



(19) **United States**
(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0179218 A1**
Martins et al. (43) **Pub. Date: Sep. 25, 2003**

(54) **AUGMENTED REALITY SYSTEM**

Publication Classification

(76) Inventors: **Fernando C. M. Martins**, Hillsboro, OR (US); **Stephen Junkins**, Bend, OR (US); **Jason E. Plumb**, Portland, OR (US)

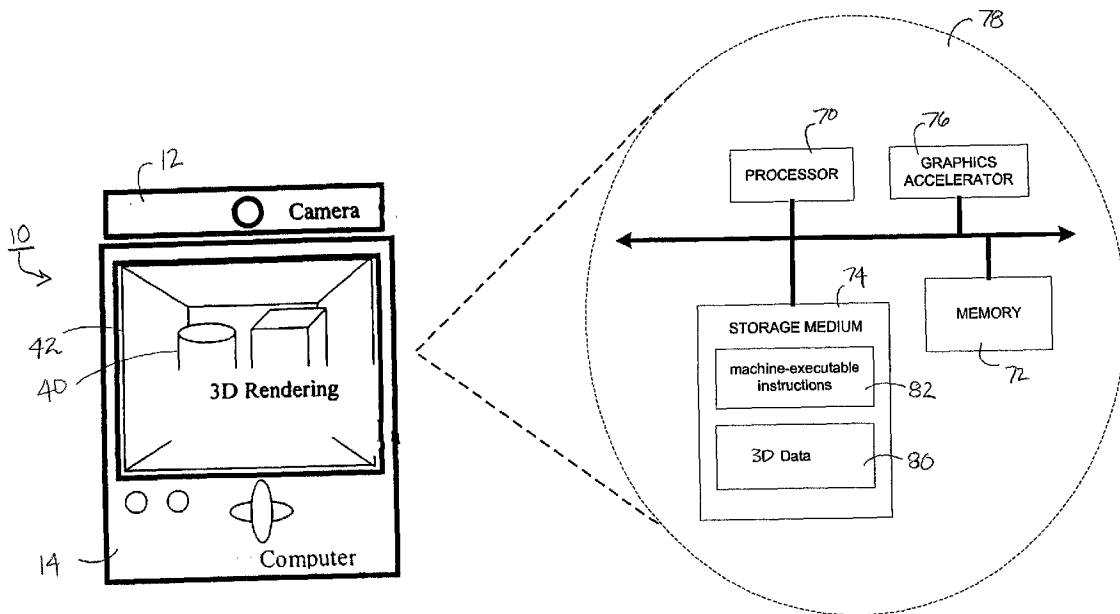
(51) **Int. Cl.⁷** **G09G 5/00**
(52) **U.S. Cl.** **345/633**

Correspondence Address:
FISH & RICHARDSON, PC
4350 LA JOLLA VILLAGE DRIVE
SUITE 500
SAN DIEGO, CA 92122 (US)

(57) **ABSTRACT**

A virtual reality system surveys a real-world environment, generates 3D data that defines the real-world environment, renders a virtual 3D environment using the 3D data, retrieves a virtual object from a database comprised of pre-stored virtual objects, renders the virtual object in the virtual 3D environment, and re-positions the virtual object in the virtual 3D environment.

