



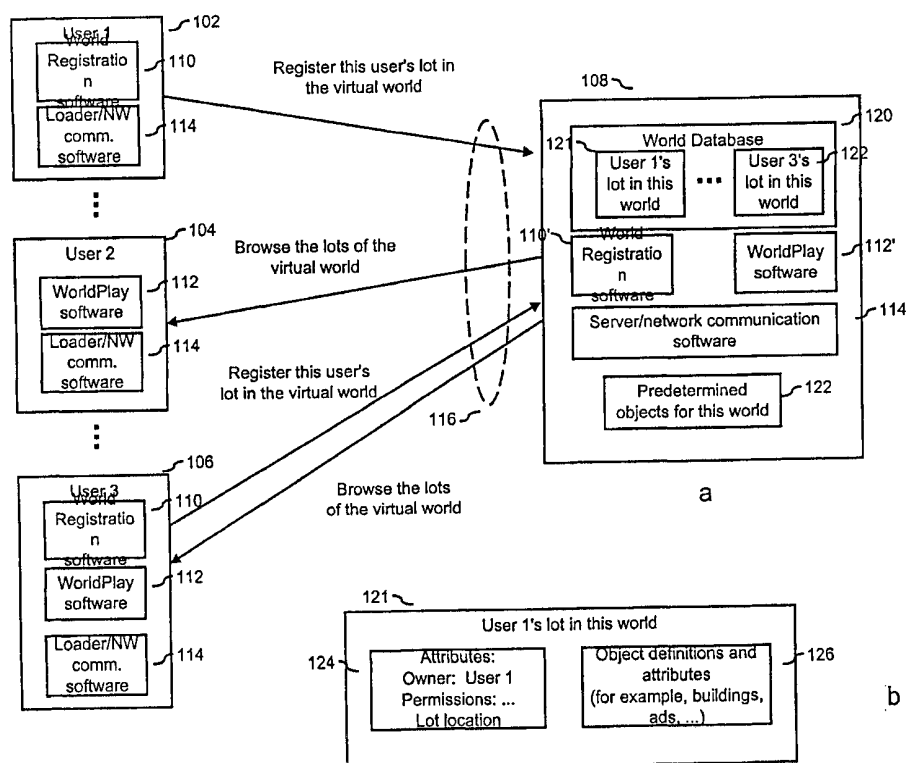
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 17/00		A2	(11) International Publication Number: WO 00/70489
			(43) International Publication Date: 23 November 2000 (23.11.00)
(21) International Application Number: PCT/US00/13181 (22) International Filing Date: 12 May 2000 (12.05.00) (30) Priority Data: 09/311,787 14 May 1999 (14.05.99) US 09/366,503 3 August 1999 (03.08.99) US (71) Applicant: GRAPHIC GEMS [US/US]; Suite 600, 1850 Gateway Drive, San Mateo, CA 94404 (US). (72) Inventors: JIAU, Winifred; 2520 Buena Vista Avenue, Belmont, CA 94002 (US). TOLAR, Robert; 57 Bradford Street, San Francisco, CA 94110 (US). (74) Agents: MAJERUS, Laura, A. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>	

(54) Title: METHOD AND APPARATUS FOR PROVIDING HOTSPOTS IN A SHARED VIRTUAL WORLD

(57) Abstract

A method and apparatus for providing hotspots and hotspaces in a shared three-dimensional virtual world. When the user in the virtual world activates a hotspace or a hotspot, an action occurs. Hot spots are areas of an object (or a whole object) that have an associated link. Hot spaces are two or three-dimensional areas or the virtual world (which may or may not have an object located therein) that have an associated link. For example, a link may cause a web page to be fetched and displayed. Activation of a hotspot or a hotspace link can also cause an action, such as display of an advertisement or an animation. Some of the types of links that can be associated with objects include: an absolute link to a lot, an absolute link to an object, a link that points to the interior or exterior of an object, a relative link specifying a direction and a number of units from the current location, a link with an associated action or animation, a link with a URL.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Method and Apparatus for Providing Hotspots in a Shared Virtual World

This application is a continuation under 37 C.F.R. § 53(b) of U.S. Application Serial No. 09/311,787 of Jiau et al. filed May 14, 1999.

5

BACKGROUND OF THE INVENTION

The present invention relates generally to computer networks and, specifically, to a method and apparatus that allow users to access and explore a multi-owner, three-dimensional virtual world in a network environment.

10 A “virtual world” is a world that does not actually exist in the physical world. For example, worlds described in books or movies can be virtual worlds. Worlds described by data stored in a computer system (“cyberspace”) are also a type of virtual world.

Some of the earliest computer games allowed users to explore virtual worlds described in text. These simple text games usually were executed on a single-user computer.

15 Eventually, however, games were developed that allowed users to explore these virtual worlds using a three-dimensional graphical interface. Eventually, games became multi-player games, allowing more than one player to roam freely through a predefined three-dimensional virtual world. For example, some three-dimensional games involve a first-person view as users move through tunnels or corridors. In some multi-player games, the
20 players can often see each representations of each other in the three-dimensional world as they move within some predetermined distance of each other.

Such three-dimensional games, however, involve a predetermined, predefined virtual world that does not change. The virtual world is defined by the game designer and the players move within the predefined spaces of the virtual world.

25 Similarly, certain “communities” such as MUDs (Multi-user Dungeons) or MUCKs involve one or more users moving through a predefined virtual world. These worlds are usually described in textual form only, and the user types his or her movements and actions using a text interface.

30 SUMMARY OF THE INVENTION

The described embodiment of the present invention allows users to explore and interact in a multi-owner, three-dimensional virtual world. A virtual world in accordance

with the present invention is divided into a plurality of “lots,” similar to “lots” of real estate in the real world. Users can reserve one or more of these lots to design as they please. The virtual world is made up of a plurality of the lots of different users laid next to each other.

In a described embodiment of the invention, users access the virtual world via client data processing systems using a networking model such as the World Wide Web, and a network such as the internet or an intranet. When a user initially designs his lot on his client system, he registers it with a server data processing system. The server assigns each newly registered lot to a location in the virtual world and stores the lots in a memory or database of the server. Thus, each lot in the virtual world has an ownership attribute identifying the user who registered it and owns it. Each lot also has a number of neighboring lots. A user can set permissions indicating who else can alter, visit, or otherwise view or copy his lot.

Some lots are accessible from other lots that are not located contiguously to the lot in the virtual world. Users can include objects in their lots that allow a virtual person walking in their lots to move to another lot that is not physically contiguous to their lot. For example, a user may define an object that allows a person who clicks on it or walks into it to change his location in the virtual world to a lot that is located far away in the virtual world (teleporting). As another example, a user may define an object that always moves a viewer to a certain object in the virtual world, wherever that object may be located within the virtual world. Users can also define objects in their lots that have certain behaviors, such as playing an ad or animation when selected. Users can also associate objects with the URLs of pages on the world wide web, so that a web page is displayed when the object is approached or activated.

All users, even those who do not have their own lots, can browse the lots of the virtual world, visiting in turn the lots that others have designed. In some embodiments of the invention, all lots have a common theme (such as a particular city or imaginary world). In other embodiments, there are no restrictions and each user is free to design his lot as he or she desires. One advantage of allowing multiple users to define the various lots of the virtual world is that the virtual world becomes a “community” created not by a single game designer or a single company, but created, instead, by the users inhabiting the virtual world. A virtual world in accordance with the present invention is fluid and changeable. Users can alter and modify their lots, even while others are visiting them in the virtual world. Thus, the lots of the virtual world can be constantly updated and changed to suit the desires and needs of the users of the virtual world.

Users browsing in the virtual world see the virtual world as a three-dimensional view (from either a first person or third person view). This three-dimensional scene typically allows a user to “see” more than one lot at a time. For example, a user looking at three-dimensional representation of the virtual world can typically “see” the lot in which he is currently standing and the lots directly in front of him and to either side. In certain embodiments, lots are visually separated by lot indicia, such as thick lines or dotted lines so that a user is aware when he crosses from one lot to another. In other embodiments, a user cannot necessarily see the divisions between lots. A user navigates through the lots of the virtual three-dimensional virtual world using a navigation interface that allows him to move forward and back; to turn; to tilt; and any other appropriate navigational motion. Lots typically have a square shape, but can have other shapes. Users typically own one lot, but can own any number of lots.

As the user moves forward in the virtual world, for example, he will eventually cross from his current lot into another lot and still other lots will become visible in the “distance.” The described embodiment allows the user to move seamlessly from lot to lot in the virtual world by implementing multithreaded loader software in the client data processing system that fetches and displays both lots and objects in the lots from the server in such a way that the user feels as if he is moving through an endless landscape. The client data processing system fetches and discards lots (including objects in the lots) and fetches objects within lots depending, in part, on the motions of the user as the user “moves” through the virtual three-dimensional world.

In accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method of implementing hotspots in a virtual world, comprising: assigning a part of an existing object in the virtual world as having a hotspot that is smaller than the total object; and when a user in the virtual world activates the hotspot, performing a hotspot activity in the virtual world associated with the activated hotspot.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method of implementing hotspots in a virtual world, comprising: assigning an existing object in the virtual world as having a hotspot that is the entirety of the object; and when a user in the virtual world activates the hotspot, performing an action in the virtual world associated with the activated hotspot.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method of implementing hotspaces in a virtual world, comprising: assigning a hotspace to a location in a virtual world that has no object at that location; and when a user in the virtual world activates the hotspace, performing a
5 hotspace activity in the virtual world associated with the activated hotspace.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to an apparatus that implementing hotspots in a virtual world, comprising: a software portion configured to assign a part of an existing object in the virtual world as having a hotspot having a link associated therewith; and a software portion
10 configured to perform a hotspot activity in the virtual world in accordance with the link, when a user in the virtual world activates the hotspot.

Advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by
15 means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this
20 specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1(a) is a block diagram of an exemplary embodiment of the present invention showing a plurality of client data processing systems and also showing a server data processing system upon which lots owned by various users are stored.

25 Fig. 1(b) is a block diagram of attributes in for one lot of the virtual world, including ownership attributes of the lot.

Fig. 2 is a flow chart showing steps performed by world play engine software in a client data processing system.

Fig. 3 is a grid representing a portion of a virtual world showing several lots, an object
30 in one of the lots, and a location of a user in one of the lots.

Fig. 4 shows a three-dimensional, first person view as seen from the user location in Fig. 3.

Fig. 5 is a grid representing a portion of a virtual world of Fig. 3, after the user has moved to a different location, and a different lot, in the virtual world.

Fig. 6 shows a three-dimensional, first person view as seen from the user location in Fig. 5.

5 Fig. 7(a) shows an example of how a user's field of vision in the virtual world covers several lots in the virtual world.

Fig. 7(b) shows another example of how a user's field of vision in the virtual world covers several lots in the virtual world.

10 Fig. 8 shows an example of information passed between a client data processing system and a server data processing system while a user is browsing the lots of a world.

Fig. 9(a) is a flow chart showing operation of objects having a link.

Fig. 9(b) shows example of different type of links that can be associated with an object.

15 Fig. 10(a) shows an example of a three-dimensional virtual world including a horizon bitmap of objects in the far distance.

Fig. 10(b) shows an example of the three-dimensional virtual world of Fig. 10(a) including a horizon bitmap of objects in the far distance, when the user has moved in a forward direction.

20 Fig. 10(c) shows an example of a three-dimensional virtual world including objects formerly shown as a horizon bitmap, when the user has moved still further in a forward direction.

Fig. 11(a) shows an example of a third person view of the three-dimensional virtual world of Fig. 3, including an avatar of the user.

25 Fig. 11(b) shows an example of a three-dimensional view of a virtual world having varying ground elevations.

Fig. 11(c) shows an example of a three-dimensional view of a virtual world having different ground textures in different lots.

Fig. 11(d) shows an example of a three-dimensional view of the virtual world of Fig. 3(a) having advertisements activated by user actions.

30 Fig. 11(e) shows an example of an object having an hot spot allowing a user to enter an interior of the object.

Fig. 11(f) shows an example of an interior of the object of Fig. 11(e)

Fig. 12 is a diagram showing an example of multithreaded processes on a client system .

Fig. 13(a) is an example of a data structure storing properties of a lot.

Fig. 13(b) is an example of a data structure storing properties of an object.

5 Fig. 14 is an example of an object model when the invention is implemented using ActiveX.

Fig. 15 shows another embodiment of the invention operating in single user mode.

Fig. 16 shows an example of a method performed by the server system to receive and register a lot defined by a user.

10 Fig. 17 shows an example of a file format describing a lot and the objects therein.

DETAILED DESCRIPTION OF EMBODIMENTS

Reference will now be made in detail to several embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever practicable, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

In the described embodiment of the present invention, many users have defined their own lots in a multi-user virtual world.

The following discussion assumes that a multi-user virtual world having lots assigned to various owners has been established at some time in the past and discusses how a user moves within the multi-user virtual world and reacts with it.

Fig. 1(a) is a block diagram of an exemplary embodiment of the present invention showing a plurality of client data processing systems and also showing a server data processing system upon which lots of various users are stored. The system of Fig. 1(a) includes a plurality of client data processing systems 102, 104, 106 and a client data processing system 108. First client system 102 includes world registration software 110 that allow the user to register his lot in the virtual world. Second client system 104 includes world play software 112 that allows a user to browse the various lots in the virtual world. Third client system 106 includes both world registration software 110 and world play software 112. There can, of course be more than three clients in communication with a server 108 at any time. All three client systems and the server system include software for loader/network (NW) communication software 114 that allows a client system to

communicate with server system 108 in order to load lots and objects from server system 108, and to receive broadcast messages from server system 108. Although only one server system is shown, it should be understood that the function of server system 108 could be distributed over more than one computer or processor. It is also understood that certain portions of the server database for objects and lots are stored in the client side, depending on the actions of the users in the client side.

It will be understood that client systems 102, 104, 106 and server system 108 are computers or similar data processing systems having one or more processors and appropriate data storage, such as memory or mass-storage devices. Word play software 112, world registration software 110, and network/loading software 114 are preferably stored as instructions in memory and are executed by one or more appropriate processors. All software described herein may also be stored on a computer-readable medium. A connection 116 between the client system 108 and server systems 102, 104, 106 can be a network, such as the internet, an intranet, a wireless network, or any similar network or connection capable of passing data between client and server. The client systems and/or the server system can be separate systems, as shown, or one or more can be part of the same system or computer. In the described embodiment, the client software is implemented as using Microsoft's ActiveX operating system extension and a browser, such as Microsoft's Internet Explorer browser, although the client software could also be implemented using Java applets or any other suitable programming language or paradigm. (Java is a trademark of Sun Microsystems, Inc.). The described invention is not limited to any particular programming language, operating system, or programming paradigm.

Server system 108 includes a world database 120. World database 120 includes definitions of the lots of the various users and forming the virtual world. World database 120 also includes pointers into a database 122 of objects that can be located in the various lots.

As will be discussed in additional detail below, in the example of Fig. 1(a), user1 registers a lot in the virtual world by sending a registration request to server system 108 to be registered and added to the virtual world. User2 does not have his own lot, but browses the various lots created by others. User3 registers his own lot and also browses the lots of others.

As shown in Fig. 1(a), lot 121 owned by user1 is stored in world database 120. Fig. 1(b) is a block diagram of attributes for lot 121 of the virtual world, including ownership attributes 124 indicating that the lot is registered and owned by user1. Note that the creator of

the lot, user1 is initially given “administrator” permission, which allow him to change the permissions of others with regard to the lot. Example permissions that can be assigned by the administrator to other users include read only (a user can view a lot, but cannot modify it); read/write (a user can view and modify a lot), none (a user cannot view a lot – if the user tries
5 to enter this lot in the virtual world, the lot will be shown as empty, or the user will not be able to advance); admin (administrative permission).

