

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
3 June 2010 (03.06.2010)(10) International Publication Number
WO 2010/063016 A9(51) International Patent Classification:
G06T 17/00 (2006.01)234, st. th., DK-2200 Copenhagen N (DK).
GOTTDENKER, Joshua, Sills [US/DK]; **sixtusvej 17, c/o Mindegaard, DK-2300 Copenhagen S (DK). KRISTENSEN, Tor** [US/DK]; **Svanevej 18, 2 tv, DK-2400 Copenhagen NV (DK). HALL, Clifford, Lee** [US/US]; **3 Maple Ridge Court, Jonesborough, Tennessee 37659 (US).**(21) International Application Number:
PCT/US2009/066028(22) International Filing Date:
27 November 2009 (27.11.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/118,121 26 November 2008 (26.11.2008) US(71) Applicant (for all designated States except US): **LILA APS (AHEAD.)** [DK/DK]; Schiller House, Nannasgade 28, DK-2200 Copenhagen N (DK).

(72) Inventors; and

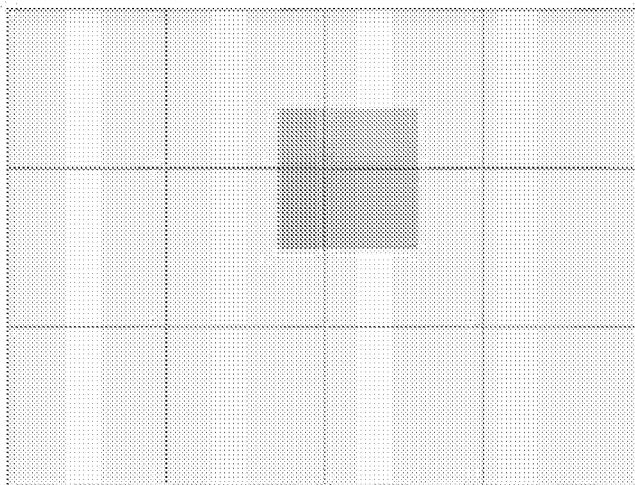
(75) Inventors/Applicants (for US only): **GROTH, Rasmus** [DK/DK]; Hjortholms Alle 36, 1, DK-2400 Copenhagen NV (DK). **TOENNESEN, Asbjorn, Sloth** [DK/DK]; Krabbesholmvej 27, 2. th, DK-2700 Broenshoej (DK). **FOLCARELLI, Manolo, Oliver** [DK/DK]; Gammel Kongevej 47 A, 4. tv., DK-1610 Copenhagen V (DK). **HALLBERG, Jonas, Tesch** [SE/DK]; Nordre Fasanvej(74) Agent: **NORANBROCK, Randy, A.**; Suite 300, 1700 Diagonal Road, Alexandria, Virginia 22314 (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: DYNAMIC LEVEL OF DETAIL



(57) Abstract: A system and method of interfacing Levels of Detail ("LoD" being a means for providing access to progressively more detailed information of a three dimensional model or scene) with User Interface Components (UICs), and is related to companion applications filed simultaneously, all of which are part and parcel of a method for browsing digital content and composing digital media and data via dynamic interaction of tiled images, external video, audio, text, and/or multimedia resources to create smoother and faster response time to user request, managed by a level of detail system which expands the ability of the user to create, manipulate, and/or interact with digital content and/or media via an application hosted in a browser or running on a desktop application which enhances the user ability to dynamically position and zoom into and out of views within a larger canvas.

FIG. 3



ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(48) Date of publication of this corrected version:

6 January 2011

(15) Information about Correction:
see Notice of 6 January 2011

Published:

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

Dynamic Level of Detail

Related Applications

[001] The instant application is related to and claims priority from U.S. Provisional Application 61/118,121, filed November 26, 2008, titled Content-Context Browser and is also related to application titled “Dynamic Image Tiling” and application titled “Dynamic Network Browser,” each filed on the same date, to the same inventors, and all of which are hereby incorporated by reference herein in their entirety.

Background

[002] One or more embodiments of the present invention relate to a system and method of interfacing Levels of Detail (LoD) with User Interface Components (UICs), and is related to companion applications filed simultaneously, all of which are part and parcel of a method for browsing digital content and composing digital media and data via dynamic interaction of tiled images, external video, audio, text, and/or multimedia resources to create smoother and faster response time to user request, managed by a level of detail system which expands the ability of the user to create, manipulate, and/or interact with digital content and/or media via an application hosted in a browser or running on a desktop

application which enhances the user ability to dynamically position and zoom into and out of views within a larger canvas.