(21) Appl. No.: **10/104,471**
(22) Filed: **Mar. 22, 2002**



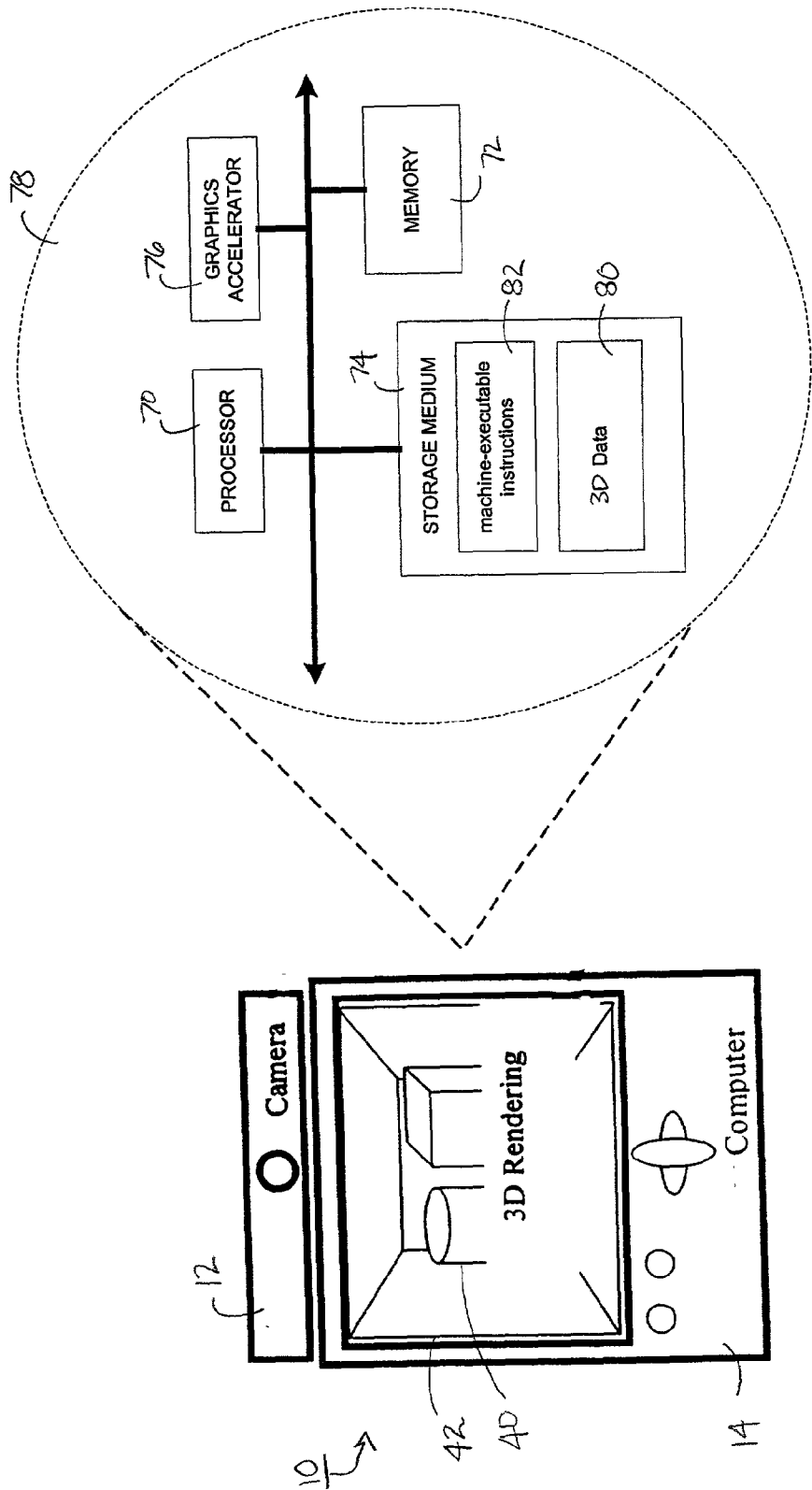


FIG. 1

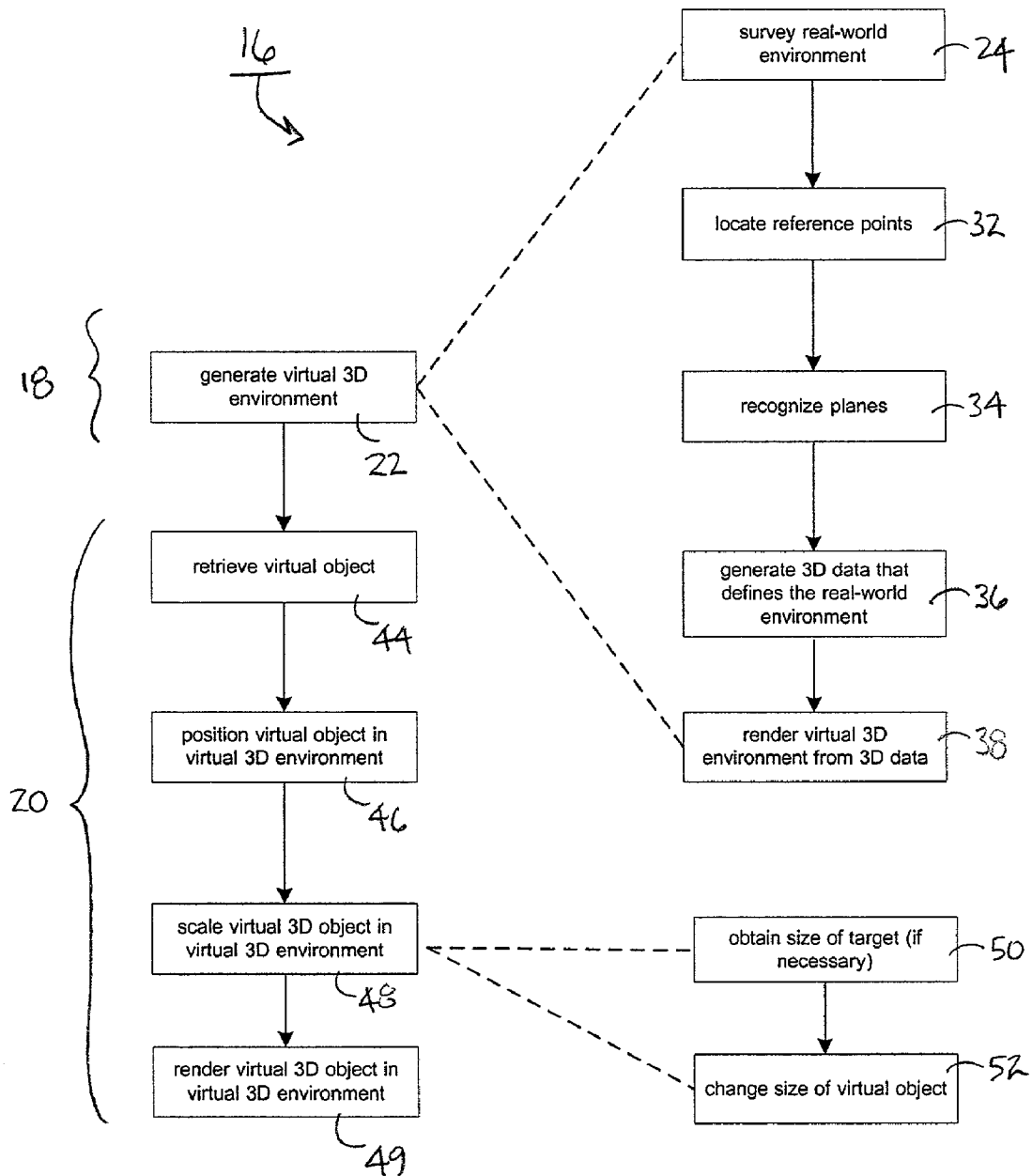


FIG. 2

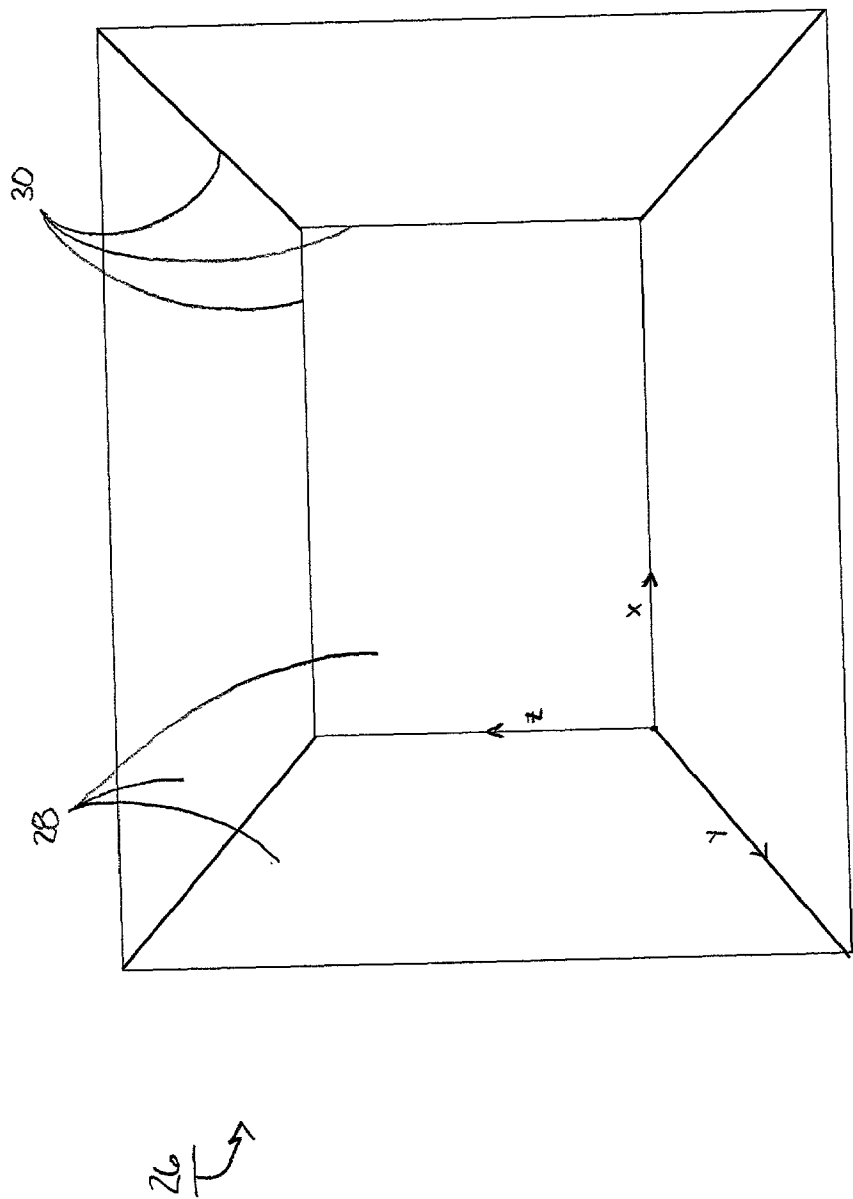


FIG. 3

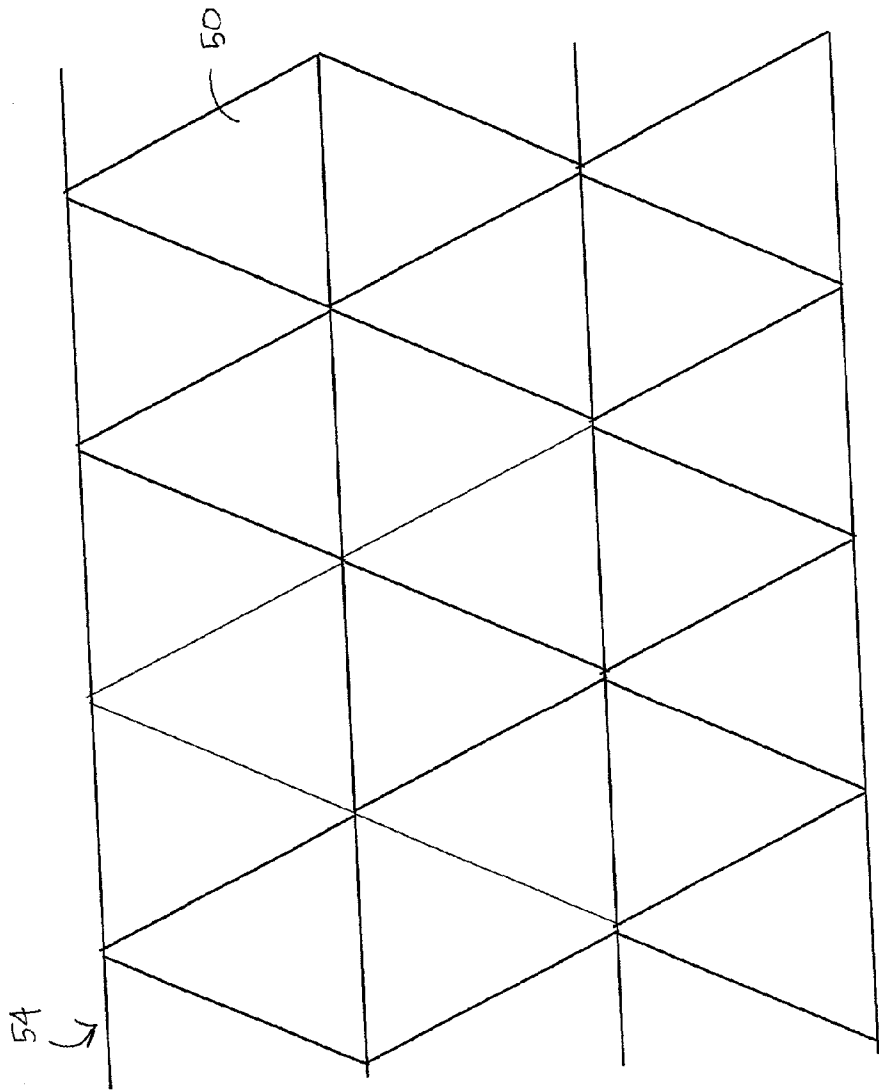


FIG. 4

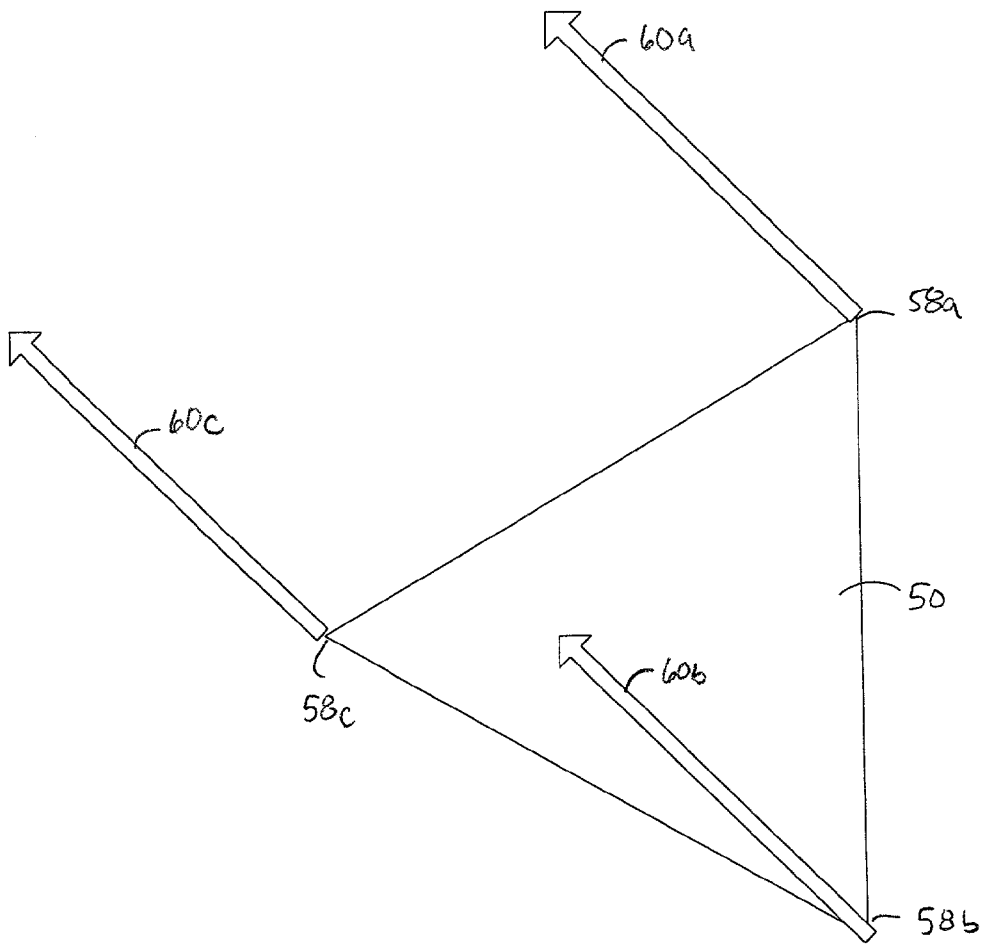


FIG. 5

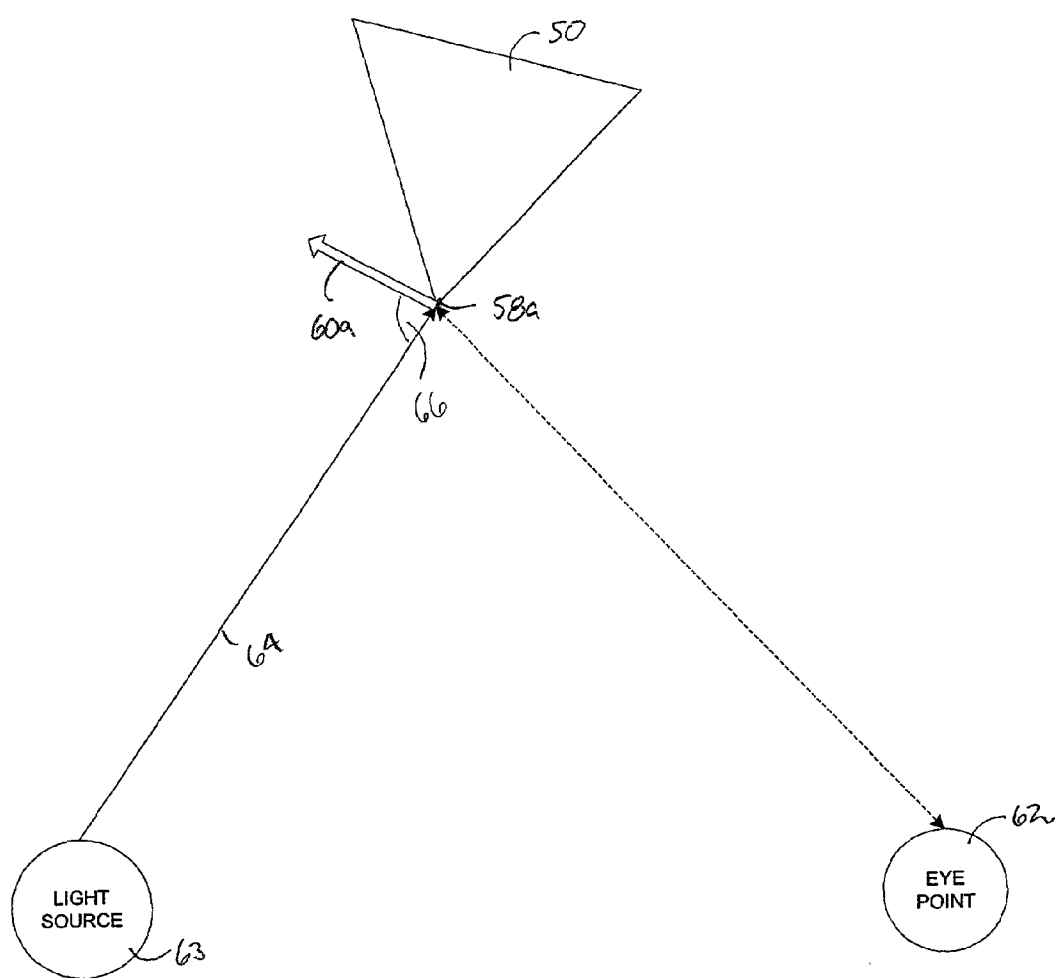


FIG. 6

AUGMENTED REALITY SYSTEM

TECHNICAL FIELD

[0001] This application relates to a virtual reality system that generates a virtual three-dimensional (3D) environment from a real-world environment and renders a virtual 3D object in the virtual 3D environment.

BACKGROUND

[0002] 3D graphics may be used to implement a virtual reality system. Conventional virtual reality systems “inject” a user into a fictitious virtual 3D environment. In that environment, the user can interact with objects, characters, and the like as if in the real world.

DESCRIPTION OF THE DRAWINGS

[0003] FIG. 1 is a block diagram of a virtual reality system.

[0004] FIG. 2 is a flowchart of a process performed using the virtual reality system.

[0005] FIG. 3 is a perspective view of a room in the real world.

[0006] FIG. 4 is a block diagram of polygons that make up a virtual 3D environment.

[0007] FIG. 5 is a close-up view of one of the polygons.

[0008] FIG. 6 is a block diagram showing how the polygon of FIG. 5 is illuminated in a virtual 3D environment.