The definition of lot 121 also includes object definitions 126 (or object pointers to database 122) defining objects in user1’s lot. Object definitions include the location and properties of objects in the lot. Figs. 13(a) and 13(b) show exemplary attributes and properties
10 of lots and objects, respectively.

Fig. 2 is a flow chart showing steps performed by loading/network communication software 114 in a client data processing system. Fig. 2 will be described in connection with the examples of Figs. 3-6 below.

Fig. 3 is a grid representing a portion of a virtual world showing several lots, an object
15 320 in one of the lots, and a location 302 of a user in one of the lots. It should be understood that the virtual world grid of Fig. 3 extends in all directions for some predetermined virtual distance and that only part of the grid is shown for the sake of convenience. In Fig. 3, lots are shown as being separated by thick lines and a current location 302 of a user in the virtual world is shown by an “X.” The user is currently located in lot 304. If the user walks forward
20 far enough in the virtual world, he will pass into lot 306. If he walks still farther, he will pass into lot 308. If he turns to the left and walks, he will pass into lot 310. If he turns to the right and walks, he will pass into lot 312. As discussed above, each of these lots is owned and defined by a different user. Fig. 3 also shows an exemplary object 320 in lot 306. Various virtual worlds may use different scales for their grids. Thus, in one world, a lot may represent
25 a square mile, while in another virtual world, a lot may represent an acre or some other predetermined or variable area.

Fig. 3 shows a conceptual view of the virtual world. When a user browses the lots of the virtual world using world play software 112, what he actually sees is a three-dimensional representation, view, or scene of the virtual world rendered by his client data processing
30 system. Fig. 4 shows a three-dimensional window 402 displayed on a screen or monitor of a client system, including a three-dimensional view of the virtual world as seen from the user location in Fig. 3. The display also includes an exemplary navigation window 404 including

user interface elements well-known in the art (including, for example, forward, back, left, right, and tilt) to allow the user to indicate his movements within the virtual world.

In the three-dimensional view of Fig. 4, the user can see his current lot 304 and the lot directly ahead of him (lot 306). He can also see certain ones of the lots to either side. He
5 can also see any objects that are located within these neighboring lots, such as object 320. In certain embodiments, the ground is shown without the grid shown in the Figure. In certain embodiments, the ground is not flat (as shown), but has varying elevations. In this embodiment, ground elevation is part of the properties of a lot. In certain embodiments, the ground is not featureless (as shown), but has textures rendered in accordance with a bitmap or
10 similar representation of texture associated with the ground of that particular lot. In this embodiment, ground texture is part of the properties of a lot. Other embodiments do not include ground texture or elevation as lot attributes.

Fig. 5 is a grid representing a portion of a virtual world of Fig. 3, after the user has moved to a different location, and a different lot, in the virtual world. In the example of Fig.
15 5, the user has walked forward far enough (using navigation window 404) to cross the boundary between lot 304 and lot 306. Fig. 6 shows a three-dimensional, first person view as seen from the user location 302' in Fig. 5. Note that the world play software 112 of the client system has rendered the view to reflect the user's movement within the virtual world. The user has moved seamlessly into a new lot 306. When the user moves into a new lot,
20 additional lots in the virtual world become visible. In the example, lot 308, which is ahead of the user's position 302', has become visible in the distance.

Fig. 7(a) shows an example of how a user's field of vision in the virtual world covers several lots in the virtual world. As shown in Fig. 7(a) if the user is facing in a first direction, lots to the front and lots to the sides that fall within a cone-shaped area 702 will be visible in
25 a first person three-dimensional view. If the user is in the same location, but is turned to the side, different lots that fall within a different cone shaped area 704 will be visible. Thus, not only the user's location, but the direction he is facing (and the tilt from the ground that he is looking at) affect which lots in the virtual world will be visible to him in a three-dimensional display such as the displays of Figs. 4 and 6.

30 The flow chart of Fig. 2 provides details of how the user's movement within the virtual world affects the actions of the client systems and server systems. The user starts at a predetermined current location of a predetermined current lot in the virtual world. Initially,

before a three-dimensional representation can be displayed, the client system will request from the server and load all eight lots surrounding this current lot (step 202). Note that some of these lots are visible to the user and some are not. In the described embodiment, as lots are loaded into the client system, they are cached by the client system to avoid having to reload from the server. Thus, whenever a lot (or object) is needed, it is loaded from the server if it has not previously been cached (step 204). Visible lots (i.e., those lots that can be “seen” in the user’s current three-dimensional view of the virtual world) are loaded before non-visible lots (step 206). As described below in detail, non-visible lots are loaded by the client system on a lower priority level (step 210). Visible lots are rendered as soon as they are loaded (step 208). Details of a preferred multi-threaded loading implementation are shown in Fig. 12.

The client system requests visible lots from the server before it requests non-visible lots. The client software continually reassesses the priority of the lots (and objects within the lots) to be rendered so that it can provide needed lots quickly, giving the impression to the user that he is moving smoothly through the virtual world. In the described embodiment, the client software attempts to “guess” where the user will move next via a lookahead method and loads non-visible lots and objects of views so that they will be ready to render if the user moves in the anticipated direction. Thus, the client system may request objects or lots that are not currently visible if all visible lots are rendered and the client anticipates that the user will walk toward the non-visible lot next. The client may also request lots or objects that are on the horizon, but are in far away lots, so that the client system can generate “horizon bitmaps” as discussed below. World play software 112 includes rendering engine software that generates three-dimensional views of the virtual world in accordance with the lots loaded from the server system. The details of how to render three-dimensional views of lots and objects in a virtual world are not discussed herein in detail. Examples of appropriate three-dimensional rendering methods are discussed in J. Foley et al., Computer Graphics : Principles and Practice, Second Edition in C, 2nd edition, Addison-Wesley Pub Co; ISBN: 0201848406, which is herein incorporated by reference. In the described embodiment, a rendering engine renders in the background and blits to the display screen. Any appropriate rendering method can be used by the rendering engine to transfer the data describing the virtual world to the three-dimensional representation of the world.

If the user indicates a move forward or backward (step 212), he may possibly move into or out of his current lot. Whenever the user moves, the client system updates his current

location and point of view (POV) (step 220). If a lot boundary was crossed (step 228), the client system determines whether it needs to load new lots or objects (step 204, etc). In any case, movement by the user will cause the rendering software to re-render and display the current three-dimensional view (step 208). As an example, the client may need to request a new object, but not its lot, if it knows that the new object was recently added to a lot that was previously rendered and is now cached.

If the user indicates a left or right turn (step 214), he may possibly view a new lot, and will always change his direction of view (step 222). If a new lot is viewed, the client system determines whether it needs to load new lots or objects (steps 204, etc). In any case, movement by the user will cause the rendering software to re-render and display the current three-dimensional view (step 208).

If the user indicates a tilt up or down (step 218), he will not view a new lot (unless he tilts enough to turn over). The tilt of his POV will change (step 226). After the user indicates a tilt, the rendering software will re-render and display the current three-dimensional view (step 208).

If the user walks into an object or selects an object through a mouse click or similar means (step 216), the client system takes various actions (step 224), depending on how the object is defined. Some examples of objects are:

- 1) solid object – walking into this type of object will cause the user to stop (this type of object does not have links to any other object or lot)
- 2) object with interior - walking into this type of object will cause the user to view a completely different view representing the interior of the object (e.g., walking through a doorway). This type of object has a link to its interior. This movement can also be a teleportation to a completely different location in the virtual world, as described below. Some objects have interiors that are the same size as their exteriors. Some objects have interiors that are larger or smaller than their exteriors. These objects will use a sub-coordinate system for locations within their exteriors.
- 3) object with pass-through - walking into this type of object will cause the user to pass through to the other side (e.g. object with a tunnel) (This type of object does not necessarily have any links).

- 4) object with associated action - walking into this type of object will cause the user to view an action, such as a pop-up ad or an animation.
- 5) teleportation object - walking into this type of object will cause the user to change location to a completely different location (and possibly a completely different lot) in the virtual world. Teleportations can be to a fixed location in the virtual world (such as a friend's object or lot) or to a moving location (such as a mobile object that moves around the virtual world). Teleportations can also be to a relative location, such as "three lots to the right" or "one step forward." (This type of object has links to other objects or lots, examples of which are shown in Fig. 9(b)).
- 6) URL object. This object has an associated link in the world wide web that causes a web page to be displayed.

Fig. 9(a) is a flow chart showing operation of an object having a link. As discussed above, when a user selects (via a mouse click, etc.) an object that is defined as having an associated link, the client system takes various actions, depending on the type of link. A link can cause the client system to teleport the user (step 904), either to an absolute location or a relative location. An absolute link can be a link to another lot or an object having a fixed location. An absolute link can also be a link to an interior/exterior view of the same object. This link may reference a sub-coordinate system within an object. An absolute link can also be a link to a specific object that does not have a fixed location (such as a car that moves around in the virtual world). A relative link can be a link to a location relative to the current position (e.g., a predetermined number of units in a predetermined direction from the current location). A link can also contain a web address, such as a URL (step 916). When the user clicks on such a link or approaches a hotspot or hotspace having such a link, a web page is fetched and displayed (possibly in a separate window). A link can also cause an action (step 910), such as display of an advertisement or an animation (which may have to be fetched from the server system).

Fig. 9(b) shows example of various types of links that can be associated with objects. A first type of absolute link is an absolute link to a specific lot in the virtual world. For example, the lot may be identified by a lot ID or by a lot name that can be resolved to a lot ID and, possibly, by a position in the lot. A second type of absolute link is an absolute link to an object, no matter where that object is located in the virtual world. For example, the object

may be identified by an object ID or by an object name that can be resolved to an object ID. A third type of link is a link that points to the interior or exterior of an object (indicated by, for example, an ID or a name). Note that, although this is technically a teleportation, the user does not actually change his location in the virtual world. A fourth type of (relative) link specifies a direction, unit (e.g., blocks, steps, etc) and a number of units from the current location. Another type of link has an associated action or animation. Another type of link has an associated URL.

Fig. 8 shows an example of information passed between a client data processing system and a server data processing system while a user is browsing the lots of a world.

Actions performed by a client system are shown on the left of the diagram. Actions performed by a server system of shown in the right of the diagram. In step 801, the user's system 106 requests the lots and/or objects needed to display an initial user location. It is possible that these lots (or any lots shown as fetched from the server) are stored in cache or a similar local memory in the user's system 106. In steps 802 and 804 rendering engine software requests and receives from the server initial lots and objects of a virtual world. The server returns the requested lots (and objects). The client initially requests all lots/objects in the eight lots surrounding the user, requesting the visible lots first. The rendering software displays the lots and objects in a three-dimensional display, such as that shown in Fig. 3.

When the user moves within the virtual world, the client system determines which lots and objects have become visible in accordance with the current lot, direction, and tilt of the user's location and POV. If the visible world changes, the client system requests and receives new objects and/or lots in steps 806 and 808. The client system also does lookahead and prefetching of lots and objects that its believes will be entered by the user in the future. As the user moves within the virtual world, the client renders the three-dimensional virtual world in real-time so that the user sees a seamless moving view of the virtual world. Objects, such as avatars or other users or moving objects, can also enter the user's vicinity. When objects move or when a user changes an object in the virtual world, the server broadcasts a message to clients in step 812. The broadcast information identifies the moved object and its new location or description. The client stores the broadcast information in step 814 and uses the broadcast information to change the user's visible three-dimensional world if needed. Note that empty lots (containing no objects) are preferably rendered by the client without having to load information from the server.

Fig. 10(a) shows an example of a three-dimensional virtual world including a horizon bitmap of objects in the far distance. In the described embodiment, only lots that are within one lot in front of a user's current position are displayed as above the horizon. In the described embodiment, objects that are in lots farther ahead than the horizon are displayed as horizon bitmaps. These lots are fetched and objects therein are sent to the rendering engine so that they can be rendered as horizon bitmaps instead of objects. Fig. 10(a) shows a horizon bitmap 1002 that represents objects in a lot in the far distance. In the described embodiment, as the user moves toward the distant lot as shown in Fig. 10(b), the horizon bitmap 1002 does not change size. Eventually, when the user gets close enough (a predetermined virtual distance) the objects are rendered as individual objects, instead of being rendered as a bitmap. Fig. 10(c) shows the objects that were formerly rendered as a horizon bitmap rendered as individual objects 1050. The use of a horizon bitmap allows the user to receive visual clues about objects in the distance, but avoids having to render all distant objects.

Fig. 11(a) shows an example of a third person view of the three-dimensional virtual world of Fig. 3, including an avatar of the user 1102 and an avatar of another user 1104. When the other user's avatar 1104 moves into the vicinity of the user, server system 108 broadcasts a message telling of the avatar's movement. When the avatar 1104 enters the visible area of the user, the rendering system renders the other user's avatar 1104. Other user's avatars are also visible in first person mode (such as that shown in Fig. 3), even if the first person user's avatar 1102 is not visible.

Fig. 11(b) shows an example of a three-dimensional view of a virtual world having varying ground elevations. A high elevation 1152 occurs in two lots 1162 and 1164. A low elevation (e.g., a hole) 1154 also occurs in lot 1162. These ground elevations are preferably part of the attributes of the lots in which they occur and are rendered by the rendering engine. Lots having ground elevations are not considered to be empty, even if they have no objects therein, and must be fetched from the server at least the first time they are encountered.

Fig. 11(c) shows an example of a three-dimensional view of a virtual world having different ground textures in different lots. In the example, lot 1162 has a ground texture that does not include a grid pattern, but does include triangles drawn on the ground of the lot. These triangles are preferably defined as bit maps in the attributes of the lot 1162. In the example, lot 1164 has a ground pattern that includes a grid pattern (which may be a default

pattern) and circles. These circles are preferably defined as bit maps in the attributes of the lot 1164. Lots may also have both elevations and ground patterns.

Fig. 11(d) shows an example of a three-dimensional view of the virtual world of Fig. 3(a) having advertisements activated by user actions. In the example, object 320 has a pop-up advertisement 1182 that is activated when the user's location becomes within a predetermined virtual distance in the virtual world. When the user approaches object 320, advertisement 1282 pops up. This activity is part of the attributes of object 320 (see Fig. 13(b)). There is a hotspot link defined around object 320.

Although the links of Fig. 9(b) are described as being associated with "objects" a link can also be associated with a hotspot or a hotspot. Fig. 11(d) further shows a static object 1184 that is always visible when it comes into view in the three dimensional representation of the virtual world. The object 1184 is a sign containing an advertisement. There is a hotspot link 1186 defined on object 1184. When the user clicks or otherwise selects hotspot link 1186 (e.g. by approaching near it), an animation 1188 (represented in the figure by a star) spins, changes color, animates, etc. Hotspot 1186 and the activity associated with object 1184 (as well as the appearance of the object, including the advertising text) are defined as attributes of the object 1184 (see Fig 13(b)).

Fig. 11(e) shows an example of an object 1192 having an hotspot link 1194 (e.g., a door-shape) allowing a user to enter an interior of the object. In the example, the user enters the interior of the object when he activates the hotspot 1192 (i.e., when he collides with or passes sufficiently close to the hotspot). Fig. 11(f) shows an example of an interior of object 1192 of Fig. 11(e). When a user activates hotspot 1194 of Fig. 11(e), the user passes to the interior of object 1192 and client computer 106 renders an interior view of the object. The interior view of Fig. 11(f) is preferably defined as an attribute of object 1192. In the example, the interior of object 1192 is the same size as its exterior, although this does not have to be the case. If the interior and exterior are different sizes, a sub-coordinate system is used for the interior of the object. In the example, if the user turns around in the interior of the object, he will see the inside walls of the object. Other objects that appear similar to object 1192 can have hotspots that are effectively teleportation links, so that the user steps through a door, but ends up in another location in the virtual world (or in a sub world or a related virtual world). Hot spots are defined as areas of an object (or a whole object) that have an associated link. Hot spaces are defined as two or three-dimensioned areas or the

virtual world (which may or may not have an object located therein) that have an associated link.