[003] The concept of "Levels of Detail" (LoD) is known as a means for providing access to progressively more detailed information of a three dimensional (3D) model or scene.

[004] In the software application development process, developers associate non-visual data sources and logic with visual UICs. Users of the system then interact with these UICs, e.g., by use of a user input device such as a mouse, keyboard, touchscreen, etc., in order to manipulate the non-visual data sources.

[005] In the environment provided by at least one embodiment according to the invention, a plurality of virtual layers provides a challenge to the visual appearance of objects, such as interactive objects like menus. This is important to those developing applications to the environment provided by the embodiment.

Term list

Application – A computer software program comprising a set of executable or interpretable instructions for execution or interpretation by a processor.

API -Application Programming Interface – A documented way for developers to write software that works with a given application.

Client – An application which runs on a user's computer and occasionally communicates with one or more servers or other clients via an internet connection.

Server – An application that responds to requests from individual clients, interacting with a database to store and retrieve information on behalf of the client as needed. A server application may also facilitate communication between clients.

Summary

[006] An object of one or more embodiments of the invention is to provide a control of objects at different zoom levels of the frame. An implementation of LoD as a fundamental property of User Interface Components (UICs) is not believed by the inventors to be known in the art.

[007] Implementation is achieved by a method for providing at least one user interface component in a graphical user interface (GUI) at a plurality of zoom depth levels by associating a level of detail with at least one user interface control. Thereby, a user interface component is resizeable and/or repositionable to fit a given dimension.

[008] One embodiment allows software developers to define not only the standard geometry and associated data for display of the UIC on a screen, but also to define a list of UICs that are paired with a Level of Detail (LoD) object. An

application according to an embodiment comprises functionality for resizing and/or repositioning a UIC to a given dimension and, in certain additional embodiments, automatically select the correct UIC to be displayed. In at least some embodiments, the application comprises functionality for broadcasting messages of interest to UICs. The broadcast messages allow the selective or global setting of Levels of Detail for a receiving UIC, thereby providing automatic selection of the LoD appropriate for a display dimension, or to cause supporting UICs to transition to a specific LoD.

[009] One embodiment also allows the application to set the LoD either by specific reference to an identifier associated with a LoD, or by a real normal value (0>n>1) where 0 is the minimal and 1 the maximal Level of Detail supported by the UIC.

[0010] In one embodiment of the invention, a LoD library is implemented in ActionScript 3.0, but other programming languages may be used without departing from the scope of the embodiments, e.g., a structurally complete programming language may be used.

Brief Description of the Drawings

Fig. 1 depicts the LoD System Hierarchy of Objects,

Fig. 2-6 are screenshots of different level of details objects associated with a space,

Fig. 7 is a flowchart depicting interaction between LoD System Objects, and

Fig. 8 is a block diagram of a computer system usable in conjunction with an embodiment.

Detailed Description

Level of Detail (LoD) System Implementation

[0011] Fig. 1, “LoD Hierarchy,” is a flow chart depicting the relationship between the object types which comprise the LoD System.

[0012] The Level of Detail system comprises six unique object types.

LODConfiguration

[0013] An LODConfiguration (705) is a Dictionary, Object, or Hash table that stores an Array of LOD objects, and provides access to the stored LOD

information by numeric index, or textual identifier. LODConfiguration (705) objects map the current or expected geometric dimensions of a visible object to an LOD object. This configuration may be applied to an entire document and shared by all LOD objects, or assigned to individual objects.

[0014] The UI components defined by this configuration may be standard UI components, LODComponents, or LODContainers (704).

LOD Object

[0015] An LOD object associates a textual identifier, a LODThreshold (706), and a LODRenderer (708) definition. LOD objects are used to describe the class of UI component to be displayed to represent data whose on screen dimensions fall within the boundaries defined by the LODThreshold (706). For example, according to a given scenario if the displayed dimensions of a UIC fall below a predetermined LODThreshold (706) value, the UIC is not displayed and a specified UI component is displayed in place of the UIC. In at least some embodiments, the LODThreshold (706) specifies a range of values within which the dimensions must fall in order for the specified UI component to be displayed. In Fig. 2, a generic Level of Detail library has been loaded, and is rendering a specific document. At the displayed level of detail in this figure, only the topmost node of the document is visible, and is displayed at the minimum Level of Detail, in this case a blue rectangle.