DESCRIPTION

[0009] FIG. 1 shows a virtual reality system 10. Virtual reality system 10 includes a camera 12 and a computer 14. Camera 12 may be a digital video camera that is capable of capturing frames of video data. Computer 14 may be a handheld computer, such as a Palm Pilot®, that is capable of rendering a virtual 3D environment from the captured video. Camera 12 and computer 14 are tethered to each other in this embodiment; however, they may be untethered.

[0010] FIG. 2 shows a process 16, performed by computer 14, for generating a virtual 3D environment from a real-world environment and placing virtual objects in that virtual 3D environment. Unlike conventional virtual reality systems that place a user into a fictitious environment, the augmented reality system of process 16 generates a 3D version of a real-world environment and places virtual objects into the 3D version of the real-world environment.

[0011] By way of example, a user may generate a virtual 3D version of a real-world living room. The user may then furnish the resulting virtual living room with virtual 3D objects, such as furniture, artwork, and the like. The virtual objects may be rearranged, as desired, in order to obtain a pleasing layout of the room. Thus, the virtual reality system described herein provides a 3D preview of a real-world space augmented with computer-generated virtual elements, hence the name “augmented reality”. The virtual reality system has other applications as well.

[0012] Referring to FIG. 2, process 16 includes a pre-processing stage 18 and a run-time stage 20. Pre-processing stage 18 may be performed only once for each real-world environment that is to be modeled. Run-time stage 20 may

be performed each time a virtual object is to be placed into a virtual 3D model of the real-world environment.

[0013] Process 16 generates (22) a virtual 3D environment. To generate the virtual 3D environment, process 16 surveys (24) a real-world environment, such as a room. The user walks camera 12 around the room, capturing frames of two-dimensional (2D) video data from different positions in the room. Process 16 uses these frames to generate a 3D model of the room.

[0014] To this end, process 16 extracts features of the room (the real-world environment, in this example) from the frames of 2D video data. These features include planes and reference points, such as corners, in the real-world environment. FIG. 3 shows a room 26 that contains planes 28 and corners 30.

[0015] Process 16 locates (32) corners 30 in room 26 using standard corner detection processes and performs standard position (“pose”) estimation processes to determine the location of camera 12 in the room. Process 16 tracks the motion of the corners over a predetermined time frame (which corresponds to a given number of frames of 2D video). It is noted that the corners themselves do not move within the room, but rather they move relative to camera 12 exclusively due to the motion of camera 12. It is this relative motion that is being tracked. Based on the locations of the corners and their movement over time, process 16 determines the position of camera 12 relative to the corners for each frame of video.

[0016] The camera position is used when constructing a virtual 3D version of the room. That is, knowing the camera position, allows process 16 to know the perspective from which each frame was taken. Knowing the perspective of each frame allows process 16 to determine where in the virtual 3D environment the additional virtual elements should be positioned.

[0017] Process 16 recognizes (34) planes in the real-world environment that are bounded by the corners. Process 16 recognizes planes by identifying clusters of three or more points (e.g., pixels) of the 2D video that behave similarly during motion of camera 12. For example, as camera 12 moves toward a cluster of pixels, the pixels may appear to “grow”, i.e., they may appear larger because they become closer to the camera. Pixels on the same plane may appear to “grow” by about the same amount. Examples of planes that may be identified include a floor, ceiling, and walls of a room.

[0018] Once process 16 identifies the corners and planes from the 2D frames of video, process 16 generates (36) 3D data that defines the corners and the planes of the 3D environment relative to the camera positions. The 3D data may define Cartesian XYZ coordinates of pixels that make up the corners and planes of the virtual 3D environment. Other types of 3D data may alternatively be used.

[0019] Process 16 renders (38) the virtual 3D environment (e.g., virtual living room) from the 3D data. The virtual 3D environment 40 is rendered on the display screen 42 of computer 14. A user can then populate this virtual 3D environment with virtual objects retrieved by computer 14.

[0020] In more detail, the user selects a virtual 3D object from a database, along with a location in the virtual 3D

environment where the selected virtual 3D object is to be displayed. The selections may be made using a light pen, stylus on a touch screen, or any other type of computer interface. As noted above, the virtual 3D objects may be virtual 3D representations of furniture or the like. Process 16 retrieves (44) the selected virtual 3D object from the database and positions (46) it at the appropriate location. Positioning is performed in response to user input.

[0021] In order to achieve a realistic effect, process 16 may scale (48) the selected virtual 3D object (i.e., model) before rendering. In this context, scaling may include changing the size of the virtual 3D object so that the virtual 3D object is appropriate given the size of the virtual 3D environment.

[0022] Process 16 scales the virtual 3D object by obtaining (50) the size of a target object in the real-world environment and changing (52) the size of the virtual 3D object in accordance with the size of the target. For example, the size of an object (e.g., the height of a ceiling, distance between two objects, etc.) in the 3D environment may be captured beforehand. Using the size of the target as a reference, process 16 may change the size of the virtual 3D object so that its size is smaller or larger to correlate substantially to the size of the target. Process 16 then renders (49) the virtual objects in the virtual environment. By way of example, process 16 may retrieve a virtual 3D model for a table from a database. Data for the table may include its dimensions, such as length, width and height. Knowing these dimensions and the size of the target, process 16 can scale the table to its appropriate size within the virtual 3D environment and then render the virtual table.