Fig. 12 is a diagram showing an example of multithreaded processes on a client system. In the example, the client system includes a navigator thread 1202, a queue manager thread 1204, a loader queue 1206, a background loader thread 1208, and a rendering engine thread 1210. Loader queue 1206 includes lot ids sorted in accordance with the highest loading priority. Each lot id entry in loader queue 1206 references a plurality of object ids, which are sorted in accordance with the highest loading priority. Thus, for example, when another user enters a lot, the object for the other user's avatar will be within the virtual world the object id of the avatar will be added to the object list for that lot in the loader queue. Thus, for example, when a user selects a link, clicks on a link or activates a hotspot or a hotspot in the virtual world, the navigator software reports this action to a queue manager and the queue manager adds the lot or object to the loader queue at an appropriate priority level. The sorting software of the queue manager is continually reorganizing the loader queue in accordance with the user actions and with server broadcast messages. As an example, all lots that could appear below the horizon are fetched as a low priority and their objects used to generate a horizon bitmap if appropriate.

The sorting software of the queue manager contains internal rules that help determine which objects and lots should be loaded next to allow the user to have a seamless view of the virtual world. The objects and lots that are to be loaded from the server next are placed at the head of the loader queue 1206. By default, objects and lots that have newly entered the visible virtual world are moved to the head of the loader queue. Hotspots and hotspaces are also added to the loader queue at a lower priority than visible objects. Non-visible, but close-by, lots are added to the loader queue at a still lower priority. The specific priority rules used by the sorter software may differ in various embodiments of the present invention. As discussed above, objects or lots that the user has no permission to enter or view will not be fetched from the server and will not be displayed (for example, minors cannot enter adult lots).

A virtual world can also include "hotspots" and "hotspaces" that cause the performance of a predefined action or teleportation. Objects can have predetermined "hotspots" that cause actions to occur when the user touches or selects the object. For example, when the user touches or selects an object, a pop-up advertisement may appear. In

contrast, a “hotspace” is a predetermined space surrounding an object. For example, when a user walks within ten virtual feet of a hotspace, a pop-up advertisement may appear. In the example, even though the hotspot or hotspace was previously within the visible field of the user, the user’s movements cause the client to display the pop-up advertisement. Alternately,
 5 the client may have pre-requested the pop-up advertisement by way of its lookahead method when the user moved closer to the object.

Background loader 1208 requests lot ids and object ids from the queue manager and sends requests to the server system for the needed lot or object (assuming the lot or object is not cached locally). As soon as the object or lot is received from the server by the
 10 background loader, it is sent to the rendering engine to be rendered on the display of the client system. In the described embodiment, the entire three dimensional screen display of the visible portion of the virtual world is redrawn in its entirety. Alternately, only portions (such as pop-up objects) that have changed may be redrawn.

Fig. 13(a) is an example of a data structure showing properties of a lot stored in server system 108. The data structure of Fig. 13(a) is an example of the information sent between
 15 the server system and the client system. Exemplary properties include a lot name 1302, a unique lot id 1304, a list of current visitors/objects/avatars/chat boxes 1306, a user id 1308, permissions 1310, a list of 3D horizon bitmaps 1312 and a list of ground bitmaps 1314 (describing the appearance of the ground in the lot), along with a list of ground elevations for
 20 the lot (describing the height of pieces of the lot in relation to a reference height). It should be understood that the properties shown are provided only for the purposes of explanation and that other appropriate properties could belong to lots in the virtual world. The use id 1302 is the id of the user that registers and owns the lot. This user can change the permissions of other users.

Fig. 13(b) is an example of a data structure showing properties of a lot stored in server system 108. The data structure of Fig. 13(a) is an example of the information sent between
 25 the server system and the client system. The exemplary properties of Fig. 13(b) include a name 1352, a unique lot id 1354, a type of object 1356 (for example, avatar, movable by user, non-movable, moves under own power and algorithm, has an interior/exterior), list of
 30 hotspots and hotspaces 1358, pointer/reference to object database (either explicitly by lot or by user id) 1360, user id of this object 1362, permissions of this object 1364 (e.g., R, R/W, none, admin), list of interactive properties associated with this object (e.g., pop-up ads) 1366.

It should be understood that the properties shown are provided only for the purposes of explanation and that other appropriate properties could belong to lots in the virtual world. The use id 1362 is the id of the user that defined and owns the object. This user can change the permissions of other users for the object. An object in a lot may have a different owner
5 than the owner of the lot. For example, an avatar owned by one user may be present in the lot of a second user. While the avatar of the first user may enter and drop off of the lists 1206 of objects in the second user's lot, objects placed by the user into his lot are permanently located in the lot and are permanently on the object list 1206. (Unless they are objects that can be moved around by the user or that move according to their own rules).

10 Fig. 14 is an example of an object model when the invention is implemented using ActiveX. The virtual world includes a game object layer, a 3D engine abstract object layer, a DirectX layer, and a display hardware level.

Fig. 15 shows another embodiment of the invention operating in single user mode. In this mode, a client system does not interact with the server, but allows the user to browse
15 within his own lot. The user can only enter areas of the virtual world defined by himself. No information is fetched from the server during single user mode, since all information about a user's own lots and objects are assumed to be located in his client system.

Any acceptable way of registering a lot by a user may be used with the present invention. Fig. 16 shows an example of a method performed by server system 108 to receive
20 and register a lot defined by a user. In step 1602, the server receives the definition of the user's lot from the user. In certain embodiments, the user may optionally define where in the virtual world he would like to place his lot. The user may, for example, choose from a map of available lots displayed on a computer screen, printed, or e-mailed to the user. The user may define his desired lot location as an absolute location (such as "lot number 500") or he
25 may define his desired lot location as relative to the lot of another ("next to or within three lots of A's lot"). He may also define his desired lot location as relative to an object in the virtual world ("next to or close to the Golden Gate Bridge"). This object may be either a predetermined static object in the virtual world (such as a bridge or a mountain) or may be a movable object ("near wherever A's avatar is currently located"). Any other appropriate
30 ways of indicating a lot location in the virtual world can be used by the user to indicate his desired lot location. The user specifies the objects in his lot (by size, appearance, links, etc) and specifies any hotspots or hotspaces in his lot.

Step 1608 determines whether the user has permission to determine his lot location. For example, in certain embodiments, users may pay a premium to be able to place their lots where they want. If the user has this permission and, in step 1610, the desired lot location is available to this user, then the server 108 registers the user's lot and places it in the user's
5 desired location in the virtual world. Server 108 maintains a data structure, such as a table indicating available lots in the virtual world and a data structure, such as a table, indicating which users have permission to choose their own lot locations.

After the new lot is registered, other user's entering this location of the virtual world will enter the newly registered lot. Server 108 also registers any objects, hotspots,
10 hotspaces, links, etc. in the new lot that have been defined by the user as part of his lot definition. Server 108 also sets permissions for the newly registered lot. In the described embodiment, the lot has an attribute identifying the user who defined it and is its owner. In the described embodiment, the new user is assigned admin permissions and can define who else may enter his lot, and/or view or change his lot. If the user does not have the ability to
15 define his lot location or the desired location is not available, the user is asked to provide another location (or a default location is used).

If the user does not indicate a desired lot location, in step 1604, server 108 places the new lot in a next available lot location in the virtual world. Server 108 maintains a data structure, such as a table, indicating a "next available" location. Lots may be assigned by the
20 server in the virtual world according to any appropriate lot assignment scheme. Some of these lot assignment schemes include: at random, in rows, in columns, by area of interest or by affinity group (as indicated by the users), according to the level of service purchased by the user for the virtual world (higher levels are assigned better locations), in advertising or non-advertising areas of the virtual world, according to age (as indicated by the users),
25 according to ground texture chosen by the users (for example, all triangle-covered lots are located together, or all triangle-covered lots are not located together), whether the user's lot is defined as interior or exterior, or as a sub-world having a portal to another world.

Fig. 17 shows an example of a file format describing a lot and the objects therein. The file format is a file storing information for a lot. A file preferably includes a header and a
30 linked list of objects associated with the lot. Each lot has a lot id and ownership and permission attributes (not shown, since they are in a separate file). Each object has an object id, a description of the object, and a link. The lot file is preferably sent to the client and

stored in the client in the format shown. The lot file can be stored in the same format on the server or can be stored as a database on the server and converted to the format shown before being sent to the client.

5 Although the embodiment of the present invention described above uses a flat, two-dimensional display to represent the three-dimensional virtual world, other embodiments of the invention could represent the virtual world using appropriate virtual reality technology, such as a stereoscopic viewer, for example.

10 While the invention has been described in conjunction with a specific embodiment, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims and equivalents.

WHAT IS CLAIMED IS:

1. A method of implementing hotspots in a virtual world, comprising:
assigning a part of an existing object in the virtual world as having a hotspot
that is smaller than the total object; and
5 when a user in the virtual world activates the hotspot, performing a hotspot
activity in the virtual world associated with the activated hotspot.
2. The method of claim 1, wherein the hotspot activity is transporting the user to
another location in the virtual world.
- 10 3. The method of claim 1, wherein the hotspot activity is changing the user's location
inside the object having the hotspot.
4. The method of claim 1, wherein the hotspot activity is activating an animation
15 associated with the hotspot.
5. The method of claim 1, wherein the hotspot is activated by the user clicking on the
hotspot.
- 20 6. The method of claim 1, wherein the hotspot is a hotspace that is activated by the
user approaching nearer than a predetermined virtual distance to the hotspace in the virtual
world.
7. The method of claim 1, wherein the hotspot is activate by the user colliding in the
25 virtual world with the object having the hot spot.
8. The method of claim 1, wherein an object has a plurality of hotspots.
9. The method of claim 1, wherein an object has a plurality of hotspots, at least two
30 of the plurality of hotspots having different hotspot activities associated therewith.

10. The method of claim 1, wherein an object has a plurality of hotspots, at least two of the plurality of hotspots capable of being activated separately.

11. A method of implementing hotspots in a virtual world, comprising:
5 assigning an existing object in the virtual world as having a hotspot that is the entirety of the object; and
 when a user in the virtual world activates the hotspot, performing an action in the virtual world associated with the activated hotspot.

10 12. A method of implementing hotspaces in a virtual world, comprising:
 assigning a hotspot to a location in a virtual world that has no object at that location; and
 when a user in the virtual world activates the hotspot, performing a hotspot activity in the virtual world associated with the activated hotspot.

15 13. The method of claim 12, wherein the hotspot activity is transporting the user to another location in the virtual world;

 14. The method of claim 12, wherein the hotspot activity is changing the user's
20 location to the inside of an object in the virtual world.

 15. The method of claim 12, wherein the hotspot activity is activating an animation associated with the hotspot.

25 16. The method of claim 12, wherein the hotspot is activated by the user clicking on the hotspot.

 17. The method of claim 12, wherein the hotspot is activated by the user
approaching nearer than a predetermined virtual distance to the hotspot in the virtual world.
30

 18. The method of claim 12, wherein the hotspot is activate by the user walking through the hotspot in the virtual world.

19. An apparatus that implementing hotspots in a virtual world, comprising:
a software portion configured to assign a part of an existing object in the
virtual world as having a hotspot having a link associated therewith; and

5 a software portion configured to perform a hotspot activity in the virtual world
in accordance with the link, when a user in the virtual world activates the hotspot.

20. An apparatus that implements hotspaces in a virtual world, comprising:
a software portion configured to assign a hotspot to a location in a virtual
10 world that has no object at that location; and

a software portion configured to perform a hotspot activity in the virtual
world associated with the activated hotspot, when a user in the virtual world activates the
hotspot.

15 21. A computer program product, comprising:
a computer readable medium having instructions thereon to implement hotspots in a
virtual world, including:

computer program devices configured to assign a part of an existing object in
the virtual world as having a hotspot that is smaller than the total object; and

20 computer program devices configured to perform a hotspot activity in the
virtual world associated with the activated hotspot, when a user in the virtual world activates
the hotspot.

22. The computer program product of claim 21, wherein the hotspot activity is
25 transporting the user to another location in the virtual world.

23. The computer program product of claim 21, wherein the hotspot activity is
changing the user's location inside the object having the hotspot.

30 24. The computer program product of claim 21, wherein the hotspot activity is
activating an animation associated with the hotspot.

25. The computer program product of claim 21, wherein the hotspot is activated by the user clicking on the hotspot.

26. The computer program product of claim 21, wherein the hotspot is a hotspace that is activated by the user approaching nearer than a predetermined virtual distance to the hotspace in the virtual world.

27. The computer program product of claim 21, wherein the hotspot is activate by the user colliding in the virtual world with the object having the hot spot.

28. The computer program product of claim 21, wherein an object has a plurality of hotspots.

29. The computer program product of claim 21, wherein an object has a plurality of hotspots, at least two of the plurality of hotspots having different hotspot activities associated therewith.

30. The computer program product of claim 21, wherein an object has a plurality of hotspots, at least two of the plurality of hotspots capable of being activated separately.

31. A computer program product, comprising:
a computer readable medium having instructions thereon to implement hotspots in a virtual world, including:

computer program code devices configured to assign an existing object in the virtual world as having a hotspot that is the entirety of the object; and
computer program code devices configured to perform an action in the virtual world associated with the activated hotspot, when a user in the virtual world activates the hotspot.

32. A computer program product, comprising:
a computer readable medium having instructions thereon to implement hotspots in a virtual world, including:

computer program code devices configured to assign a hotspot to a location in a virtual world that has no object at that location; and

computer program code devices configured to perform a hotspot activity in the virtual world associated with the activated hotspot, when a user in the virtual world
5 activates the hotspot.

