each have at least a corresponding window 108. Processes are well known operating system objects that are discretely managed units of execution, typically having a process identifier, memory managed by the operating system (heap and/or stack memory), and may be managed in a process table of process identifiers or the like that is maintained by the operating system.

[0019] The windows 108 may be managed and rendered by a windowing system which may be a part of the operating system 102 or may be closely coupled with operating system 102. The windowing system may itself be a process 104. The windowing system typically responds to user input events and window events, handles the generation and removal of windows 108, resizing windows, moving windows, and many other window-related functions. Techniques described herein may be implemented directly in a windowing system, or in a user interface shell or layer wrapping a windowing system, or a combination thereof. For discussion, implementations in a windowing system are described herein. However, those skilled in the art of computer programming will be able to apply the teachings herein to other implementations and such implementations are contemplated by this description. In short, a "windowing system" is any executing computer component(s) managing windows as described herein.

[0020] As mentioned, the windowing system may implement a tiled management scheme where windows are automatically laid out in a tiled arrangement. A tiled arrangement may generally be one in which windows (or perhaps at least active windows) are automatically kept from overlapping or obscuring each other and yet occupation of the display (or a managed region thereof) by windows is maximized such that non-window space is minimized. Such a tiled arrangement may be automatically maintained while accommodating windows being newly displayed or redisplayed, opened, closed, hidden, moved, inserted, resized, etc. For an example of a tiled windowing system see U.S. Patent application number **13/863,369 (attorney docket 338635.01) filed 4/15/2013. Non-window space may include borders or aesthetic margins, areas for user interface elements such as search tools, shortcuts for functions or particular windows, etc.

[0021] Figure 2 shows examples of tiled window arrangements. Starting from the top, example 120 is a single application window 122 displayed in a managed display region 124. Assuming that another application window 128 is added by a user, for example by opening an application, dragging an indicator of an application into the managed display region 124, invoking an application shortcut, etc., then the windowing system or the like manages the layout of the application windows 122, 128 to maximize window occupancy of the display region 124, as shown in example 126. When there are multiple application windows displayed, the windowing system may automatically provide a user-adjustable divider 130 between the application windows. When the divider 130 is interactively moved by the user two or more affected windows are resized accordingly to maintain the tiled arrangement. If another application window is added the tiled arrangement may continue to be maintained by automatically resizing the existing application windows, as shown in example 132. In one embodiment, application windows are tiled into linear slots (e.g., a single vertical column or a single horizontal row), as seen in the previously mentioned examples (grid-style layouts may also be used as discussed below). In addition, as described further below, the windowing system may be pre-configured to transition to different types of arrangements according to conditions such as the number of application windows, the types of windows, user preferences, minimum window sizes, and others. As seen in other examples 134, 136, automatically managed tiled arrangements may have asymmetries and various rectilinear layouts such as a grid layout.

[0022] Figure 3 shows an example of predefined input regions that allow a user to insert or replace application windows. As mentioned above, a windowing system may allow a user to interactively add an application window to a display (or managed region thereof). The user may control, by various forms of input from an input device, where and how an application window is to be added. In the example of Figure 3, a windowing system dynamically arranges input-sensitive regions in relation to the number and/or location of application windows being displayed. Such regions may include insertion regions 150 and replacement regions 152. When the user directs input for a to-be-displayed (target) application window to a particular insertion region 150 then the target window is inserted accordingly. Similarly, when a target window is directed to a replacement region 152 a corresponding application window may be replaced (note that any of the insertion techniques described herein may be applied to or used in conjunction with window replacement). If the user directs the target application window to the leftmost insertion region 150 then the target application window will be inserted accordingly; the previously

displayed application windows (e.g., "app1" and "app2") may be moved and/or resized to fit both the previously displayed application windows and the target application window.

[0023] The regions may be defined in a number of ways. The regions may have predefined default sizes and locations for different numbers of displayed application windows. The regions may be dynamically configured according to factors such as the size of the display, accessibility settings, font size, user preferences derived from past manipulation of dividers, ratios of window size and/or display size, user input such as modifier keys (e.g., left-click vs. right-click), and others. Also, there may be inert regions where user input will neither insert nor replace the target window. User-activatable regions with other functions may also be included.

[0024] Any type of input device may be manipulated by a user to direct placement of a target application window (as described elsewhere, alternatively, a user may initiate a window insertion but the insertion location or position can be automatically determined). The input device may be, for example, a touch sensitive surface, a mouse, a digitizer pad, an object-recognizing camera system, and so on. In addition, the input thereof may take a variety of forms. For example, a user may hover an input point over an input-sensitive region, an input-sensitive region may be explicitly designated, a marker or icon may be dragged and dropped in an input-sensitive region, etc. In one embodiment, a representation of the target application is moved on the display by the user and a nearest region may be selected with explicit user input or by the user ending or releasing the movement.

[0025] Figure 4 shows an example of an algorithm for automatically maintaining a tiled window arrangement when an application window is inserted by a user. It will be assumed for discussion that the user is providing a stream of two (or more) dimensional input points as by a touch stroke on a surface or a mouse movement, for example. The input points may move, on the display, an input pointer, an application representation (icon, thumbnail, etc.), the target application window itself, or some other graphic object. Graphics or highlights may be displayed to indicate an input-sensitive region (or a destination screen area) that the user is about to select or target. Details for previewing a user's intended insertion are discussed further below. According to the user's input (e.g., when the input points stop being inputted or when the input is modified or another input is provided) a screen location for the target window is designated and step 200 occurs; an application insertion event is received.

[0026] Following step 200, at step 202 there is a determination of where the insertion maps on the display or a managed region thereof. If the insertion maps to an edge or an exterior element of the display or managed display region (to be referred to inclusively as "display" for short) then step 204 is performed, otherwise step 206 is performed. Note that an edge can correspond to an exterior input-sensitive insertion region such as the leftmost and rightmost insertion regions 150. If the user's insertion maps to an edge then there is an application window adjacent to the selected edge, and that application window is used in performing step 204. Step 204 determines whether the space occupied by the adjacent application window is large enough to fit the target application window and the adjacent application window with a reduced size. In one embodiment, this space might be required to be large enough to accommodate the new application and divider without disturbing the existing dividers on screen. This may be determined by various means such as by determining if the space occupied by the adjacent application windows is larger than a minimum size of the adjacent application window plus a minimum size of the target application window (plus any addition needed space such as for a new divider or margins). Such minimum sizes can be application-specific settings, default system settings, dynamically computed according to features such the size of the display, and so forth. As used herein, "minimum size" also refers to a size preference, possibly an application attribute, which might or might not be strictly enforced.