[0023] Process 16 continuously tracks the position of the camera during movement throughout the real-world 3D environment and updates the position of the camera periodically in order to ensure that virtual objects are placed at correct locations within the virtual 3D environment. That is, process 16 uses the position of the camera to further refine (and render) the definition of the virtual 3D environment and to place the virtual objects within the virtual 3D environment.

[0024] Process 16 may also illuminate the virtual 3D object in the virtual 3D environment to simulate lighting from one or more light sources in the real-world environment. This may be done using well-known processes, such as that described in "Adaptive Estimation Of Illumination Distribution With Unknown Reflectance Properties In Shadow Regions", by Sato, I, Sato, Y., and Ikeuchi, K., The Proceedings of the Seventh International Institute of Electrical and Electronics Engineers (IEEE) Conference, Vol. 2, pgs. 875-882 (1999).

[0025] Referring to FIG. 4, 3D data for a virtual 3D environment defines interconnecting polygons 54. Polygons 54 are triangles in this embodiment; however, other types of polygons may be used to construct the 3D environment.

[0026] Referring to FIG. 5, the 3D data for a polygon 50 is comprised of coordinates for three vertices 58a, 58b and 58c positioned in Cartesian XYZ (or other) space. A unit normal vector ("normal") 60a, 60b and 60c at each respective vertex 58a, 58b and 58c affects how the vertex is perceived relative to a predefined reference point (the "eye-point") 62 (FIG. 6) in the virtual 3D environment. Taking

vertex 58a as an example in FIG. 6, normal 60a determines the amount of light that reaches vertex 58a from a predefined light source 63 in the virtual world. The amount of light is determined using the dot product of unit normal 60a and a unit vector 64 from the light source. The dot product value defines the cosine of angle 66 between the light and the normal. The shading applied to each polygon face is determined based on this angle, as described below. Coordinates for the normals may be stored with the 3D data for each vertex.

[0027] Process 16 can change the illumination of the objects by altering the positions of virtual light sources in the virtual 3D environment and/or adding virtual light sources. Process 16 can also affect how the lighting hits the virtual objects by changing the positions of normal vectors on the virtual 3D object. Thus, process 16 can simulate light hitting an object from both an inside light source, such as a lamp, and/or outside light, such as a window. This provides for a more realistic overall effect in the resulting simulation. Also, the colors of the various light sources may be varied.

[0028] Process 16 also permits a user to re-position virtual 3D objects in the virtual 3D environment. For example, a user may drag and drop a virtual 3D object from one location in the virtual 3D environment to another location. This allows the user to experiment with several different layouts.

[0029] As shown in FIG. 1, computer 14 may include a processor 70, a memory 72, a storage medium 74 (e.g., a computer hard disk), and a 3D graphics accelerator 76 for rendering the virtual 3D environment and processing 3D data (see view 78). Storage medium 74 stores 3D data 80 that defines the virtual 3D environment, and machine-executable instructions 82, which are executed by processor 70 out of memory 72 to perform process 16.

[0030] Process 16, however, is not limited to use with the hardware and software of FIG. 1; it may find applicability in any computing or processing environment. Process 16 may be implemented in hardware, software, or a combination of the two. Process 16 may be implemented in computer programs executing on programmable computers that each includes a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code may be applied to data entered using an input device, such as a mouse or a keyboard, to perform process 16 and to generate output information.

[0031] Each such program may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language may be a compiled or an interpreted language.

[0032] Each computer program may be implemented as a computer program stored on a storage medium (e.g., CD-ROM, hard disk, or magnetic diskette) that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium is read by the computer to perform process 16. Process 16 may also be implemented as an article of manufacture, such as a machine-readable storage medium, configured with a computer program, where, upon execu-

tion, instructions in the computer program cause a machine to operate in accordance with process 16.

[0033] The process described herein is not limited to the embodiments set forth herein. The order of the blocks in **FIG. 2** may be changed to achieve the same result. The process is not limited to simulating rooms or indoor environments. The process can be used with any type of computer or video camera that together capture and process digital data. In this regard, the process is not limited to use with a digital video camera or to hand-held computers.

[0034] Other embodiments not described herein are also within the scope of the following claims.