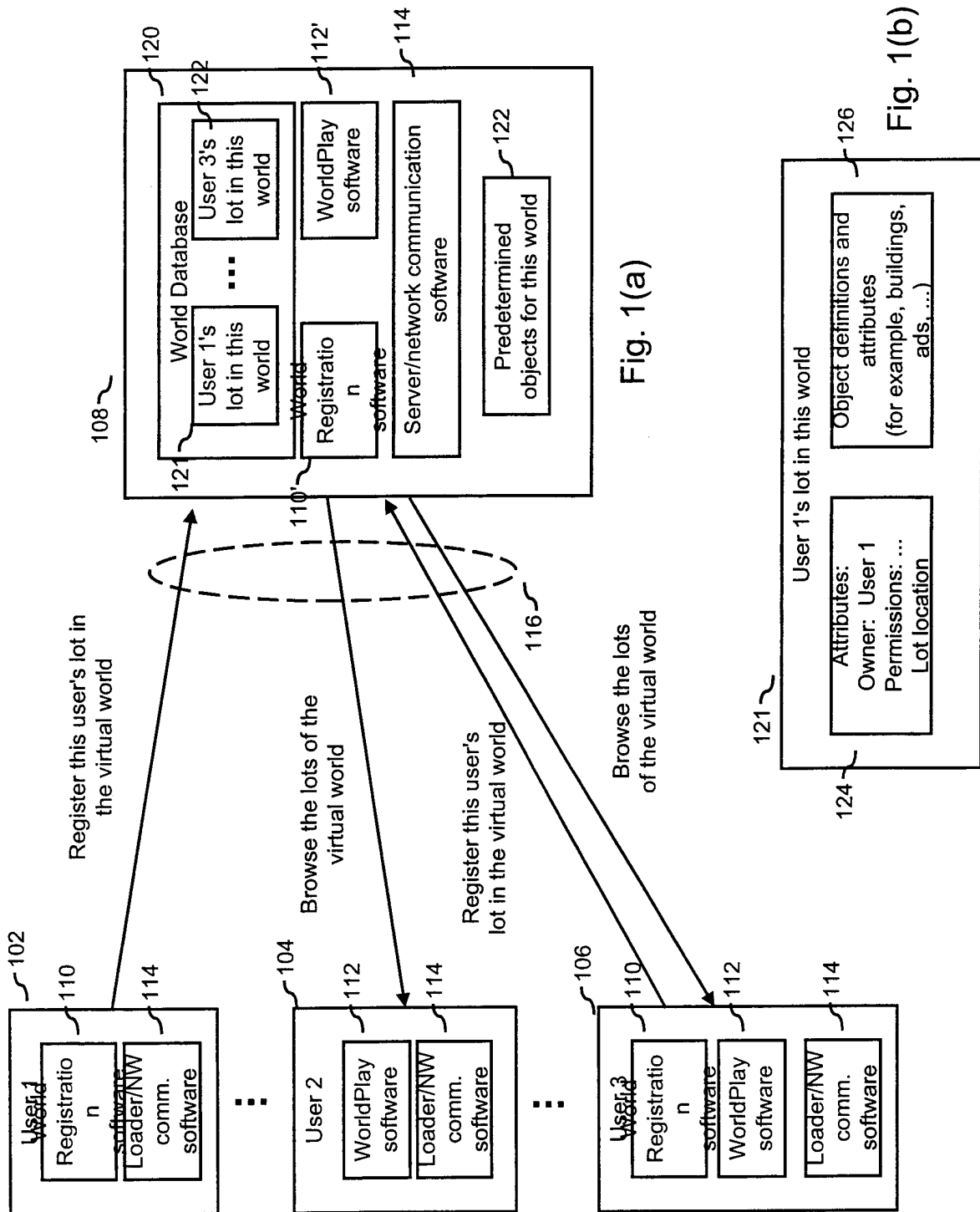
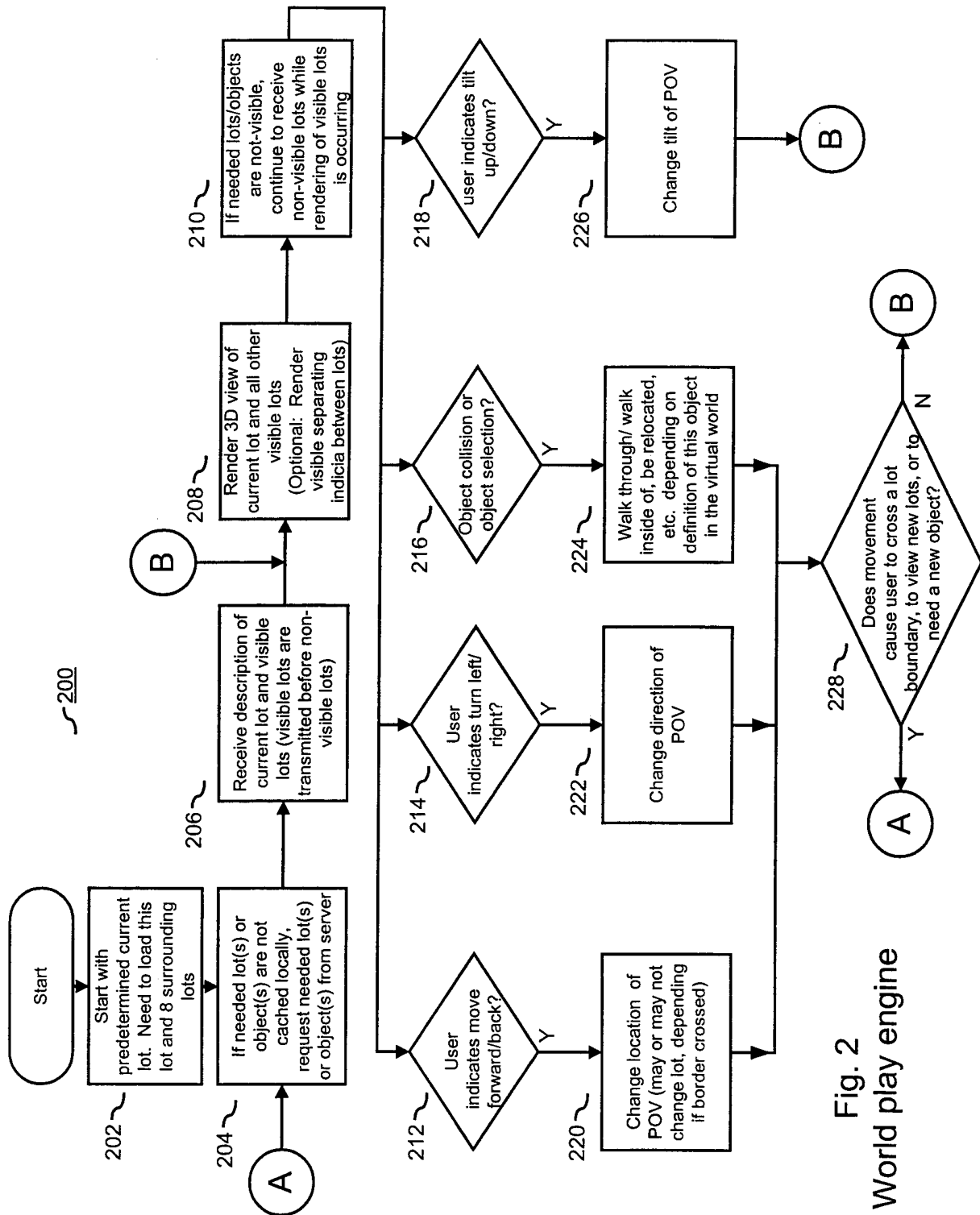


Fig. 1(a)

Fig. 1(b)