[0016]LODThreshold (706)

[0017]In at least some embodiments, the LODThreshold (706) represents a geometric criteria defined by a minimum and maximum Rectangle. The LODThreshold (706) stores the minimum and maximum criteria for surface area, aspect ratio, width, and/or height for UI components on screen. In at least some embodiments, the LODThreshold (706) represents a shape other than rectangular, e.g., circular, elliptical, polygonal, etc. In at least some other embodiments, the LODThreshold (706) stores a single threshold criteria value.

LODRenderer(708)

[0018]The LODRenderer (708) is an object that communicates with a host application in order to request the correct visual component, i.e., the specified UI component, for representation of associated data at a specific visual size. LODRenderer (708) objects also control the transition between different visual components as the components are made available by the host application. A host application is any application or module comprising functionality to create a tree (709) of LOD Library objects, communicate geometric changes made to the display to the LOD Library objects, and/or authorize requests (703) for visual components from the tree of LODComponents (709). In Fig. 3, the user has zoomed the viewport in by manipulation of an input device. The change in zoom level has begun a transition to a new level of detail. At the new Level of Detail, the topmost node will be removed, and child document nodes will be displayed. This figure illustrates the crossfading transition at a halfway point between the old

and new visual components for the old and new levels of detail. In Fig. 4, the crossfading transition has completed, and the document is now displayed on screen using the new visual components that were assigned when the Level of Detail was changed as a result of user manipulation of the viewport.

[0019] The LODRenderer (708) supports synchronous and/or asynchronous messaging patterns for requesting and displaying UI components. The LODRenderer (708) also implements token-based authentication for the assignment of new UI views, so that the hosting application evaluates the requests for new UI components for authenticity, and thus prevents exploitation of the host application by dynamically loaded code which may or may not come from trusted sources.

[0020] LODRenderer (708) objects support different UI component creation policies, allowing the system to decide whether a component retains previously used LODRenderer (708) UI components in memory for reuse.

[0021] In at least some embodiments, the rendered UI components are either standard UI components, LODComponents (709), or LODContainers (704) and therefore the resulting UI functionality supports objects that:

1. switch between standard UI components

icon > button > list > detailed grid

In the example above, the LODRenderer (708) transitions from an icon, to

an interactive button, to a list summary, to a detailed summary as the Level of Detail is increased; and/or

2. switch from standard UI components to LOD enabled components or containers

icon > detailed image > LODContainer

In the example above, the LODRenderer (708) transitions from an icon, to a detailed image, to a LODContainer (704) which automatically instances and renders a new tree of LOD enabled child objects.

LODContainer (704)

[0022] The LODContainer (704) is a top-level container, and comprises a mix of LOD-enabled and/or standard UI components. As the user manipulates and zooms the view of the visual components, e.g., by the manipulation of a user input device, the LODContainer (704) manages the synchronization of LOD levels and individual child object visibility. The LODContainer (704) also determines child components visibility, and transmits messages that inform the child objects (709) of their current zoom level, and whether the child objects (709) need to update or request new visual representations as a consequence of user manipulation of the view. In Fig. 5, the user has zoomed the viewport in by manipulation of an input device. The change in zoom level has begun a transition to a new level of detail. At the new Level of Detail, the currently visible nodes

transition from grey rectangles to visual components capable of displaying bitmap image data. This figure illustrates the crossfading transition at a halfway point between the old and new visual components for the old and new levels of detail. In Fig. 6, the crossfading transition has completed, and the document is now displayed on screen using the new visual components that were assigned when the Level of Detail was changed as a result of user manipulation of the viewport.

[0023]The LODContainer (704) also comprises functionality for locking and unlocking some or all of the children into explicit Levels of Detail. This is achieved by calling the releaseLODControl() or lockLockLODControl() methods of child LODComponents. Locked LODComponent (709) objects ignore requests to reevaluate their current level of detail. The LODContainer (704) assigns a specific level of detail to child objects by calling the setRequiredLODLevel(n:Number) method of child objects. When assigning an explicit requiredLODLevel, n is a normal number value (0>n>1), where 0 is the lowest defined level of detail, and 1 is the highest level of detail available to the target LODComponent (709) .