[0027] When step 204 determines that a split is to be performed, then at step 208 the space occupied by the adjacent application window is split, for instance divided in half, the adjacent application window is resized to fit its half, and the target application window is displayed with a size to fit its half (the half next to the selected edge). Ratios other than 50/50 may be used, for instance to accommodate a minimum size of either application window or application preference, but the split may be limited to splitting the space of only the adjacent application window; any other application windows that are already displayed are not affected.

[0028] If it is determined at step 202 that an edge is not selected or a divider or other non-edge or internal element has been designated by the user as an insertion area, then step 206 is performed (in some cases, when step 204 determines that there will be no split, the process may go to step 206). In this case, another test is performed to determine if all dividers are in their default positions. In a general sense, the algorithm checks to see whether the window applications are in a curated or managed mode where window tiling is managed by the system and has not been overridden by the user. If the dividers (or

application window screen partitions) are in their default positions then automated or curated management of the tiled layout continues at step 212. The automated or curated layout management may involve using pre-defined layouts for the number of current application windows, evenly dividing the display into sections for the respective application windows, or other fully managed tiled layouts.

[0029] If at step 206 it is determined that the user has overridden the default layout, for instance by checking to see if a divider has been moved, then at step 208 the target application window is inserted with techniques to minimize disruption of the current display layout. In other words, the algorithm attempts to minimize disruption of the layout preference indicated by the user. In one embodiment, the target application window is inserted at a minimal size with minimal automatic rearrangement of dividers to allow fitting the target application. This may also require moving one or more other dividers the least amount possible. Specifically, an implementation may start by targeting the applications that are closest and try to obtain as much space as possible. If an application-minimum is a limiting factor, then the implementation may radiate out to take more space until it has sufficient space for an insertion. If a set of applications in a managed or curated layout (e.g., 50/50, 75/25, 40/30/30) are encountered then space may be taken equally from those applications (rather than one at a time), thus preserving the pre-existing applications' relative size relationships.

[0030] The process described above uses various types of signals or indicators to determine how to automatically arrange a tiled layout. The steps and decisions mentioned with reference may be varied in order and in other ways while achieving similar ends. Variations of the process are able to fully manage the insertion of application windows without requiring the user to do anything other than to designate a location to insert the application window. However, if the user has manually configured the current tiled arrangement of application windows then the process may insert windows in a way that avoids or minimizes any change to the layout of relevant application windows. In addition, features of the application windows themselves may be taken into account. For example, minimum size, range of supported sizes, expected or average size, past manual sizings of an application window, or other size-related features may be used.

[0031] Note that while interactive placement has been described above, the same space-allocation techniques may be used regardless of how an insertion location has been selected. For example, if an application is automatically launched by a user action, a

location for inserting that application can be determined automatically and then that location can be used for insertion.

[0032] Figure 5 shows examples of inserting an application window at an edge when an adjacent split is found to be possible. At the top of Figure 5 the user has designated left edge 240 as an insertion point and the target application window "app2" has been inserted accordingly with tiled layout computed automatically. The next sequence of Figure 5 shows selection of right edge 242 and "app2" inserted at the corresponding side of the display 106. The lower sequence of Figure 5 shows insertion at the left edge 240 when two application windows are present. There is a determination that the space occupied by adjacent "app1" is able to contain both the target application window "app3" and the "app1" window and the space is split 50/50. The "app3" window is inserted next to the selected edge and the adjacent application window of "app1" is resized resulting in the tiled arrangement at the lower right of Figure 5. In an embodiment where dividers are used, a divider is added between the two application windows of "app3" and "app1".

[0033] Figure 6 shows an example where an edge insertion includes an application representation 260 inserted at an edge. The insertion event is received, there is a determination that a split of the adjacent application "app2" is possible, and the application window for "appN" is inserted into the space created by the split. A divider may be added between the application windows for "app2" and "appN". The application representation 260 may move without restraint, may move on a horizontal "rail", or may snap from one insertion or replacement region to another, for example.

[0034] Figure 7 shows an example of an insertion directed to an edge when a split of the adjacent application is not possible. The insertion event for "appN" is received and there is a determination according to application size information 270 and the size of the adjacent space (600 pixels) a split is not to be performed. While approximately 600 pixels are available and the "app2" window's minimum size of 300 pixels would fit in half of the space. However, the 400 pixel minimum width of the "appN" window would not fit in a 50/50 split. Consequently, step 206 is performed to determine how to insert the application window for "appN".

[0035] Figure 8 shows an example of an insertion at a divider or other interior point 280. In this case, when the target application is inserted by the user at a divider or an insertion region that is not an edge, if the divider or default layout has not been altered by the user then the layout manager determines how to arrange the application windows. In this curated or fully-managed mode the layout is arranged by equally distributing display space

between the newly displayed target application window and the previously displayed application windows. In practice the distribution may not be precisely equal due to rounding, borders, margins, dividers, etc.

[0036] Figure 9 shows an example inserting a target application window at a divider or interior point 280 that has been manually adjusted. In this case, when it is determined that the insertion point maps to an interior area or a divider and it is also determined that the divider or default managed layout has not been altered, then a tiled arrangement is computed to minimize impact on the user preference implicit in the detected manual adjustment. For example, inserting the target application window at its minimum size, reducing the pre-existing applications to preserve the ratio between the pre-existing application windows (e.g., "app1" and "app2"), or other approaches may be used. Note that minimal sizes or other features of the windows may be used to inform the impact-minimizing redistribution of space on the display 106.

[0037] Target application windows or the need to insert same may originate from a variety of sources. For example, a target window may be a newly launched application. A target window may be a result of interaction with a user interface element for tracking active or launchable applications, for instance selecting an application icon or dragging an application thumbnail into the managed area. A keypress or other input for launching or switching-to an application can also result in a target window to be interactively inserted. It may be appreciated that target application windows being interactively inserted are application windows that are not currently displayed on the display or the managed region thereof.

[0038] Some embodiments may repeatedly compute and preview a new tiled arrangement that would occur if a target were placed according to a current input location. In other words, as a user provides locational input such as touch or mouse input points the windowing system may use recent input points as potential insertion points and compute how the displayed and target windows would be tiled if the recent input points were used as the insertion points. Thus the user may be able to customize the insertion and tiling process. Such previews may be simple graphics such as highlights of regions to be affected by a potential insertion, highlights to indicate expected sizes of windows. It is also possible that such input points can be displayed as actual insertions (changing according to changing input points) that are then finalized or undone according to additional user input or ending the current user input. It may be appreciated that the tiled application windows displayed on the display or the managed display region are live or

executing application windows susceptible to user input and each able to dynamically display content as the case may be. While horizontal tiling schemes have been described vertical schemes or vertical and horizontal schemes (and corresponding insertion regions) are equally applicable.