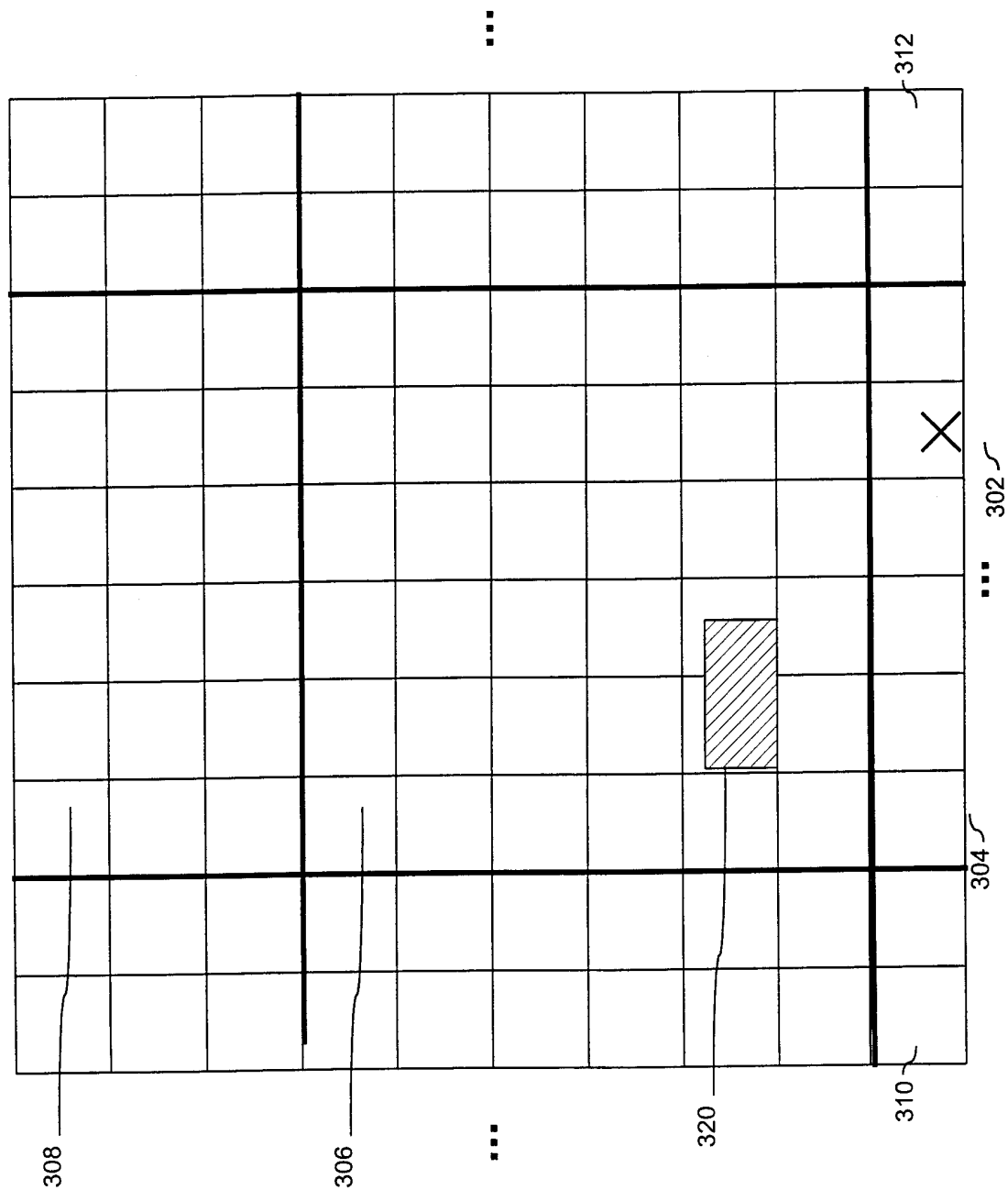


Fig. 3

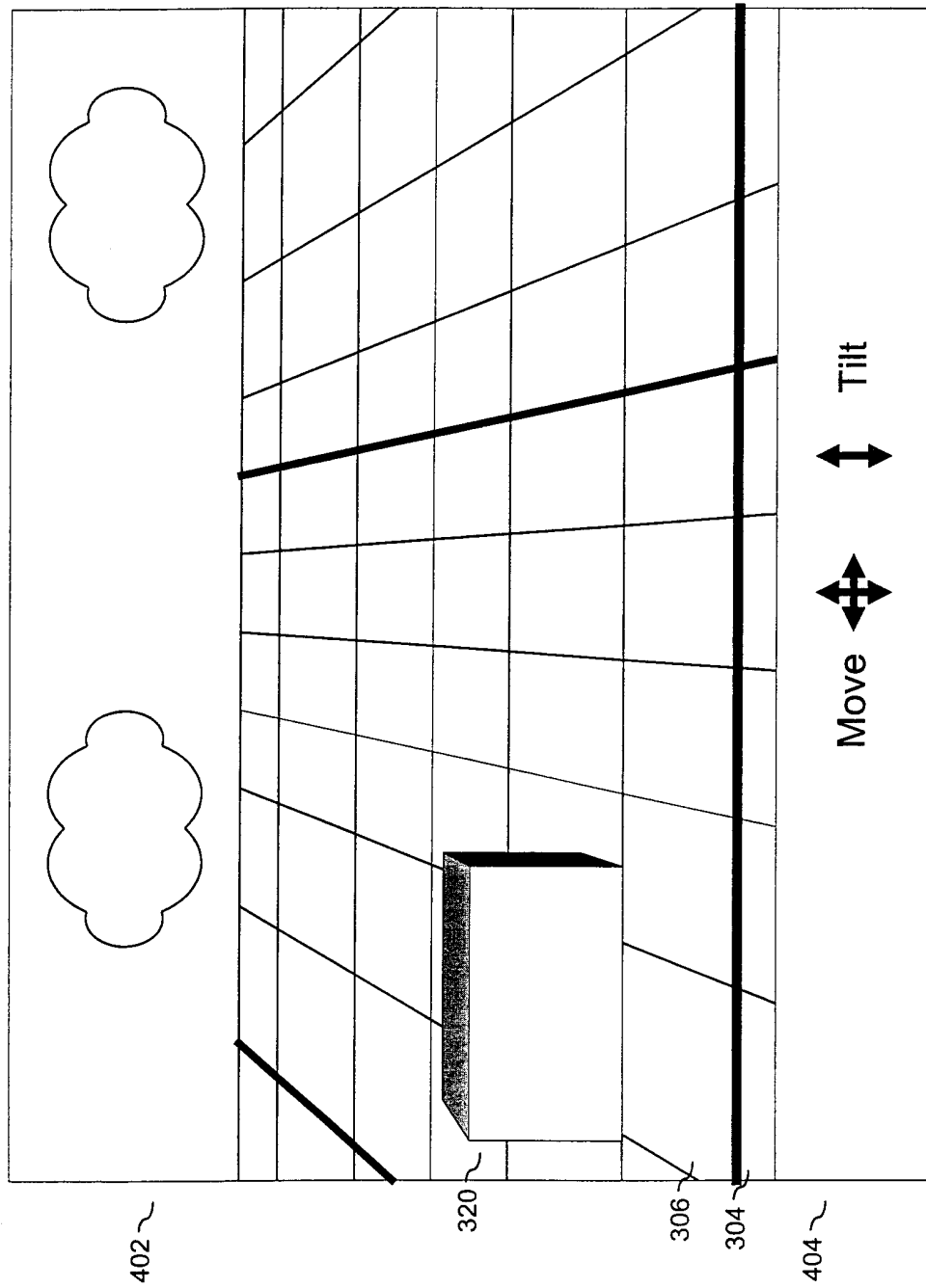


Fig. 4

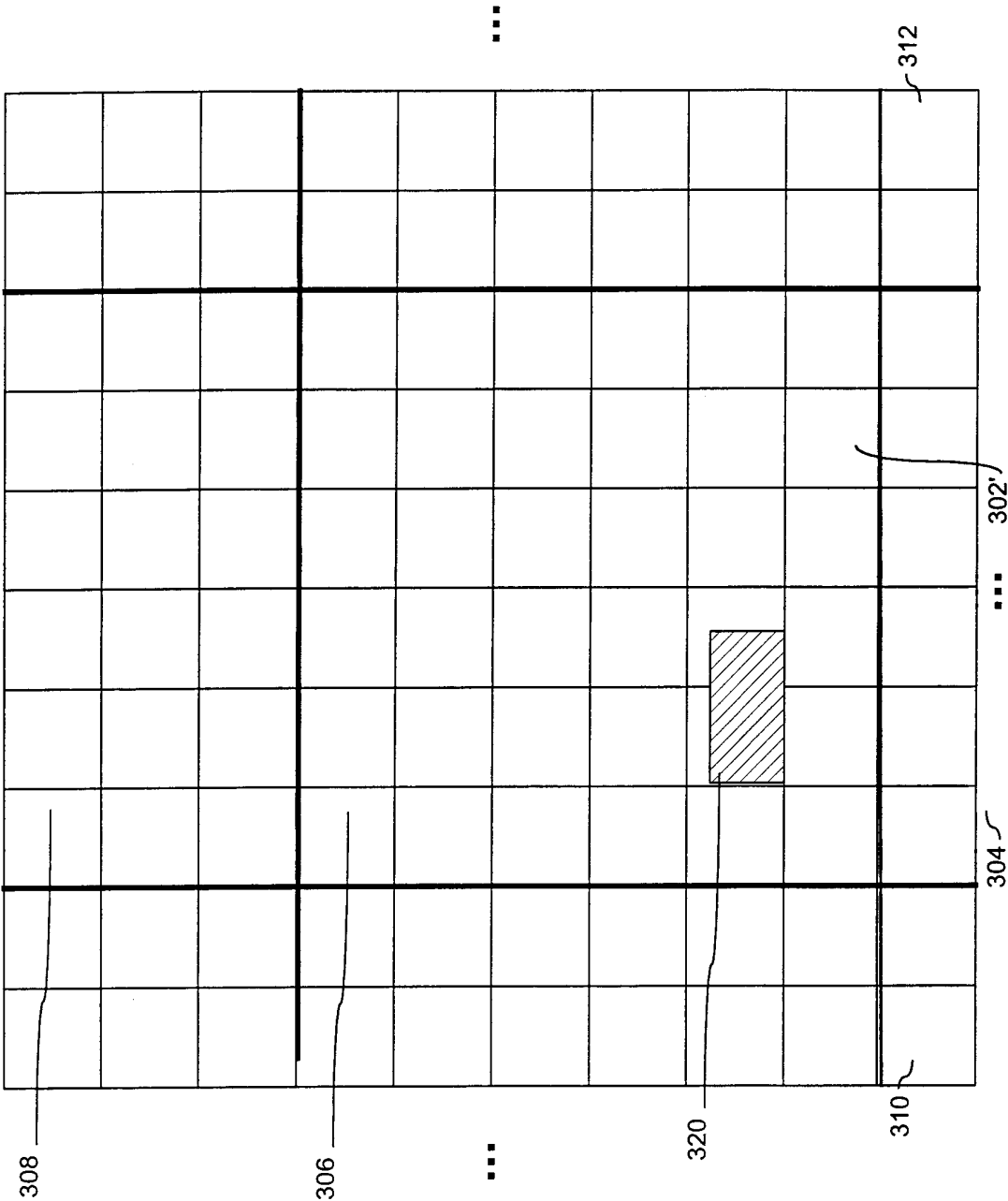


Fig. 5

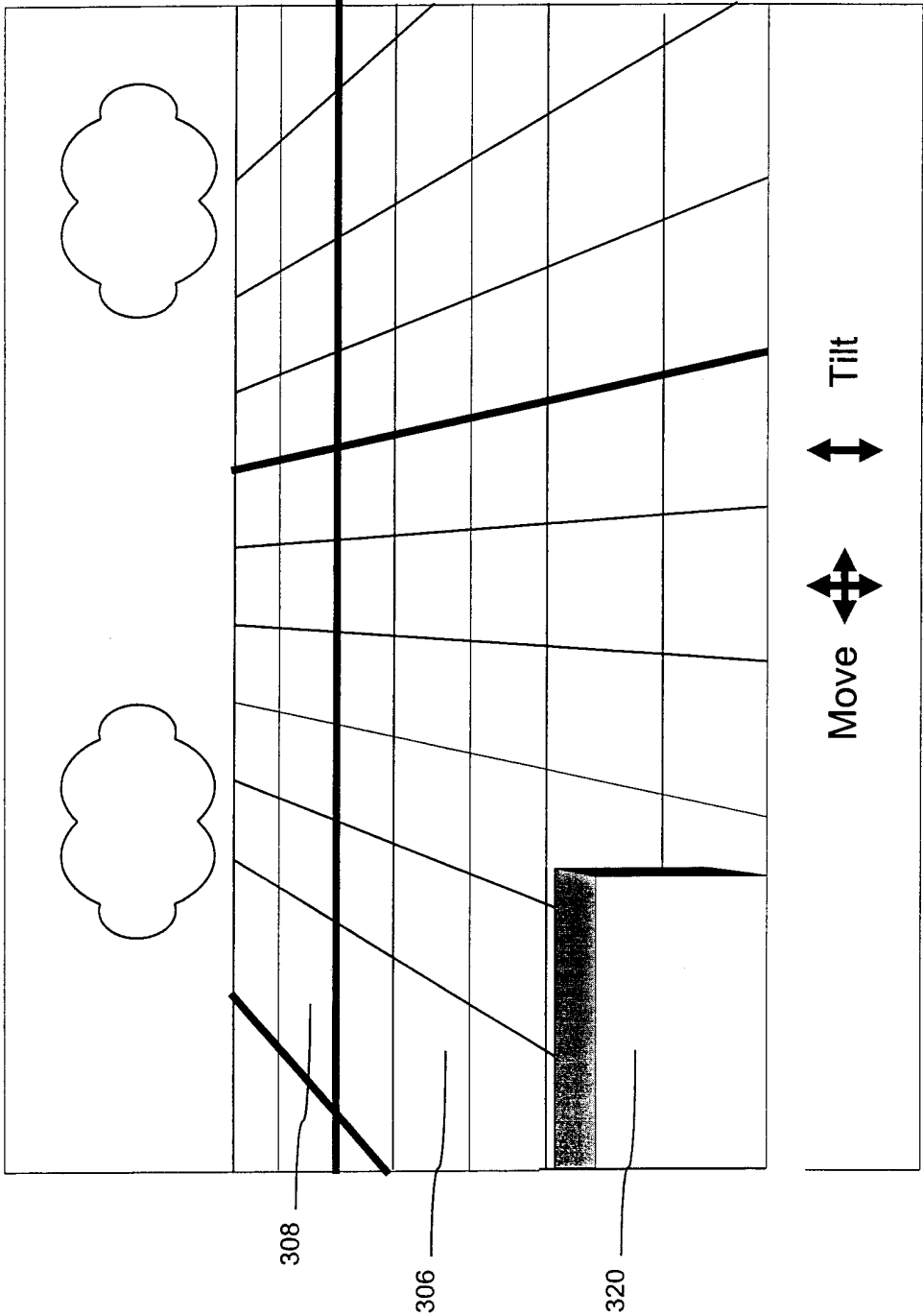
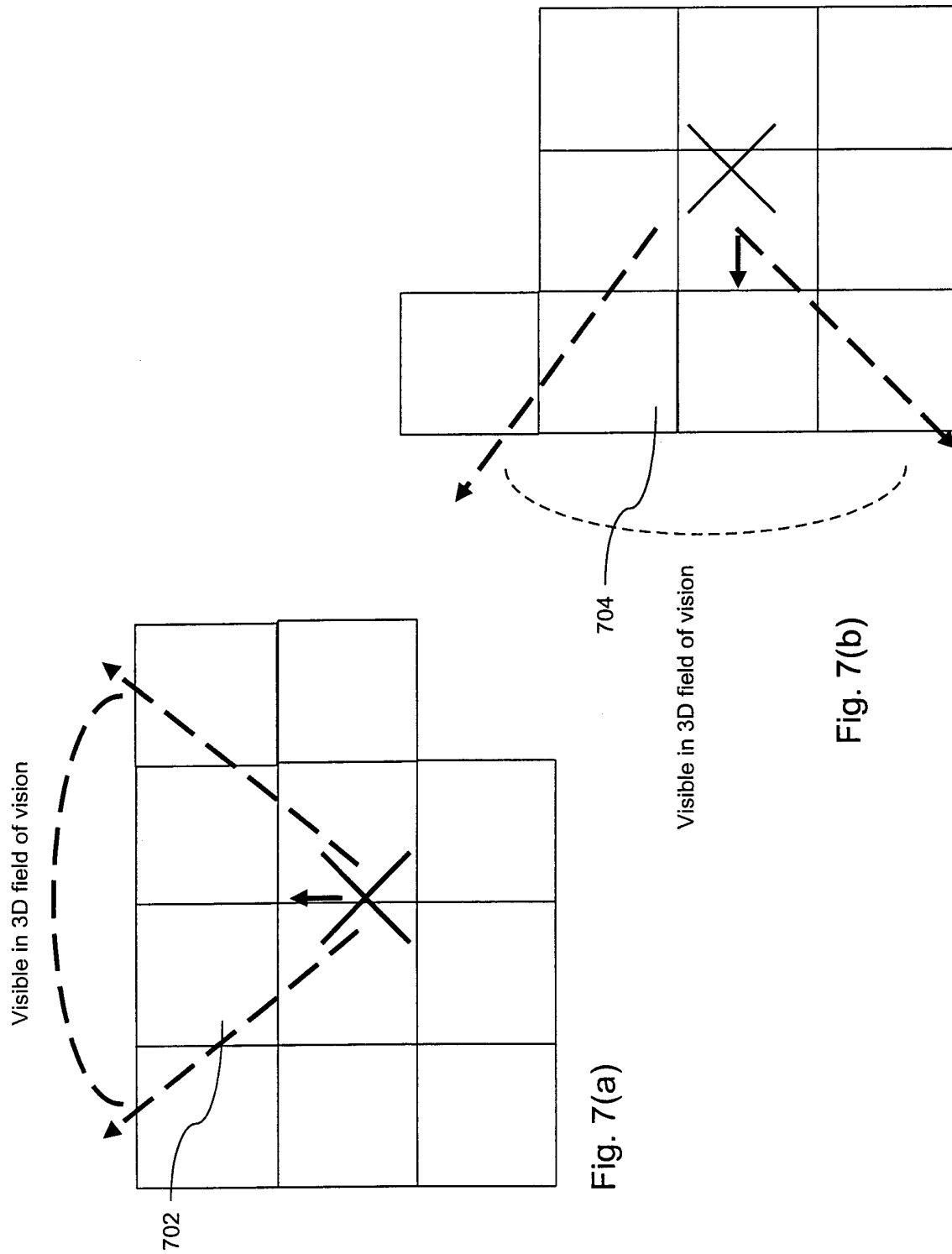


Fig. 6



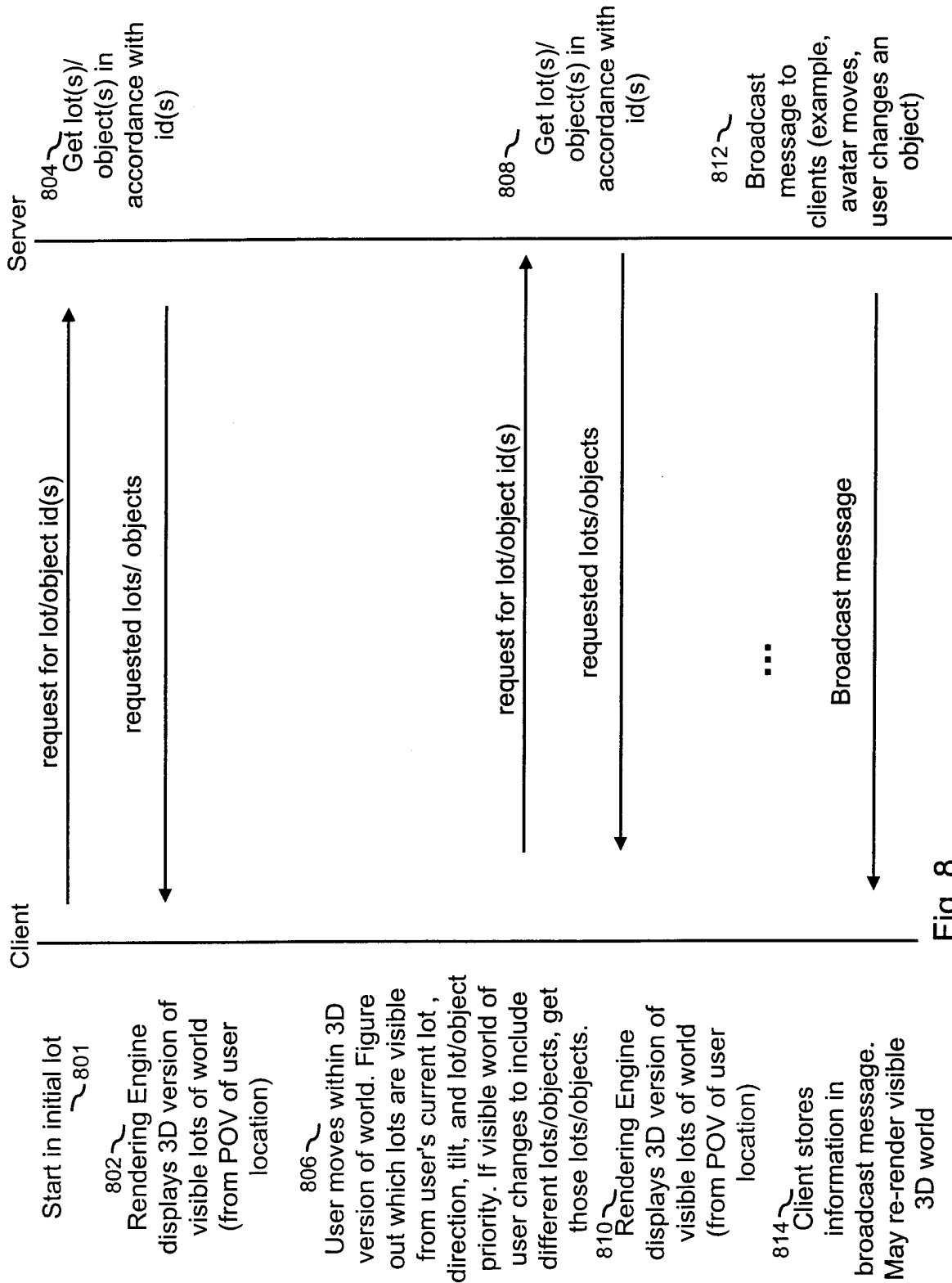


Fig. 8

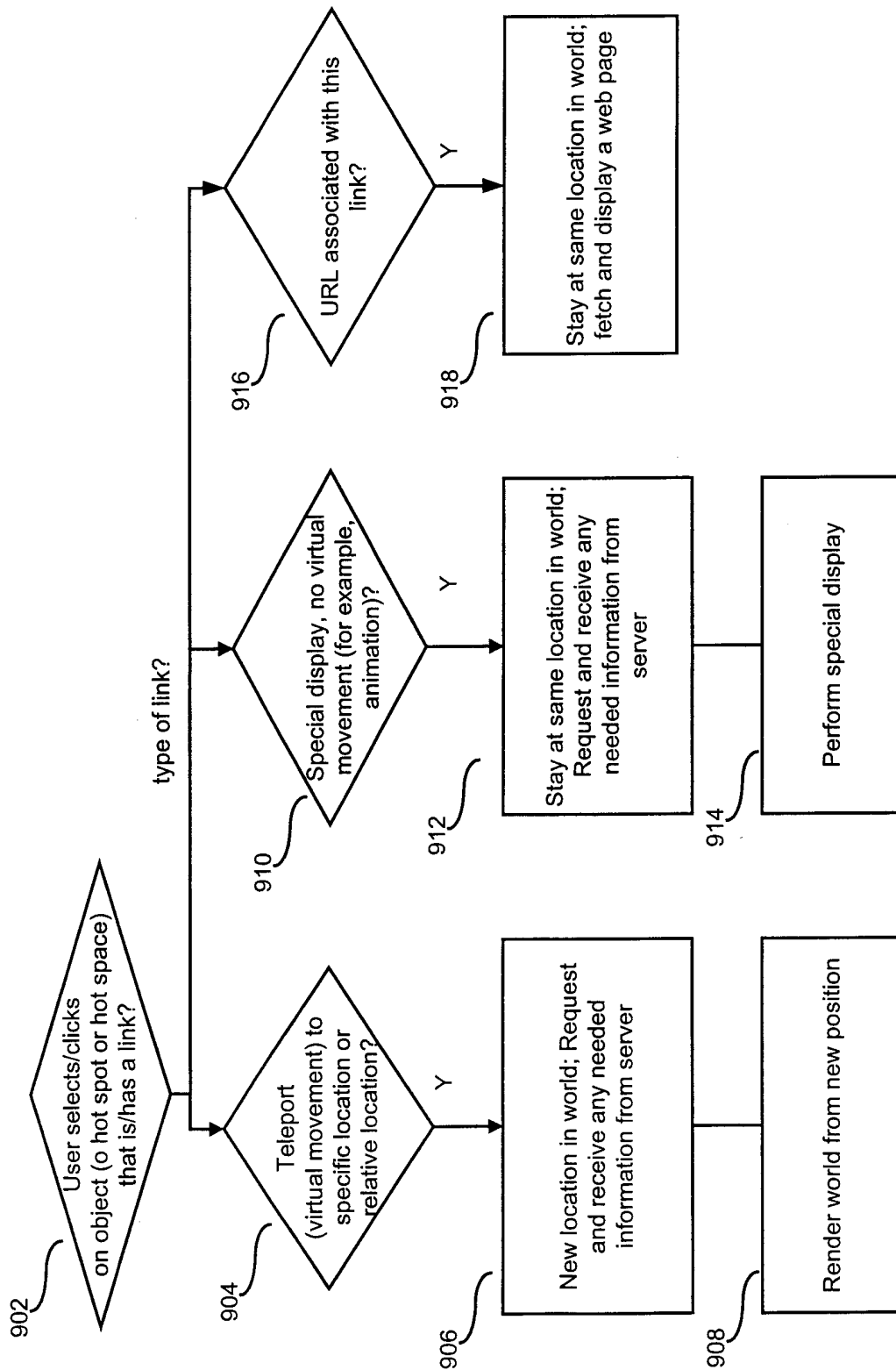


Fig. 9(a)