LODComponent (709)

[0024]LODComponent objects comprise the core logic functionality to request and transition between visual LODRenderer (708) (UI) components based on the current or projected geometry of the LODComponent (709) and the criteria and LODRenderer (708) associations defined in the LODConfiguration (705).

[0025]The LODComponent (709) assigns DOM data to each LODRenderer (708) as it becomes ready, so that each renderer displays a new view onto the same data structure.

[0026]The LODComponent (709) also allows the system to lock or unlock the current LOD level, thus enabling or disabling the individual components' ability to change the LOD state.

Runtime Narrative

Fig. 7 is a flow chart depicting the LoD System progression in response to user requests.

1. A DOM is loaded into memory, and a top level LODContainer (704) is created and displayed on screen.
2. For each node in the DOM, an LODComponent (709) is created and stored in memory.
3. For each LODComponent (709), both a DOM node and a LODConfiguration (705) are assigned.
4. The top level LODContainer (704) informs the child LODComponents (709) of said LODContainer (704) of their current on screen dimensions, and invalidates their current visual state.

5. The LODComponents (709) examine their associated LODConfiguration (705) in response to invalidation of their current visual state, and determine the first LOD object whose LODThreshold (706) minimum is less than or equal to the current geometry of the object, and whose maximum is greater than the current geometry of the object. The LODComponent (709) then transmits a message to the host Application (702) requesting the correct visual component for the discovered LOD object.

6. The host application (702) examines the requests received from LODComponents (709), and if they are determined to be valid and authentic (703), instances a new object of the visual component requested by the LODComponent (709). This instance is returned to the requesting LODComponent (709). In at least some embodiments, authenticity is determined by shared keys, or keys issued to third party developers, or keys issued via a remote authentication process.

7. Each LODComponent (709) receives a reply from the host application, which, in at least some embodiments, contains a new UI component for display, or an error message describing the reason the request was denied. If the LODComponent (709) receives a new UI component for display, the LODComponent (709) assigns the associated DOM node as data for the UI

Component, and transitions the new UI Component to a visible state. Based on the LODComponents' (709) rendererCreationPolicy, the old UI component is either destroyed, or retained in memory for reuse.

8. The user manipulates the current view by panning, zooming, and/or rotating the current view.

9. The LODContainer (704) responds to the manipulation by returning to step 4 above, and the process is repeated.

[0027] FIG. 8 depicts a high-level functional block diagram of a computer system 800 usable in conjunction with an embodiment. Computer system 800 comprises a processor 802 (alternatively referred to as a processing or controller-based device), a memory 806, a network interface (I/F) 808, and an input/output device 804 communicatively coupled via a bus 810 or other interconnection communication mechanism.

[0028] Memory 806 (also referred to as a computer-readable medium) may comprise a random access memory (RAM) or other dynamic storage device, coupled to the bus 810 for storing data and/or instructions to be executed by processor 802. Memory 806 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 802. Memory 806 may also comprise a read only memory (ROM) or

other static storage device coupled to the bus 810 for storing static information and instructions for the processor 802.

[0029] In at least some embodiments, memory 806 also stores a set of executable or interpretable instructions, e.g., dynamic zoom/pan function 812, which, when executed or interpreted by the processor, cause the processor to perform a method according to an embodiment of the invention. In at least some embodiments, execution by processor 802 of the executable instructions causes the display of a user interface to a user of computer system 800 either via I/O device 804 or network I/F 808.

[0030] Network I/F 808 comprises a mechanism for connecting to a network. In at least some embodiments, computer system 800 comprises more than a single network interface. In at least some embodiments, network I/F 808 may comprise a wired and/or wireless connection mechanism.

[0031] A storage device, such as a magnetic disk, optical disk, or electromagnetic disk, may also be provided and coupled to the bus 810 for storing data and/or instructions.