[0039] Figure 10 shows an example of a computing device 300 on which embodiments described above may be implemented. The computing device 300 may have the display 106, as well as storage 302 and a processor 304. These elements may cooperate in ways well understood in the art of computing. In addition, input devices 306 may be integrated with or in communication with the computing device 300. The display 106 may be a touch-sensitive display that also functions as an input device. The computing device 300 may have any form factor or be used in any type of encompassing device. For example, touch-sensitive control panels are often used to control appliances, robots, and other machines. The computing device 300 may be in the form of a handheld device such as a smartphone, a tablet computer, a gaming device, a server, or others.

[0040] Embodiments and features discussed above can be realized in the form of information stored in volatile or non-volatile computer or device readable devices. This is deemed to include at least devices such as optical storage (e.g., compact-disk read-only memory (CD-ROM)), magnetic media, flash read-only memory (ROM), or devices for storing digital information. The stored information can be in the form of machine executable instructions (e.g., compiled executable binary code), source code, bytecode, or any other information that can be used to enable or configure computing devices to perform the various embodiments discussed above. This is also deemed to include at least volatile memory such as random-access memory (RAM) and/or virtual memory storing information such as central processing unit (CPU) instructions during execution of a program carrying out an embodiment, as well as non-volatile devices storing information that allows a program or executable to be loaded and executed. The embodiments and features can be performed on any type of computing device, including portable devices, workstations, servers, mobile wireless devices, and so on.

CLAIMS

1. A method of managing display of windows of respective mutually autonomous applications executing on a computer comprising a display, a processor, an input device, and storage, each application comprising a respective process managed by an operating system of the computer, the method comprising:

displaying a plurality of windows on the display, each window separately managed by a windowing system that manages windows for the applications;

receiving interactive user inputs specifying respective locations for insertions of the windows to user-designated locations within a managed region of the display; and

each time one of the interactive inputs is received to place an insertion of a corresponding target window when only one or more of the windows, excluding the target window, are currently displayed in a tiled arrangement in the managed region, responding to the interactive input to insert the target window into the managed region by:

automatically resizing either or both of the one or more windows that are currently displayed in the region,

and newly displaying the target window in the managed region such that the target window and the one or more windows currently displayed are displayed in a tiled arrangement in the managed region.

2. A method according to claim 1, further comprising displaying a graphic indicator that is displayed as moving in correspondence with user interaction with an input device, locations of the graphic indicator corresponding to the input-specified locations.

3. A method according to claim 1, further comprising accessing size information comprising minimum or preferred sizes of the applications and using the minimum or preferred sizes to perform the resizing.

4. A method according to claim 1, further comprising providing a user-adjustable divider between two currently displayed windows and, when performing the resizing responsive to an interactive input placing a corresponding target window, resizing either or both of the two currently displayed windows according to whether the user-adjustable divider has been adjusted by the user.

5. A method according to claim 1, further comprising computing a tiled arrangement using minimum window sizes, or indications of user adjustment of sizes of one or more displayed windows, or the user-specified insertion locations of the target windows, or by taking into account a user override input that controls the tiled arrangement.

6. A method according to claim 5, wherein when a target window is being inserted the method first determines whether a corresponding insertion location is at an edge of the managed region and when so determined the method determines whether a space occupied by an edge window that is adjacent to the edge can accommodate the edge window and the target window without changing the size of any other displayed windows, and when so determined splitting the space into two parts to which the target window and edge window are respectively fitted, the target window being displayed in the part that is next to the edge.

7. A method according to claim 6, further comprising: when determined that the space cannot accommodate resizing, according to whether a manual adjustment to a window size has been made and/or according to minimum window sizes, resizing at least two currently displayed windows to accommodate insertion of the target window next to the edge.

8. A computing device comprising storage, a processor, a display, and an input device, the computing device comprising:

the storage storing instructions executable by the processor;

the input device providing, according to user interaction therewith, user inputs that actuate window insertions on the display, each window insertion having a respective insertion location relative to the display;

the processor, according to the instructions, maintaining a tiled arrangement of all application windows displayed by the display; and

the processor, according to the instructions, responding to a window insertion for inserting a target window into the tiled arrangement by determining, according to (i) whether the tiled arrangement has been manually adjusted by the user and/or (ii) whether the insertion location corresponding to the target window corresponds to an edge of the display, how to allocate display space to the target window, wherein after the target window is inserted the windows displayed by the display are in a tiled arrangement.

9. A computing device according to claim 8, the processor further determining how to allocate display space according to application preferences of the windows.

10. A computing device according to claim 8, wherein the display space is allocated substantially among the windows according to a predefined ratio when the target windows is not inserted at an edge and the display space is split from an edge window when the target window is inserted at an edge.