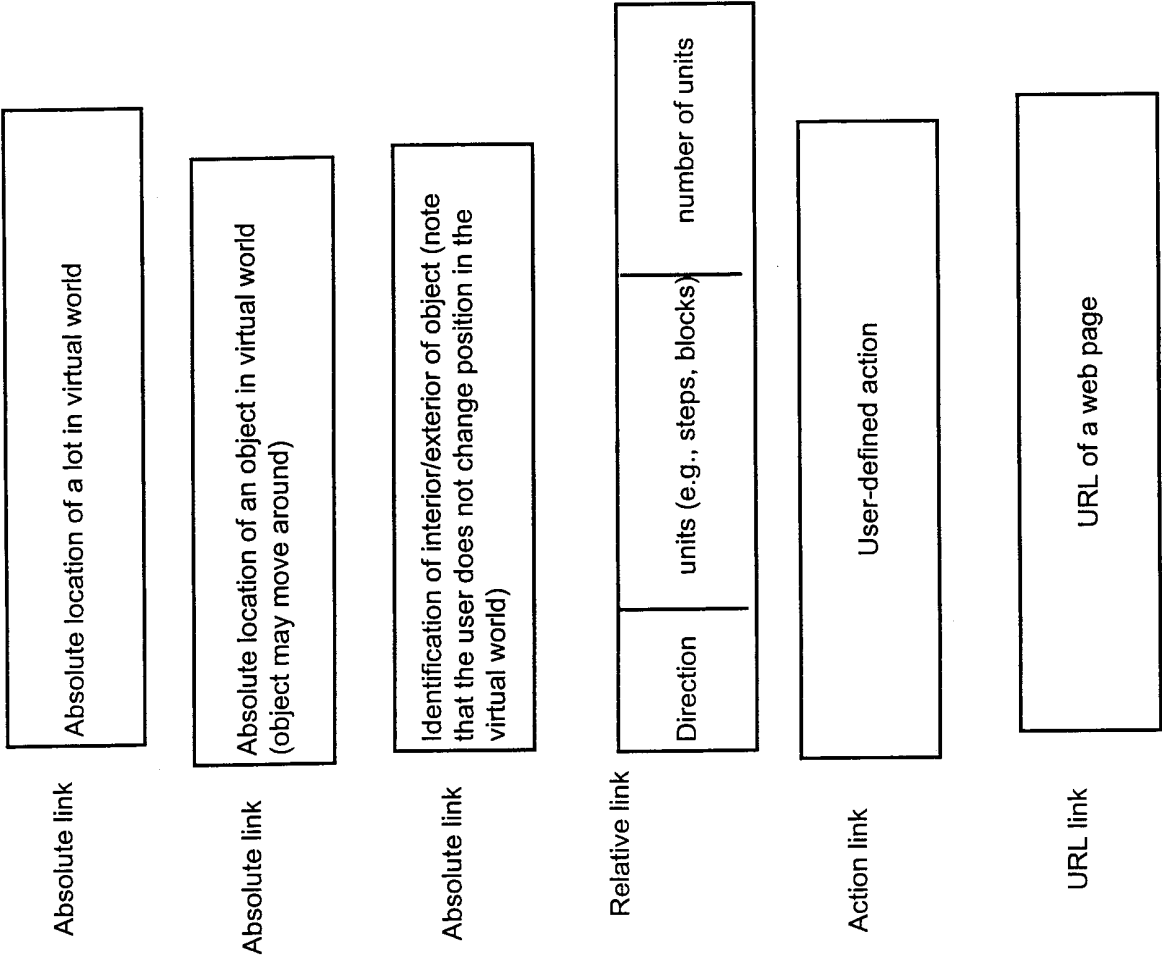


Fig. 9(b)

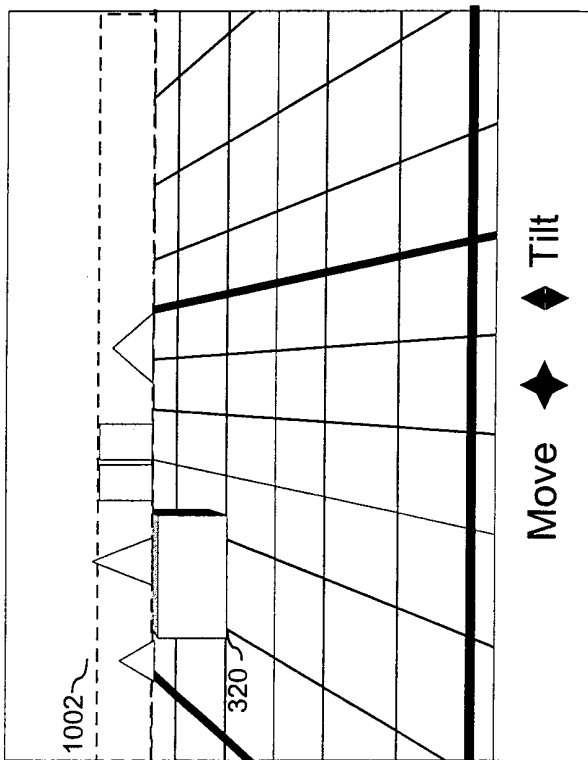


Fig. 10(a)

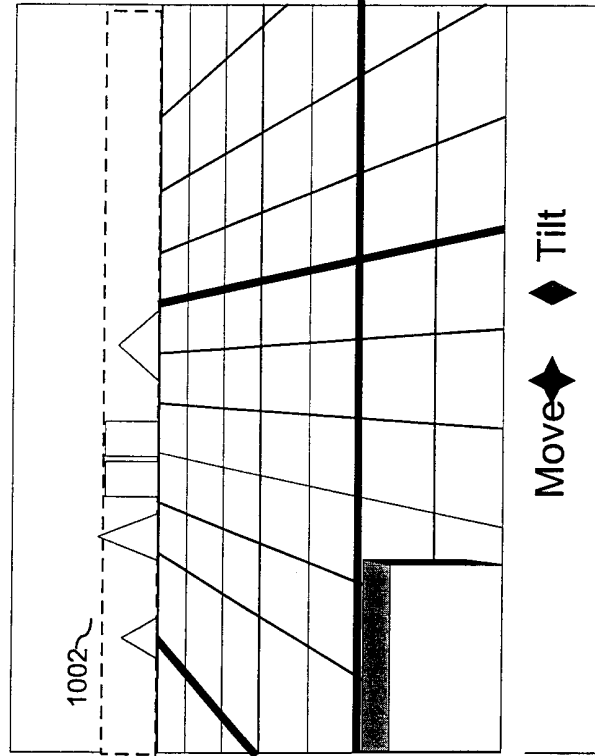


Fig. 10(b)

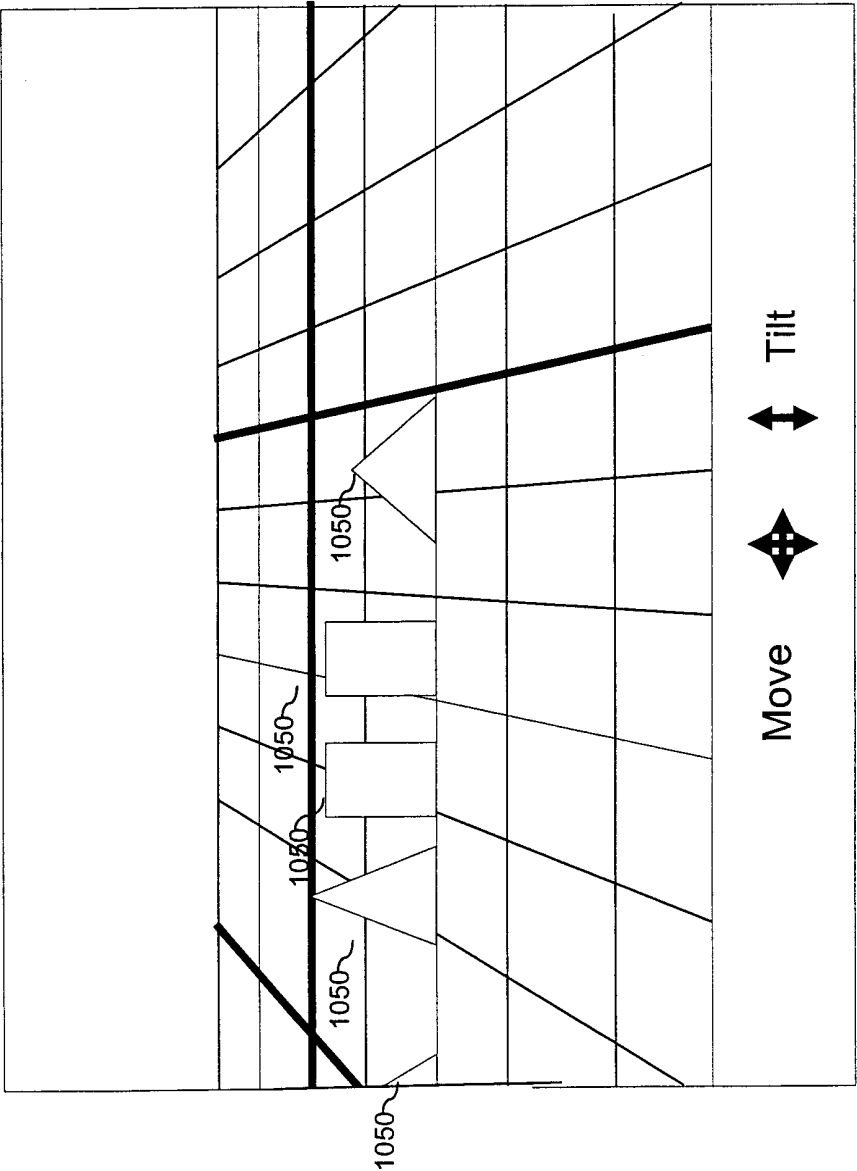


Fig. 10(c)

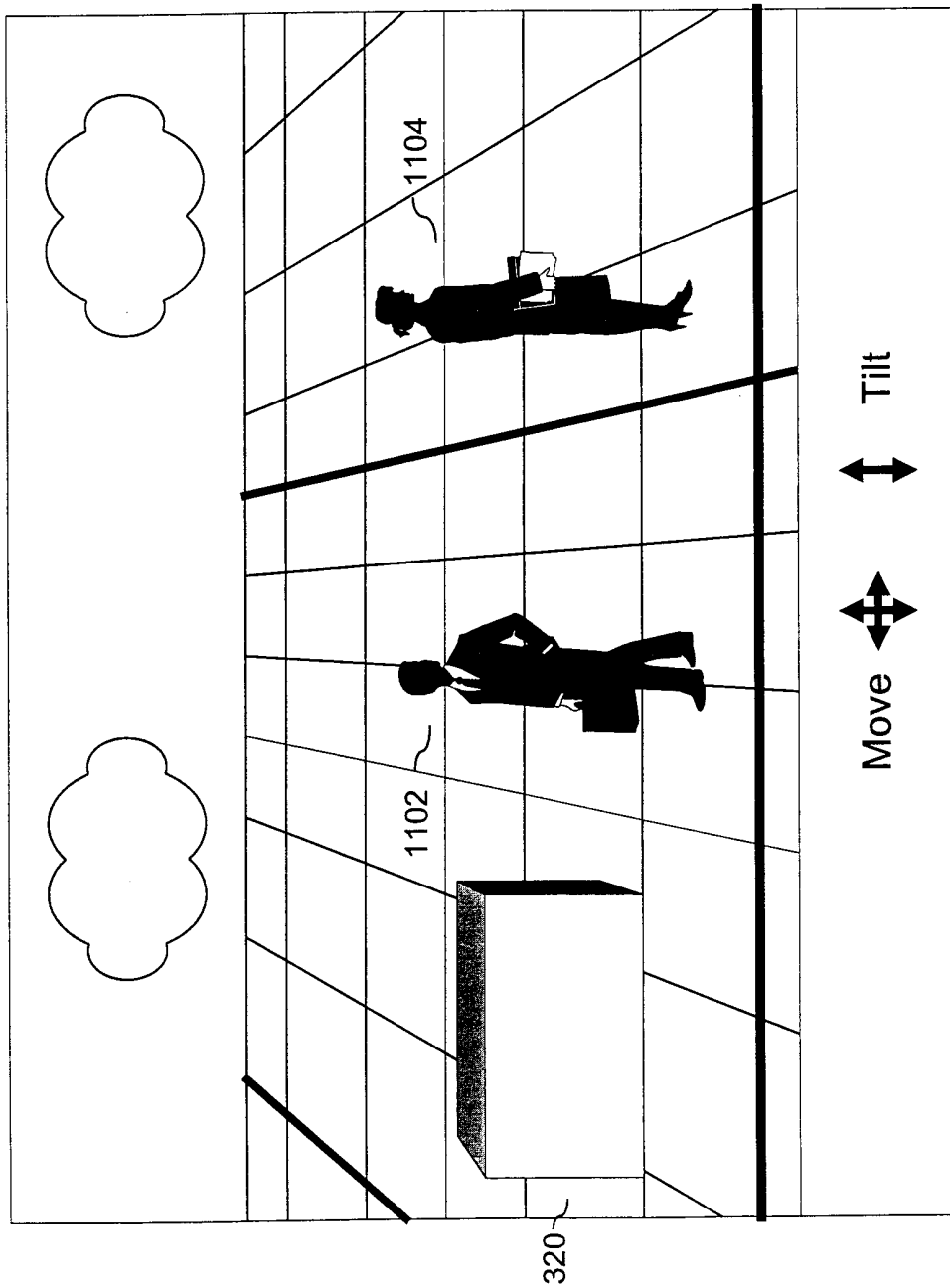


Fig. 11(a)

