



(51) International Patent Classification:
G06T 19/00 (2011.01)

(21) International Application Number:
PCT/US2017/036927

(22) International Filing Date:
12 June 2017 (12.06.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/190,085 22 June 2016 (22.06.2016) US

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

(72) Inventors: MICHAIL, Ashraf Ayman; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way,

Redmond, Washington 98052-6399 (US). KLEIN, Georg; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). PEARSON, Andrew Martin; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). MATHE, Zsolt; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). GROSSMAN, Mark S.; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). XU, Ning; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: MINHAS, Sandip et al.; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

(54) Title: DEPTH-AWARE REPROJECTION

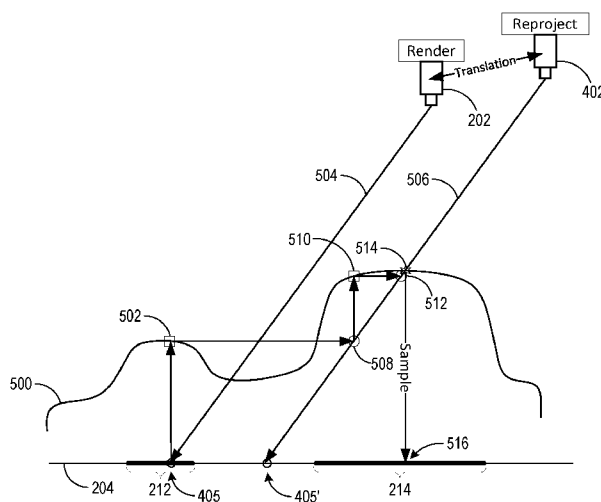


FIG. 5

(57) Abstract: Examples are disclosed that relate to depth-aware late-stage reprojection. One example provides a computing system configured to receive and store image data, receive a depth map for the image data, processing the depth map to obtain a blurred depth map, and based upon motion data, determine a translation to be made to the image data. Further, for each pixel, the computing system is configured to translate an original ray extending from an original virtual camera location to an original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location, determine a location at which the reprojected ray intersects the blurred depth map, and sample a color of a pixel for display based upon a color corresponding to the location at which the reprojected ray intersects the blurred depth map.

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

DEPTH-AWARE REPROJECTION

BACKGROUND

[0001] Head-mounted display devices and other portable display devices may be used to display immersive virtual reality and augmented reality presentations.

SUMMARY

[0002] Examples are disclosed that relate to translating, on a display device, rendered image data stored in a frame buffer prior to display based upon motion data. One example provides a computing system comprising a display, a sensor subsystem, a logic subsystem, and a reprojection subsystem configured to receive image data to be displayed via the display, receive a depth map for the image data, store the image data in a frame buffer, process the depth map to obtain a blurred depth map, and based upon motion data received from the sensor subsystem, determine a translation to be made to the image data prior to display. The reprojection subsystem is further configured to, for each pixel of a plurality of pixels in the frame buffer, translate an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location, determine a location at which the reprojected ray intersects the blurred depth map, and sample a color of a pixel for display based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 shows a use scenario for an example head-mounted display device.

[0005] FIG. 2 shows an example two-dimensional viewing frustum for a head-mounted display device.

[0006] FIG. 3 shows example effects of applying rotational and translational corrections to rendered image data.

[0007] FIG. 4 shows example rays cast for a pixel from a rendering camera position and a reprojected camera position after translation.

[0008] FIG. 5 shows a schematic illustration of example successive approximation steps for determining a color at a reprojected frame buffer pixel.

[0009] FIGS. 6A and 6B shows an example method of translating rendered image data prior to display based upon motion data.

5 [0010] FIG. 7 shows a block diagram of example depth-aware reprojection processing system.