[0032] I/O device 804 may comprise an input device, an output device and/or a combined input/output device for enabling user interaction. An input device may comprise, for example, a keyboard, keypad, mouse, trackball, trackpad, and/or cursor direction keys for communicating information and commands to processor 802. An output device may comprise, for example, a display, a printer, a voice synthesizer, etc. for communicating information to a user. In at least some

embodiments, I/O device 804 may comprise a serial and/or parallel connection mechanism for enabling the transfer of one or more of files and/or commands. In at least some embodiments, I/O device 804 is an optional component of computer system 800.

[0033] It will be readily seen by one of ordinary skill in the art that the disclosed embodiments fulfill one or more of the advantages set forth above. After reading the foregoing specification, one of ordinary skill will be able to affect various changes, substitutions of equivalents and various other embodiments as broadly disclosed herein. It is therefore intended that the protection granted hereon be limited only by the definition contained in the appended claims and equivalents thereof.

Claims

What is claimed is:

1. A method of providing at least one user interface control within a Graphical User Interface (GUI) at a variety of zoom depth levels by associating each level of detail with a user interface control; and whereby each user interface control is resizable and/or repositionable with respect to one or more dimensions.

2. An article of manufacture comprising a system and apparatus for providing for providing at least one user interface control within a Graphical User Interface (GUI) at a variety of zoom depth levels by associating each level of detail with a user interface control; and, whereby each user interface control can be resized and/or repositioned to any dimension.

LODThreshold <hr/> MaxArea() Number maxAspectRatio() Number maxHeight() void maxRect() void maxWidth() void MinArea() Number minAspectRatio() Number minHeight() Number minRect() Rectangle minWidth() Number toString() String	LODComponent <hr/> CurrentTransition Transition isLodViewRemovalAllowed Boolean LodConfig LODConfiguration findBestLod() LOD getFinalRenderer() UIComponent getOnScreenRect() Rectangle handleLODChange() void handleLODEvent() void intersectsStage() Boolean invalidateLOD() void isLODReady() Boolean lockLODControl() void lodChangeComplete() void lodCurrent() LOD lodRendererCurrent() LodRenderer MarkViewOldForDeletion() void releaseLODControl() void setRequiredLODLevel() void updateLodDisplay() void
LODRenderer <hr/> Configuration() Object dispose() void maxArea() Number maxHeight() Number maxWidth() number minArea() Number minHeight() number minWidth() Number renderer() LODRenderer rendererID() String threshold() LODThreshold toString() String	LODConfiguration <hr/> FindbestFit() LOD getLODAtLevel() LOD getLODByName() Lod lods() Array updateLODSort() void
LOD <hr/> IsRegistered Boolean renderer UIComponent rendererID String authFactoryResponseHandler() void authorizationResponse() void dispose() void rendererCreationPolicy() String requestRenderer() void uirequestResponse() void	LODContainer <hr/> AddChild() DisplayObject addChild() DisplayObject handleLODEvent():void invalidateLOD() void invalidateSize() void lockLODControl() void maximizeChildrenLOD() void releaseLODControl() void releaseLODControlToChildren() void removedAllChildren() void removeChild() void SetLODForAllchildren() void setTransformMatrix() void toString() String transformMatrix Matrix