What is claimed is:

1. A method comprising:
 - generating a virtual three-dimensional environment;
 - retrieving a virtual object; and
 - rendering the virtual object in the virtual three-dimensional environment.
2. The method of claim 1, wherein generating the virtual three-dimensional environment comprises:
 - surveying a real-world environment;
 - generating three-dimensional data that defines the real-world environment; and
 - rendering the virtual three-dimensional environment using the three-dimensional data.
3. The method of claim 2, wherein generating the virtual three-dimensional environment further comprises:
 - locating reference points in the real-world environment; and
 - recognizing planes in the real-world environment.
4. The method of claim 3, wherein recognizing a plane comprises identifying a cluster of points that behave similarly during motion.
5. The method of claim 1, further comprising:
 - scaling the virtual object in accordance with a size of the virtual three-dimensional environment.
6. The method of claim 5, wherein scaling comprises:
 - obtaining a size of a target in the real-world; and
 - changing a size of the virtual object in accordance with the size of the target.
7. The method of claim 1, further comprising:
 - illuminating the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.
8. The method of claim 1, wherein the virtual object is retrieved from a database comprised of pre-stored virtual objects.
9. A method comprising:
 - surveying a real-world environment;
 - generating three-dimensional data that defines the real-world environment;
 - rendering a virtual three-dimensional environment using the three-dimensional data;
 - retrieving a virtual object from a database comprised of pre-stored virtual objects;

rendering the virtual object in the virtual three-dimensional environment; and

re-positioning the virtual object in the virtual three-dimensional environment.

10. The method of claim 9, further comprising:

illuminating the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.

11. An article comprising a machine-readable medium that stores executable instructions to:

generate a virtual three-dimensional environment;

retrieve a virtual object; and

render the virtual object in the virtual three-dimensional environment.

12. The article of claim 11, wherein generating the virtual three-dimensional environment comprises:

surveying a real-world environment;

generating three-dimensional data that defines the real-world environment; and

rendering the virtual three-dimensional environment using the three-dimensional data.

13. The article of claim 12, wherein generating the virtual three-dimensional environment further comprises:

locating reference points in the real-world environment; and

recognizing planes in the real-world environment.

14. The article of claim 13, wherein recognizing a plane comprises identifying a cluster of points that behave similarly during motion.

15. The article of claim 11, further comprising instructions that cause the machine to:

scale the virtual object in accordance with a size of the virtual three-dimensional environment.

16. The article of claim 15, wherein scaling comprises:

obtaining a size of a target in the real-world; and

changing a size of the virtual object in accordance with the size of the target.

17. The article of claim 11, further comprising instructions that cause the machine to:

illuminate the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.

18. The article of claim 11, wherein the virtual object is retrieved from a database comprised of pre-stored virtual objects.

19. An article comprising a machine-readable medium that stores executable instructions to:

survey a real-world environment;

generate three-dimensional data that defines the real-world environment;

render a virtual three-dimensional environment using the three-dimensional data;

retrieve a virtual object from a database comprised of pre-stored virtual objects;

render the virtual object in the virtual three-dimensional environment; and

re-position the virtual object in the virtual three-dimensional environment.

20. The article of claim 19, further comprising instructions that cause the machine to:

illuminate the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.

21. An apparatus comprising:

a memory that stores executable instructions; and

a processor that executes the instructions to:

generate a virtual three-dimensional environment;

retrieve a virtual object; and

render the virtual object in the virtual three-dimensional environment.

22. The apparatus of claim 21, wherein generating the virtual three-dimensional environment comprises:

surveying a real-world environment;

generating three-dimensional data that defines the real-world environment; and

rendering the virtual three-dimensional environment using the three-dimensional data.

23. The apparatus of claim 22, wherein generating the virtual three-dimensional environment further comprises:

locating reference points in the real-world environment; and

recognizing planes in the real-world environment.

24. The apparatus of claim 23, wherein recognizing a plane comprises identifying a cluster of points that behave similarly during motion.

25. The apparatus of claim 21, wherein the processor executes instructions to:

scale the virtual object in accordance with a size of the virtual three-dimensional environment.

26. The apparatus of claim 25, wherein scaling comprises:

obtaining a size of a target in the real-world; and

changing a size of the virtual object in accordance with the size of the target.

27. The apparatus of claim 21, wherein the processor executes instructions to:

illuminate the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.

28. The apparatus of claim 21, wherein the virtual object is retrieved from a database comprised of pre-stored virtual objects.

29. An apparatus comprising:

a memory that stores executable instructions; and

a processor that executes the instructions to:

survey a real-world environment;

generate three-dimensional data that defines the real-world environment;

render a virtual three-dimensional environment using the three-dimensional data;

retrieve a virtual object from a database comprised of pre-stored virtual objects;

render the virtual object in the virtual three-dimensional environment; and

re-position the virtual object in the virtual three-dimensional environment.

30. The apparatus of claim 29, wherein the processor executes instructions to:

illuminate the virtual object in the virtual three-dimensional environment to simulate lighting in the real-world environment.

31. The method of claim 9, wherein surveying is performed using a camera; and

further comprising:

monitoring a location of the camera during surveying; and

using the location of the camera in rendering the virtual object.

32. The article of claim 19, wherein surveying is performed using input from a camera; and

further comprising instructions that cause the computer to:

monitor a location of the camera during surveying; and

use the location of the camera in rendering the virtual object.

33. The apparatus of claim 29, wherein surveying is performed using input from a camera; and

the processor executes instructions to:

monitor a location of the camera during surveying; and

use the location of the camera in rendering the virtual object.

* * * * *