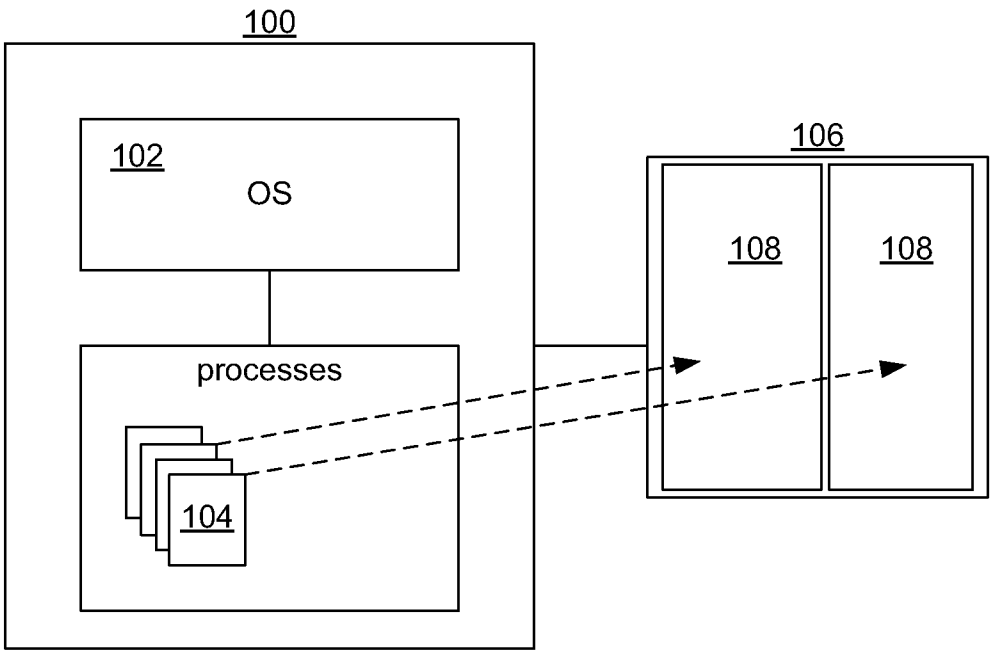

FIG. 1

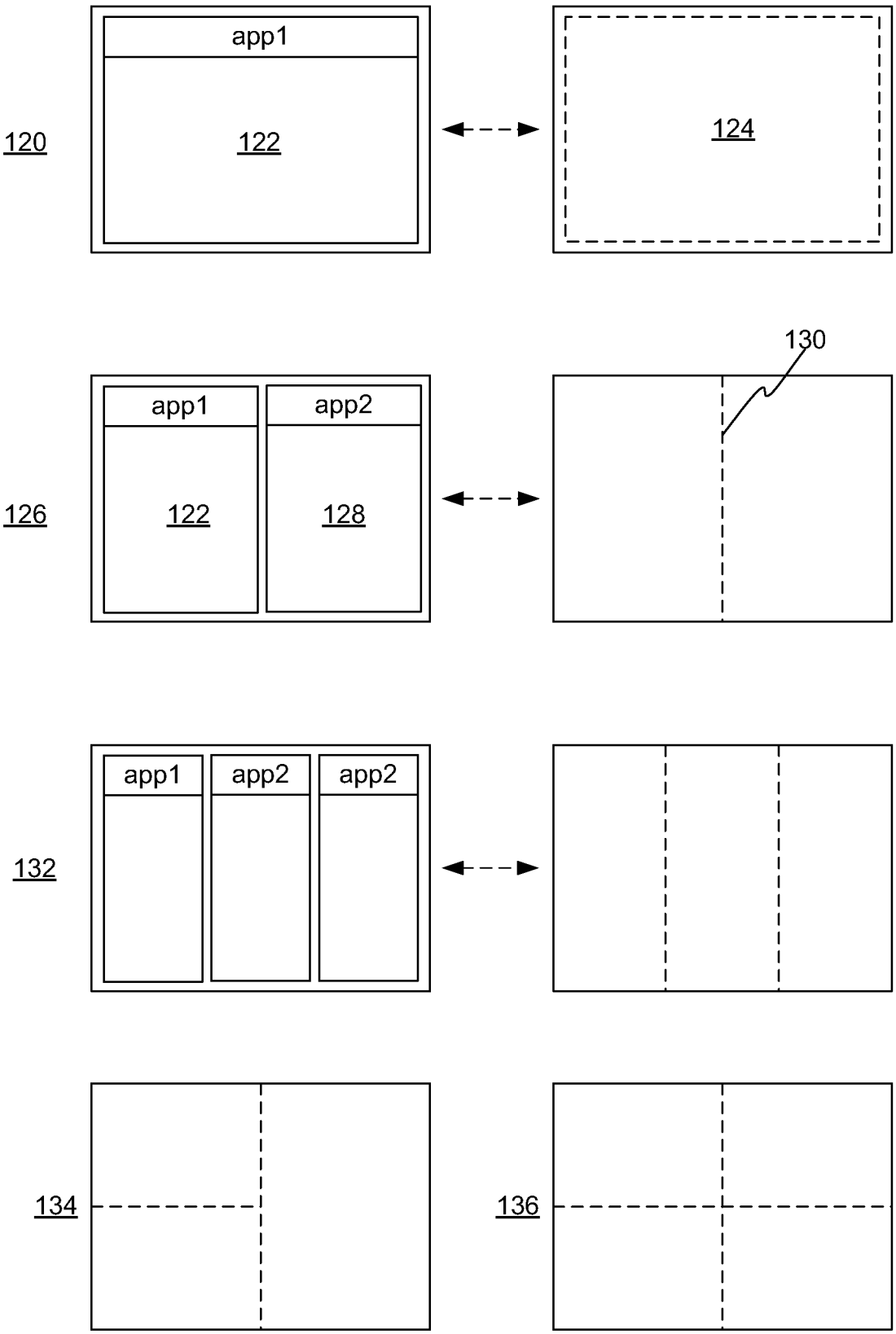
app1
120
122
124
130
app1
app2
126
122
128
app1
app2
app2
132
134
136