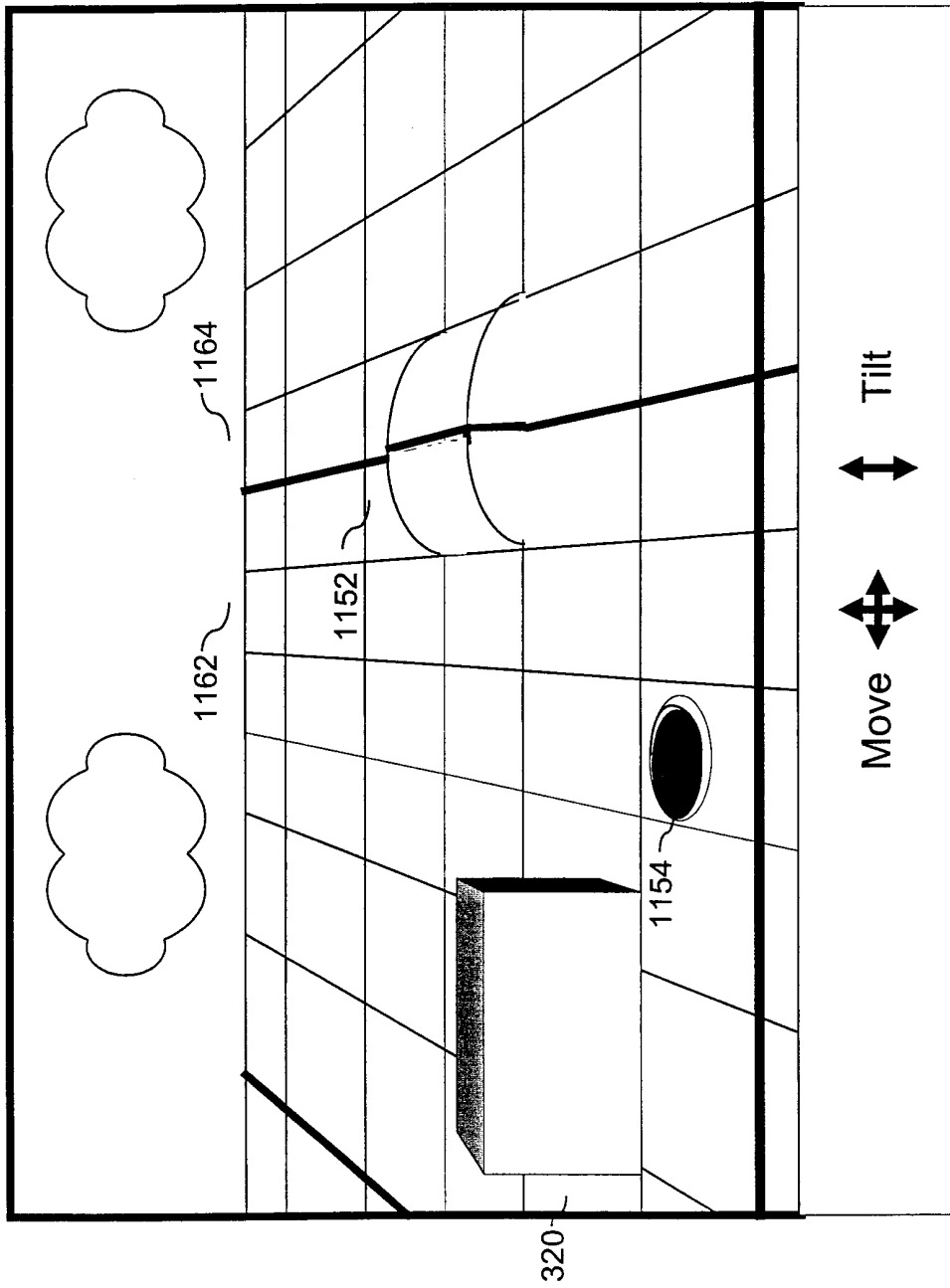


Fig. 11(b)

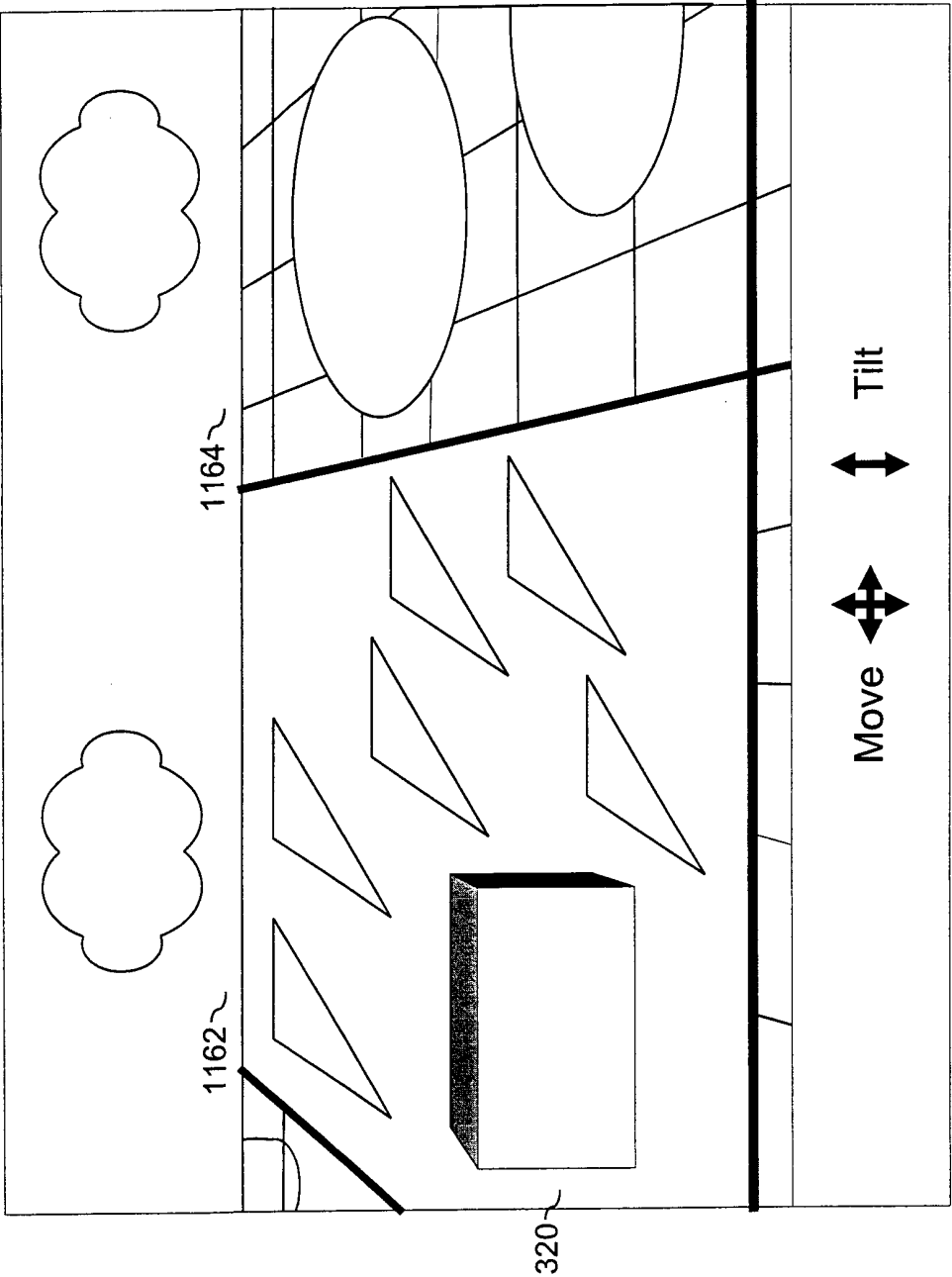


Fig. 11(c)

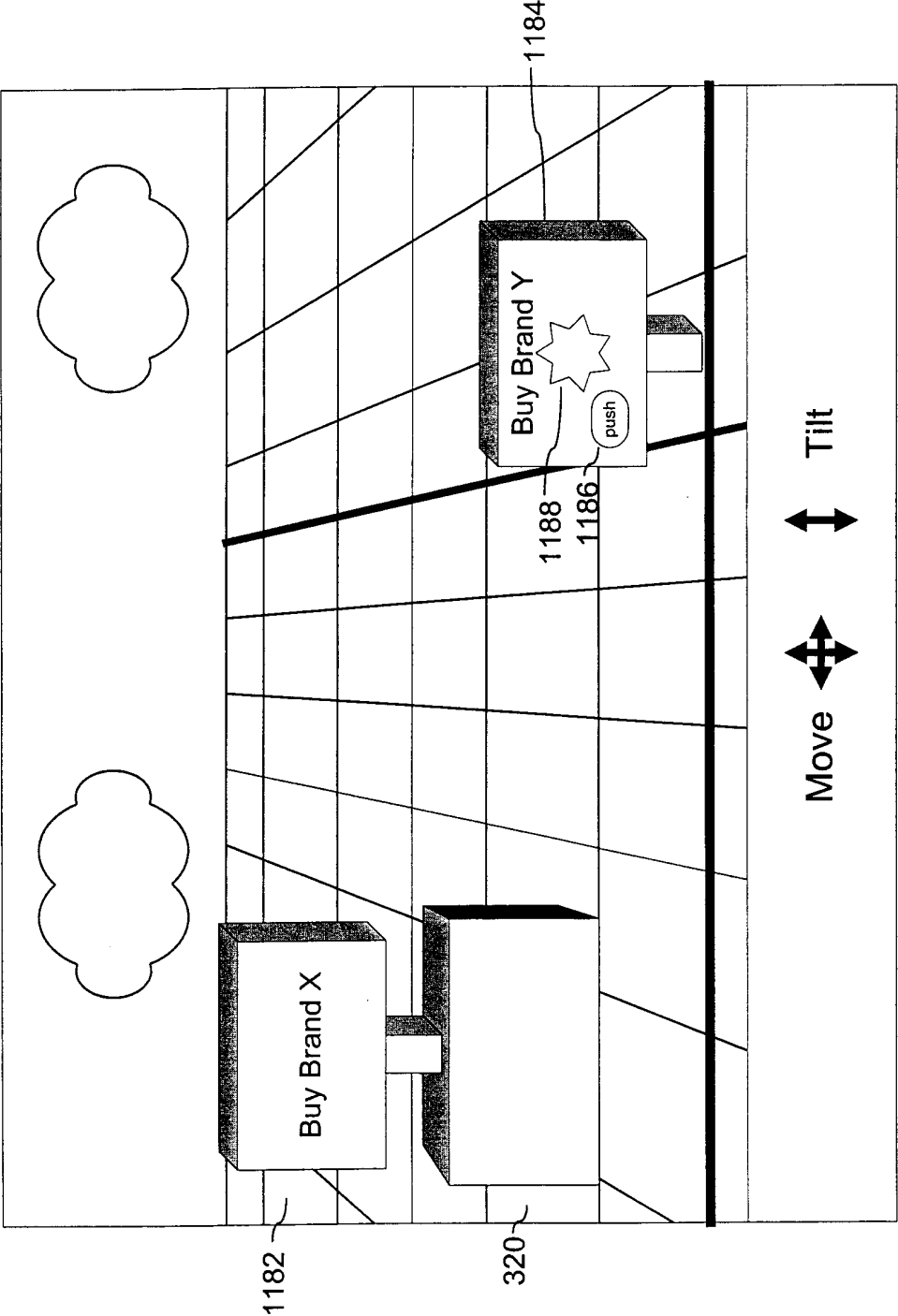


Fig. 11(d)

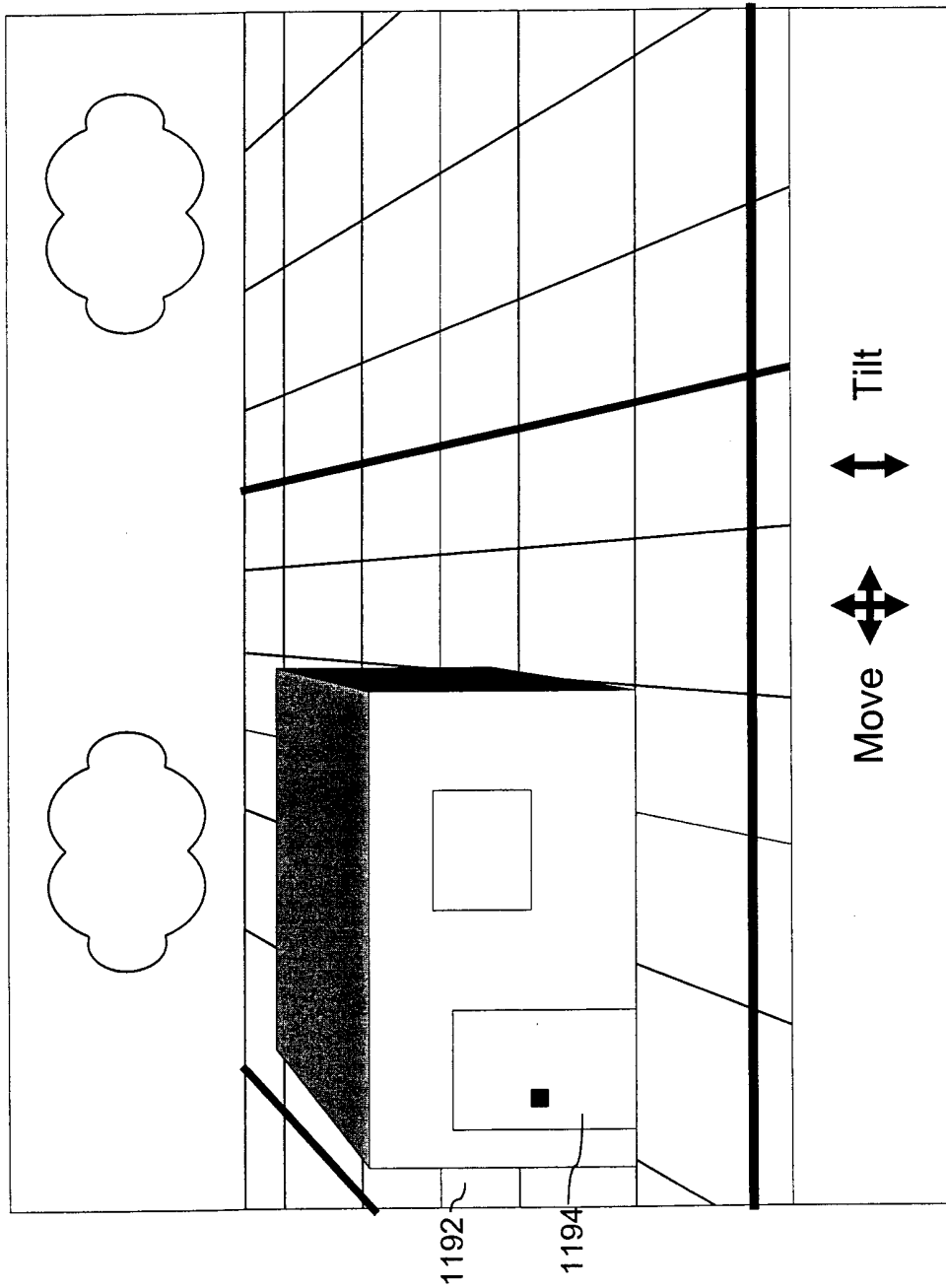


Fig. 11(e)