Fig. 1

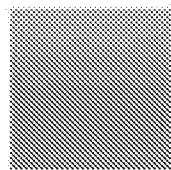


FIG. 2

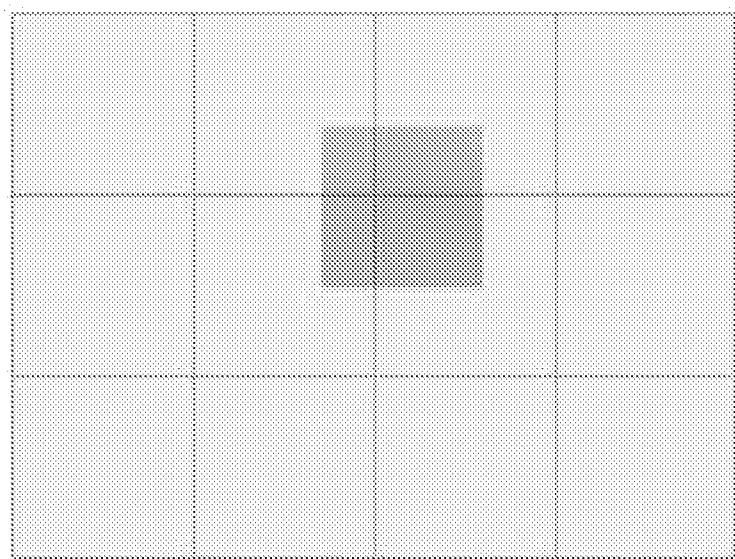


FIG. 3

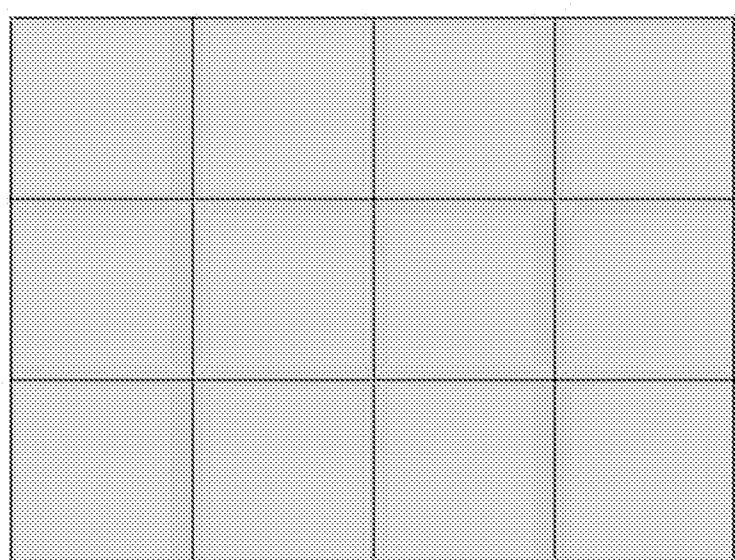


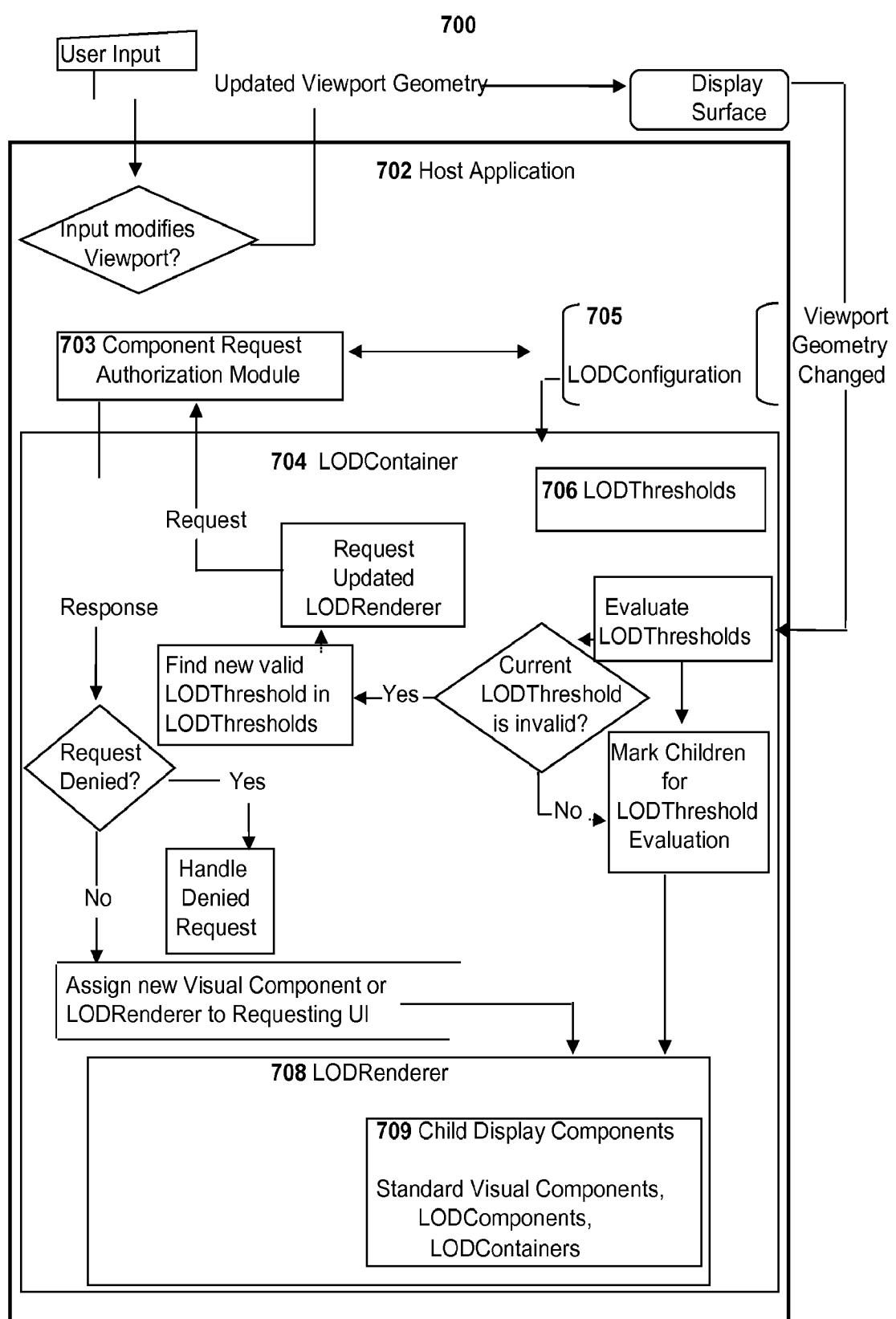
FIG. 4



FIG. 5



FIG. 6

**Fig. 7**

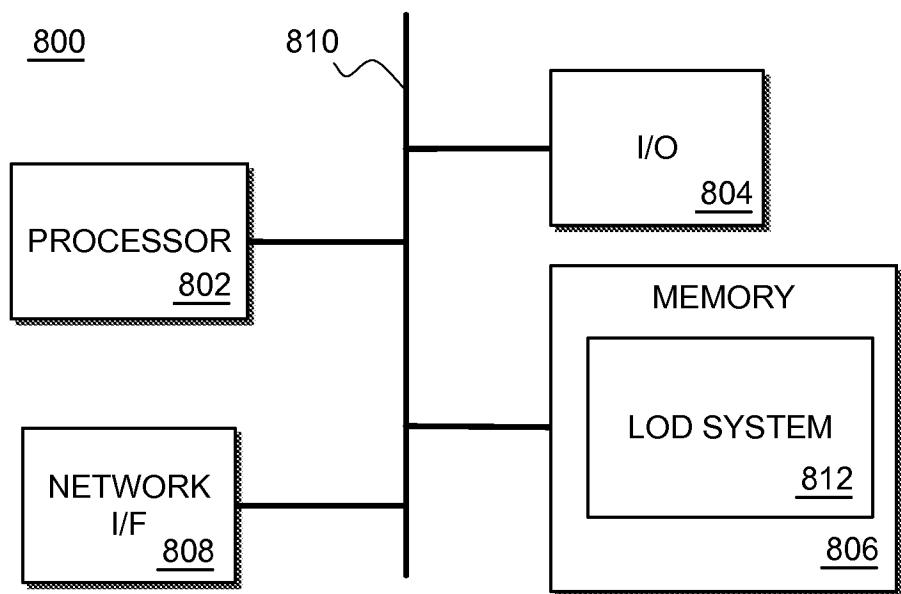


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 09/66028

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06T 17/00 (2010.01)

USPC - 345/428

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)

USPC 345/428

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC 345/418,422,428,581;709/223,224;700/1,90 (text search-see below)

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

PubWest (PGPB,USPT,EPAB,JPAB); Google Scholar

Search terms: level, detail, LoD, zoom, interface, UI, GUI, icon, image, graphic, symbol, control, resize, reposition

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006/0156228 A1 (GALLO et al.) 13 July 2006 (13.07.2006) FIG. 1 and para [0035]-[0036], [0054]-[0055], [0109]	1-2
A	US 2005/0184989 A1 (ARCAS) 25 August 2005 (25.08.2005) FIGs. 1-2, para [0026], [0037]-[0041]	1-2
A	US 2007/0208840 A1 (MCCONVILLE et al.) 06 September 2007 (06.09.2007) para [0007]-[0009]	1-2

Further documents are listed in the continuation of Box C.

* Special categories of cited documents:

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

"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

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

"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

"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)

"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

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

"&" document member of the same patent family

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

Date of the actual completion of the international search

15 February 2010 (15.02.2010)

Date of mailing of the international search report

04 MAR 2010

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-3201

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774