FIG. 2

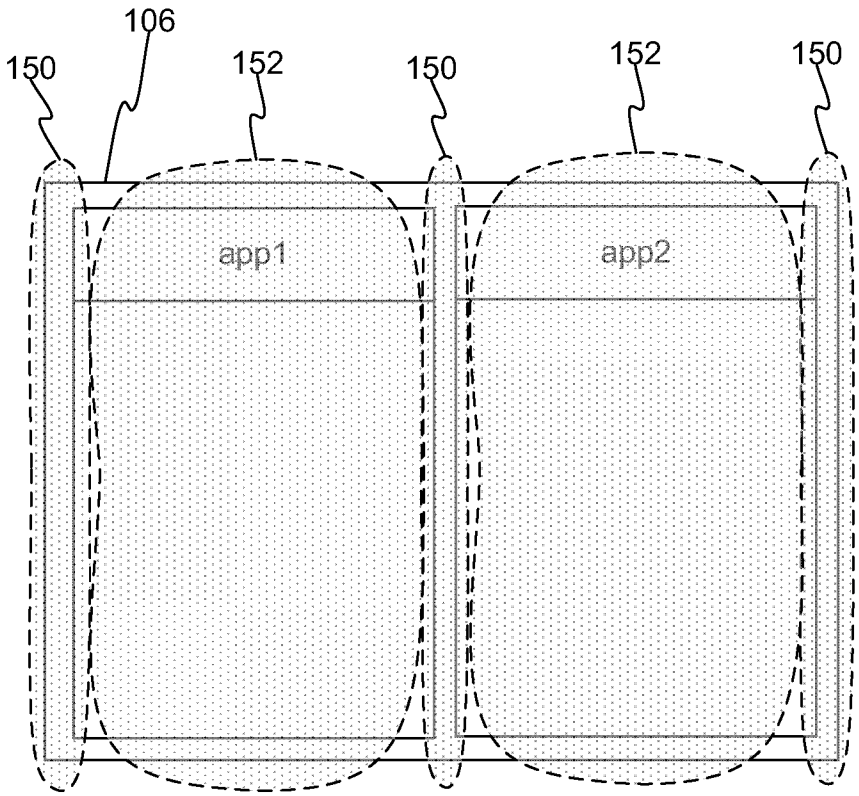

FIG. 3

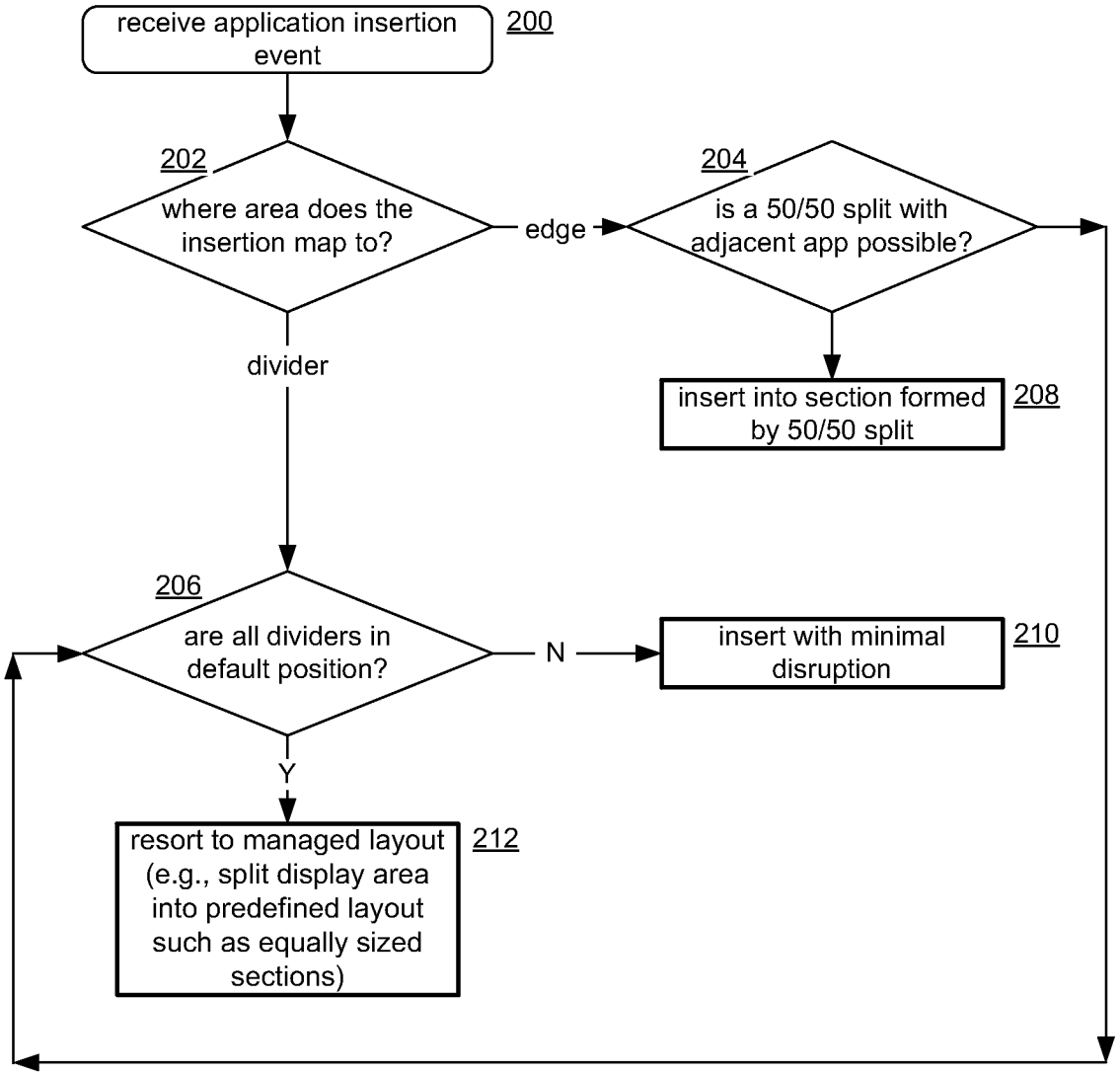
receive application insertion event
200
202
where area does the insertion map to?
edge
204
is a 50/50 split with adjacent app possible?
divider
insert into section formed by 50/50 split
208
206
are all dividers in default position?
N
insert with minimal disruption
210
Y
resort to managed layout (e.g., split display area into predefined layout such as equally sized sections)
212

FIG. 4

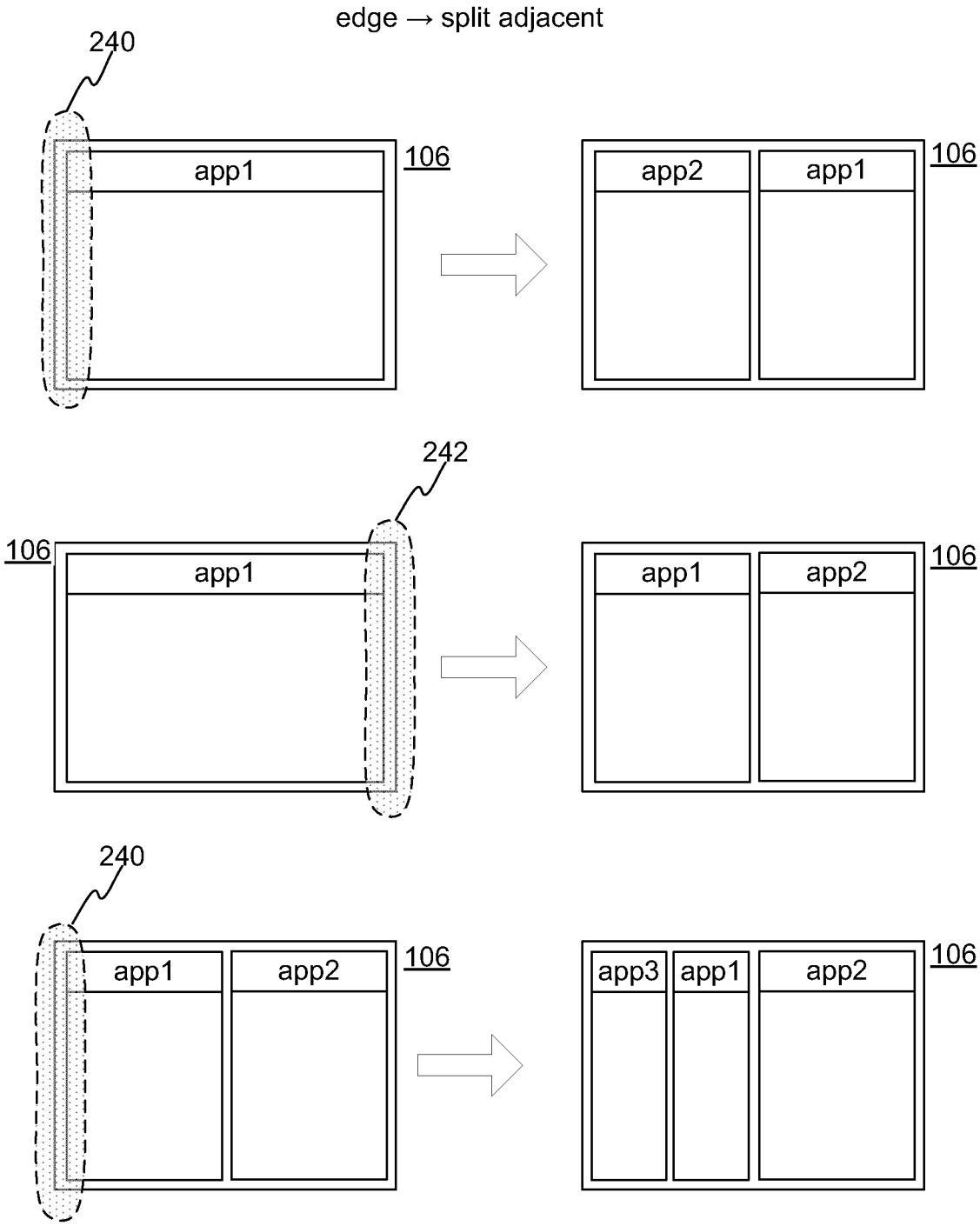
edge → split adjacent
240
app1
106
app2
app1
106
242
106
app1
app1
app2
106
240
app1
app2
106
app3
app1
app2
106