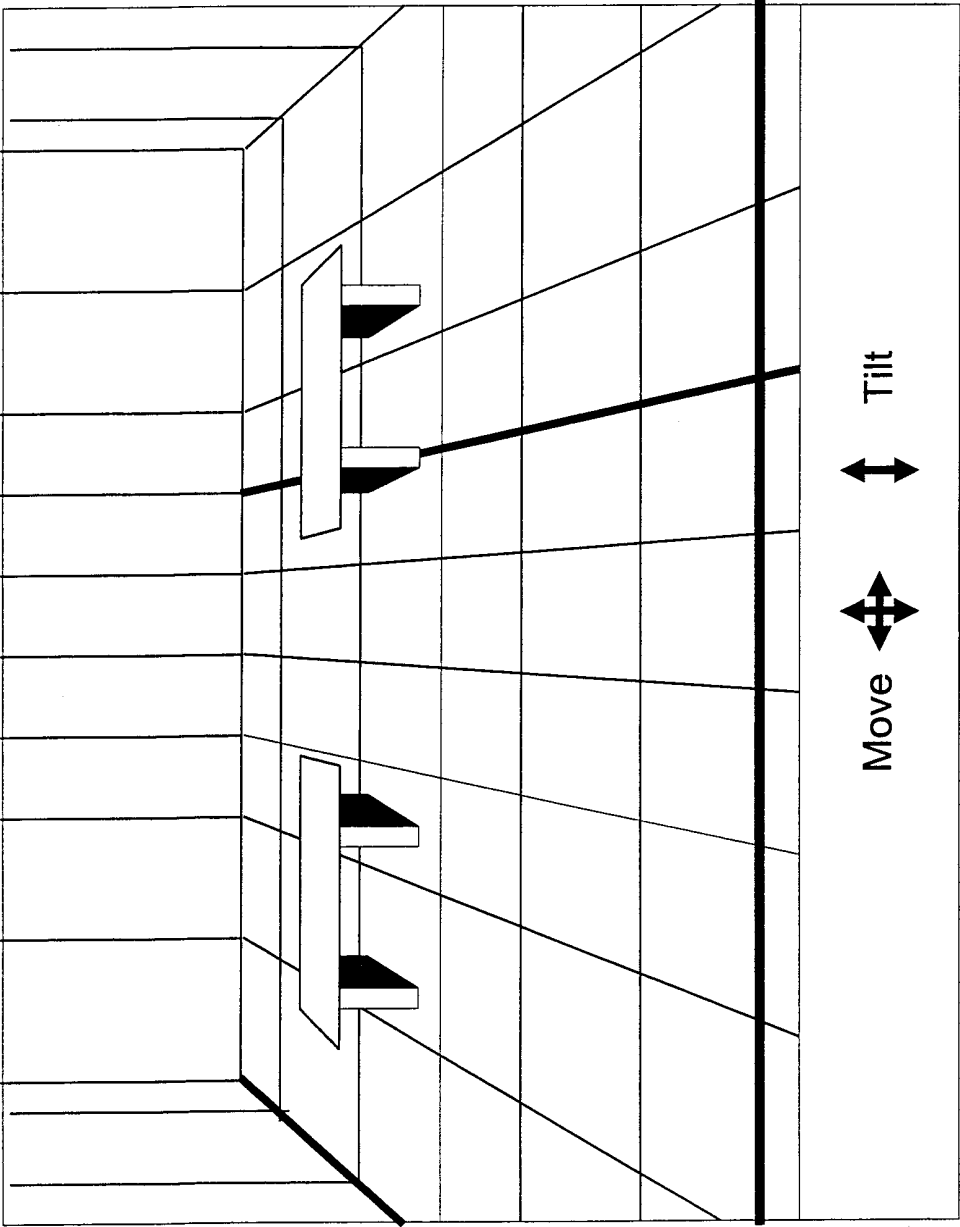


Fig. 11(f)

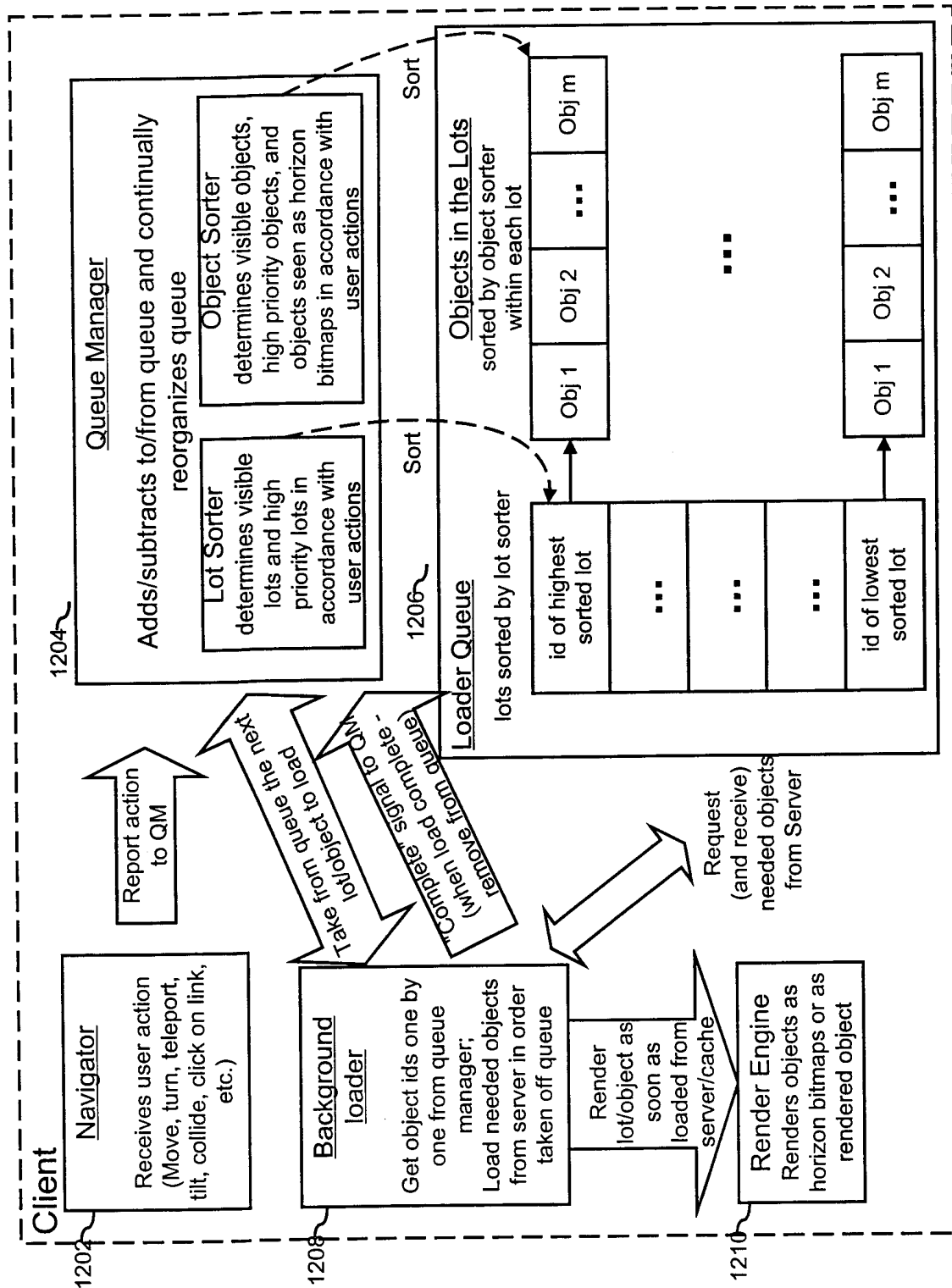


Fig. 12

Example Lot properties	
1302	Name (need not be unique)
1304	ID (128 bit coded number, unique)
1306	List of current visitors/objects/avatars/chats boxes in this lot
1308	User id (owner, initially has admin permissions)
1310	Permissions (read only, read/write, none, admin)
1312	List of 3D horizon bitmaps in this lot
1314	List of ground bitmaps in this lot; ground elevations for this lot

Example object properties	
1352	Name (need not be unique)
1354	Lot id that this object is located in (or world id that this avatar object belongs to)
1356	Type of object (for example, avatar, movable, non-movable, has an interior/exterior)
1358	List of hot spots, hot spaces in this object (links)
1360	Pointer/reference to object database (either explicitly by lot or implicitly by user id)
1362	User id (owner, initially has admin permissions)
1364	permissions(read only, read/write, none, admin)
1366	List of interactive properties associated with this object

Fig. 13(a)

Fig 13(b)

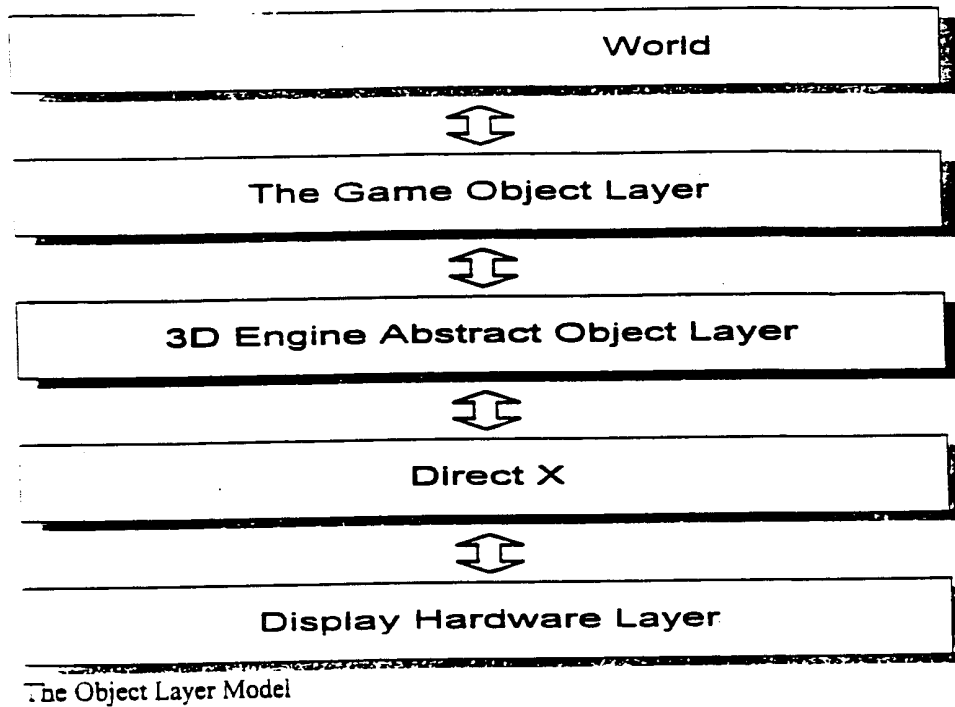


Fig. 14
Example Object Layer Model

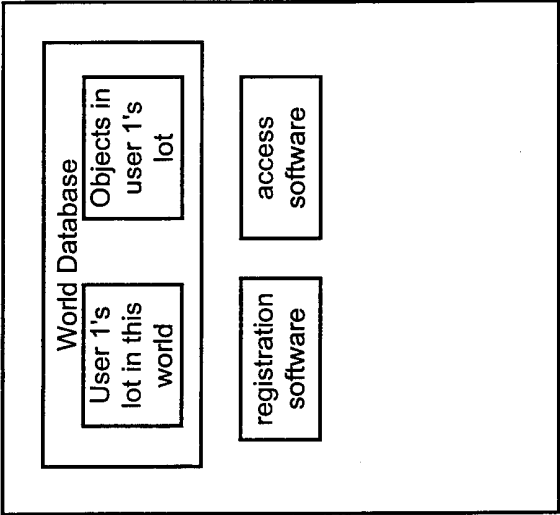


Fig.15
Single user mode

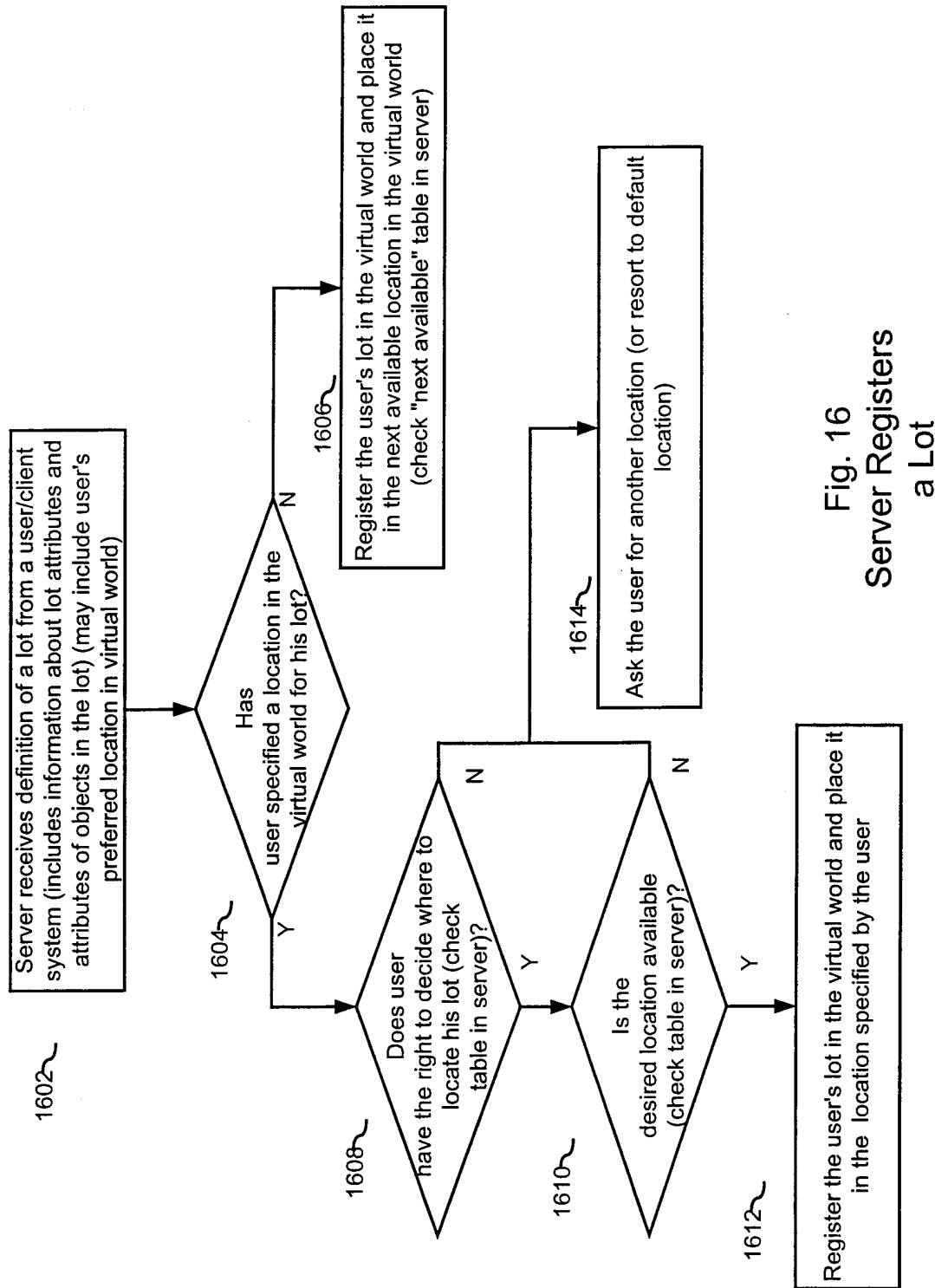


Fig. 16
Server Registers
a Lot

```

Each file for a lot includes a header and a linked list of objects.

{Beginning of File}

File Header...(of type FOREVER21HEAD)
Object1...-> (of type FOREVER21OBJ)
Object2...->
....
ObjectN
{End of File}

Here are the data types for the file:

typedef struct
{
    char szID[2];
    short nVersion;
    long INumOfObj;
    long lObjOffset;
    float fminx, fminy, fminz, // land size
    float fmaxx, fmaxy, fmaxz;
} FOREVER21HEAD;

typedef struct
{
    long lTagID;
    union
    {
        long lValue;
        long lTextOffset;
        double ddValue;
    } Value;
    double ddReserved;
} FOREVER21TAG;

typedef struct
{
    DWORD dwObjectID; // source of link
    LINKEVENT link; // source of link
    float fx, fy, fz; // init position
    float fdx, fdy, fdz; // init orientation
    float fux, fuy, fuz;
} FOREVER21LINK;

typedef struct
{
    time_t tCreateTime;
    TCHAR szFileName[64]; // name of lot file
} FILEINFO;

typedef struct
{
    TCHAR szMeshName[96]; // name of link
    TCHAR sz3DLinkFile[32]; // link (a file)
} LINKEVENT;

```

Fig. 17