FIG. 5

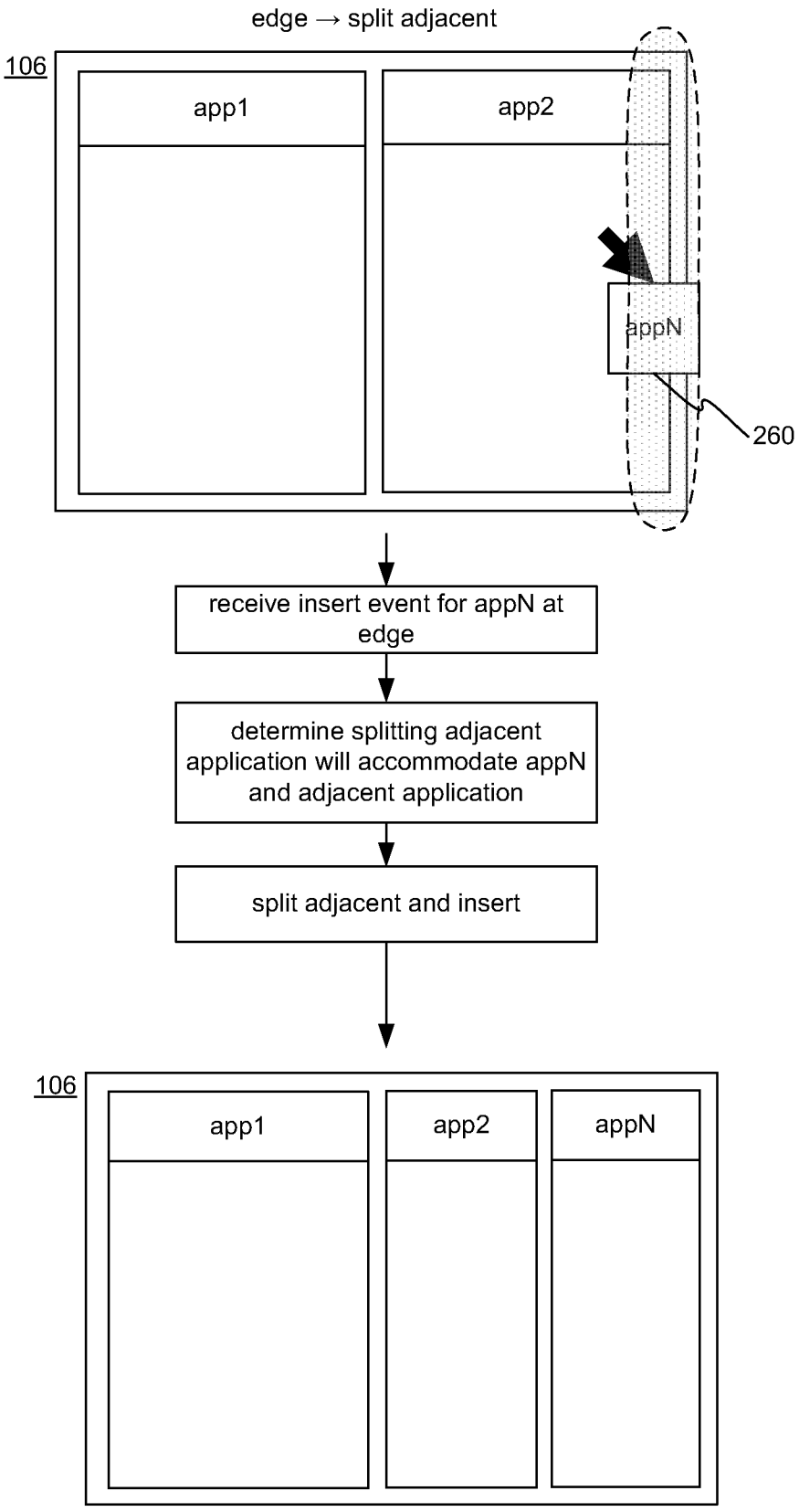
edge → split adjacent
106
app1
app2
appN
260
receive insert event for appN at edge
determine splitting adjacent application will accommodate appN and adjacent application
split adjacent and insert
106
app1
app2
appN

FIG. 6

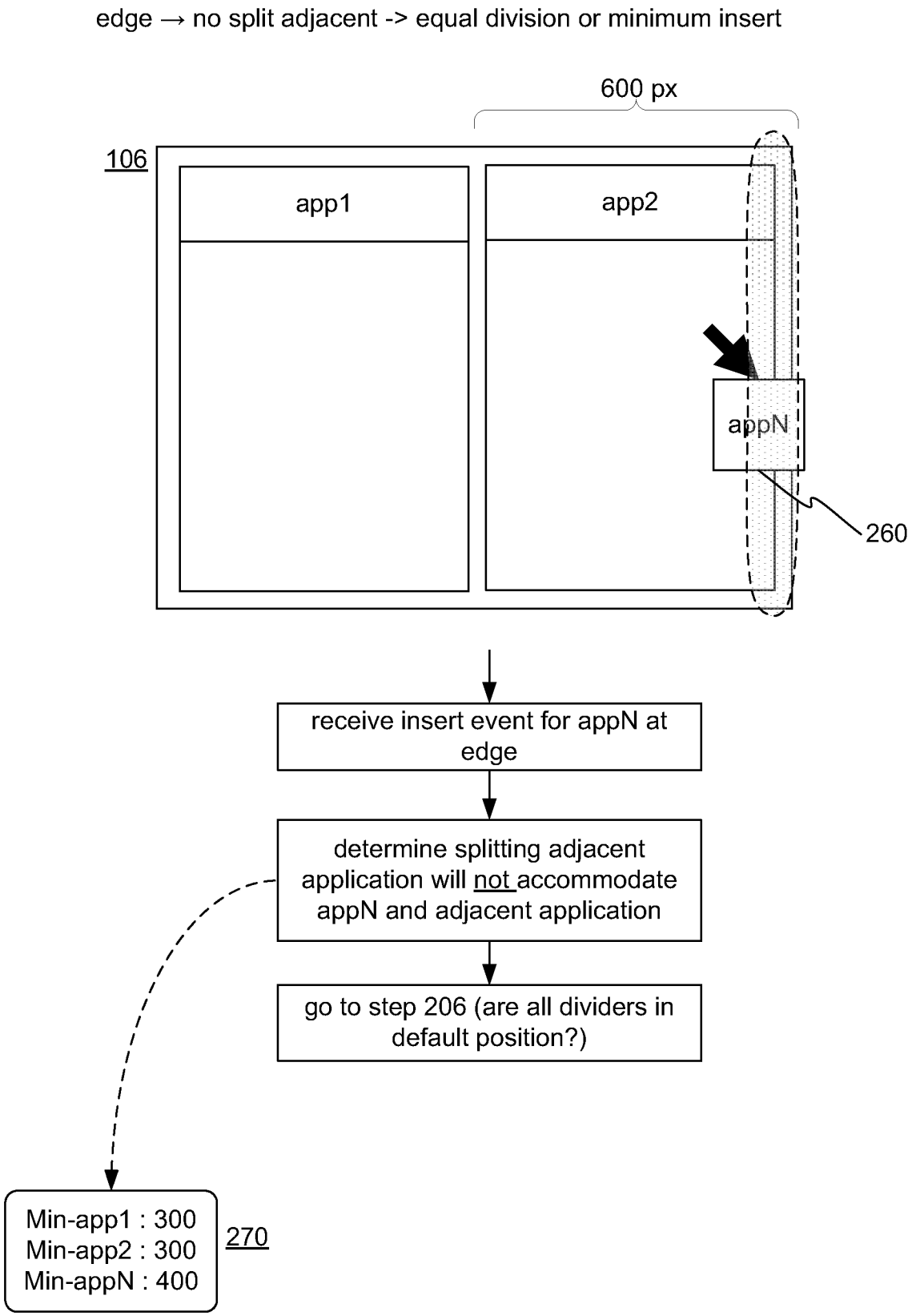
edge → no split adjacent -> equal division or minimum insert
600 px
106
app1
app2
appN
260
receive insert event for appN at edge
determine splitting adjacent application will not accommodate appN and adjacent application
go to step 206 (are all dividers in default position?)
Min-app1 : 300
Min-app2 : 300
Min-appN : 400
270

FIG. 7

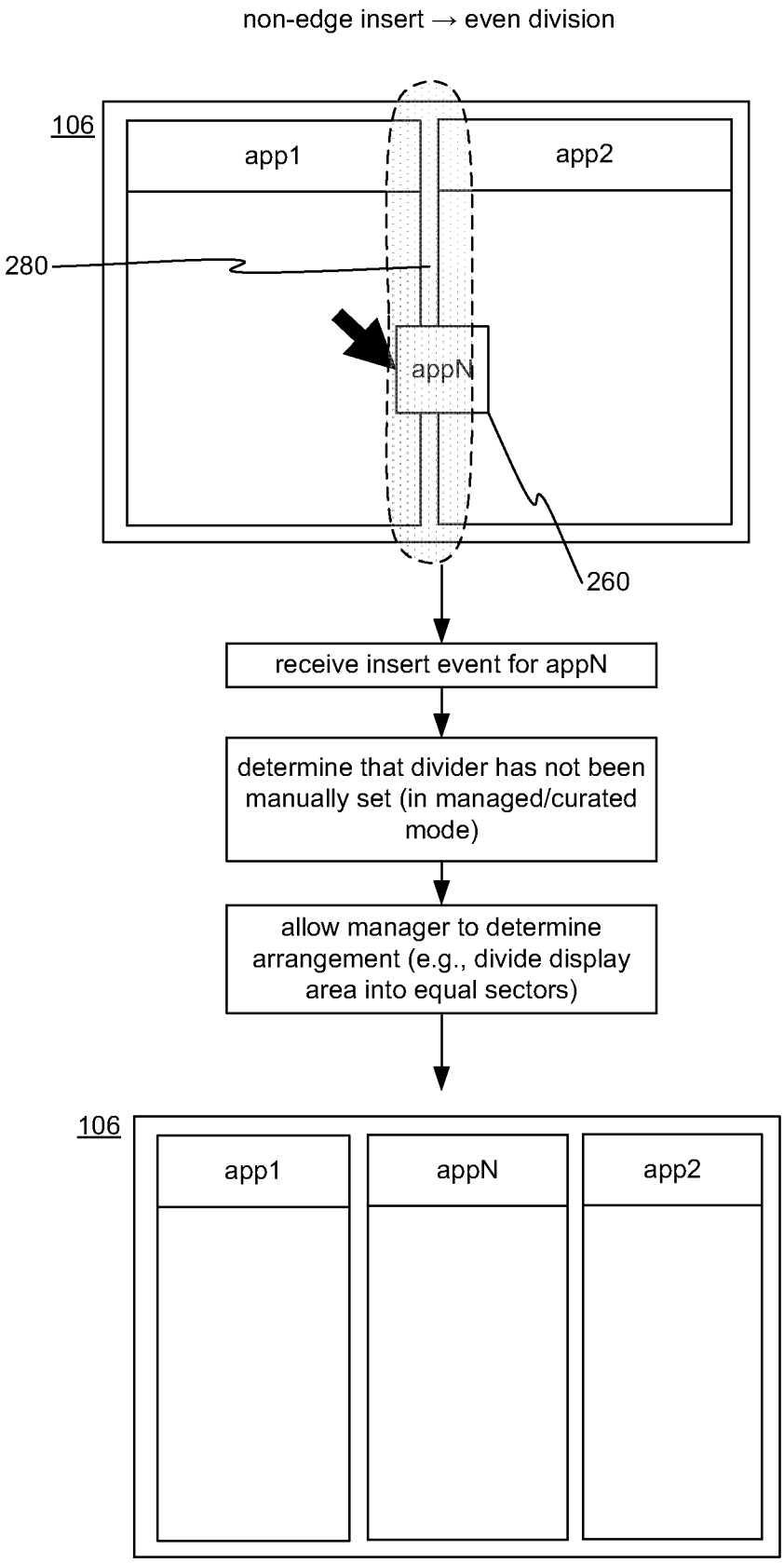
non-edge insert → even division
106
app1
app2
280
appN
260
receive insert event for appN
determine that divider has not been manually set (in managed/curated mode)
allow manager to determine arrangement (e.g., divide display area into equal sectors)
106
app1
appN
app2

FIG. 8

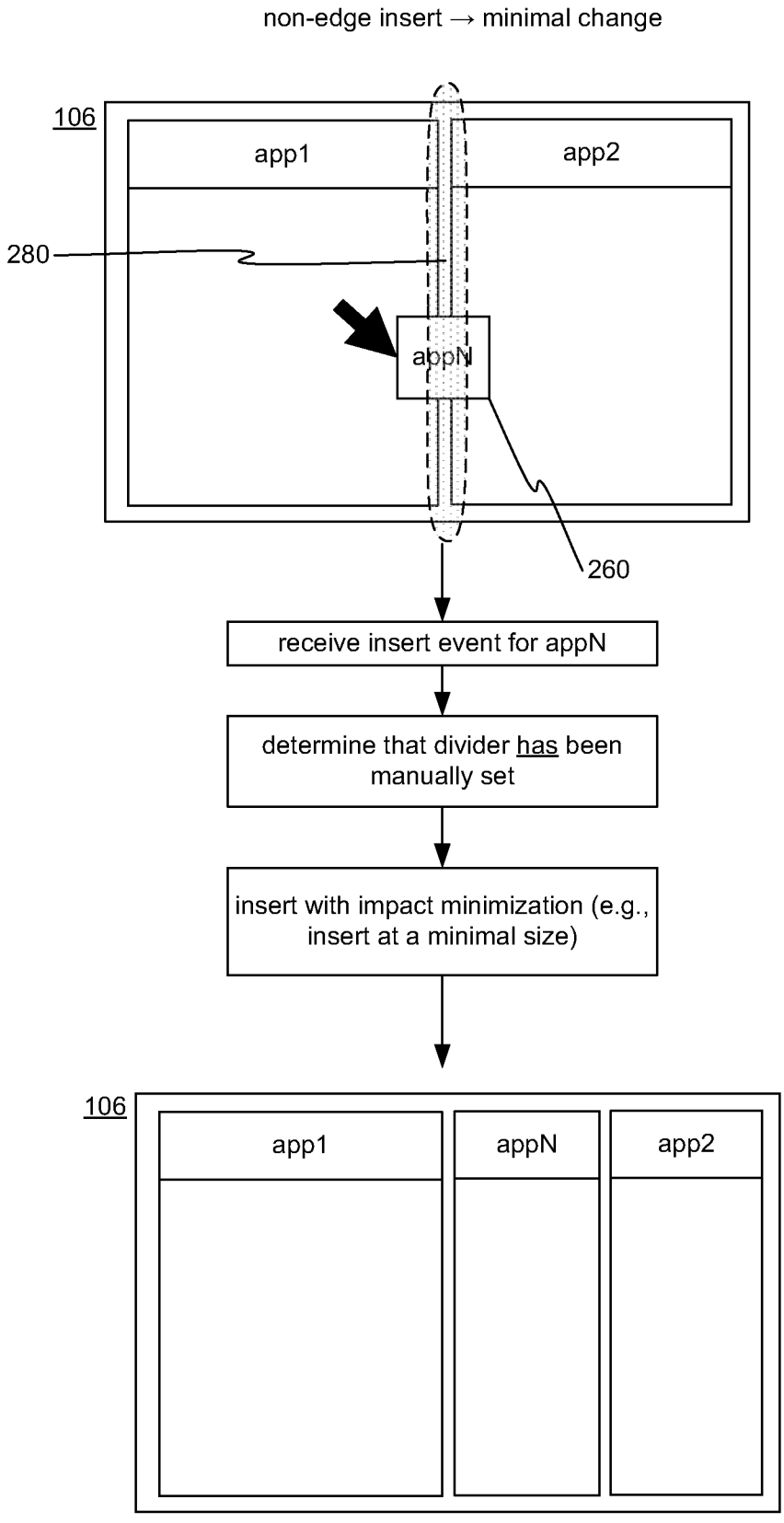
non-edge insert → minimal change
106
app1
app2
280
appN
260
receive insert event for appN
determine that divider has been manually set
insert with impact minimization (e.g., insert at a minimal size)
106
app1
appN
app2

FIG. 9

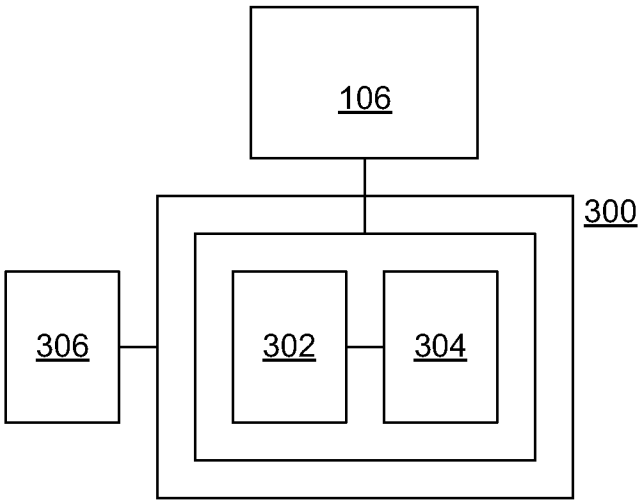

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/060752

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44
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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 008 809 A (BROOKS ARTHUR P [US]) 28 December 1999 (1999-12-28) column 5, line 31 - column 7, line 31 -----	1-10
X	EP 0 820 002 A1 (IBM [US]) 21 January 1998 (1998-01-21) paragraphs [0031] - [0033], [0042]; figures 2-4 -----	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 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	Date of mailing of the international search report
17 January 2014	29/01/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Bijn, Koen

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/060752

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6008809 A	28-12-1999	JP 3504502 B2	08-03-2004
		JP H11110099 A	23-04-1999
		US 6008809 A	28-12-1999
EP 0820002 A1	21-01-1998	DE 69716186 D1	14-11-2002
		DE 69716186 T2	12-06-2003
		EP 0820002 A1	21-01-1998
		US 5920315 A	06-07-1999