[0011] FIG. 8 shows a block diagram of an example head-mounted display system.
DETAILED DESCRIPTION

[0012] A virtual or augmented reality display device may render virtual images for
10 display based on a perspective of the display device such that a virtual object appears at an intended location relative to other visible objects. For example, on a see-through head-mounted display device (HMD), a virtual object may be displayed as world-locked, such that it appears to a user of the HMD as being located at a fixed position relative to a real-world background as the user moves within the environment. FIG. 1 shows an example use
15 scenario 100 in which a user 101 wearing an example HMD 102 is playing an augmented reality game in a room and viewing displayed virtual objects 106 and 108. HMD 102 may map the real-world environment, e.g. using image data (depth and/or two-dimensional image data) of the real-world via an on-board image sensing system, and may render virtual objects 106 and 108 at respective locations within a field of view 110 of HMD 102 based
20 on the resulting map. Then, as the user moves within the room, image data and/or other motion data (e.g. inertial measurement unit (IMU) data) may be used to update the position of the device with respect to the virtual objects 106 and 108, and thereby update the displayed position of virtual objects 106 and 108 as appropriate.

[0013] However, in some instances, issues related to latency between image
25 rendering and display may affect the presentation of virtual imagery. For example, user 101 of HMD 102 may move his/her head between the time of rendering and display. In this instance, virtual objects 106 and 108 may appear to jump or otherwise move with respect to their intended locations in the real-world background due to the user's movements during this time.

30 [0014] One possible method of addressing such latency issues is to reproject rendered image data prior to display based on rotations and translations of a user's head. However, such methods may result in visible artifacts when used to perform translational corrections. For example, such techniques may treat content as if it were solely planar content and then apply transformations to the content from a different perspective. Thus, in

frames where the content is not planar, e.g. in three-dimensional (3D) scenes where virtual objects are to appear at varying depths, virtual objects may appear to stretch or drift, as described in more detail below with regard to FIGS. 3 and 4. Further, doing late-stage reprojection and optical correction on a mobile GPU/CPU system may consume significant amounts of power that would no longer be available for application rendering. This may complicate the use of such techniques on mobile devices.

[0015] Accordingly, examples are disclosed that relate to the use of virtual object depth data in late-stage reprojection. The term “late-stage reprojection” refers to applying corrections after a scene has been rendered but before the scene is translated into photons.

As described in more detail below, depth-aware late-stage reprojection applies translational corrections to rendered image data prior to display using an iterative processes involving a blurred depth map for the image data. Briefly, a corrected color of each pixel in a frame buffer of stored rendered image data may be determined by a virtual ray cast from the reprojected position of a rendering camera of the display device to a location where the ray intersects the blurred depth map. A pixel at that location in the blurred depth map then may be used to correct the color at the pixel at which the virtual ray intersects the frame buffer. Depth-aware late-stage reprojection allows the stabilization of reprojected virtual images at varying depths without the use of a full-resolution depth map, and may be operated using relatively low power consumption and memory bandwidth, making the process suitable for portable devices. Further, as described in more detail below, the disclosed reprojection methods may be configured to converge in a deterministic time per pixel. This may be advantageous in portable devices and/or special-purpose hardware, as no matter the complexity of the original rendered image, the time and power required to reproject the image have fixed upper bounds.

[0016] Prior to discussing depth-aware late-stage reprojection, a brief description of a 3D image rendering process is provided. To render virtual objects for display, a GPU of the display device maps a known 3D scene of virtual objects in a 3D world coordinate system to a two-dimensional (2D) plane in a 2D camera coordinate system. This process is known as projection transformation. The 2D plane represents the 2D view space (also referred to herein as the viewing frustum) from the perspective of the camera in the 2D camera coordinate system, which corresponds to the perspective from the display device. Portions of virtual objects in the scene that fall outside of the view space of the camera are then clipped, e.g. discarded as to not be rendered, and the 2D camera coordinate system is again transformed into a 2D “clip space” in which rays cast from the camera are transformed

to be parallel and spaced apart based upon pixels at the far end of the viewing frustum. The GPU then converts the clip space into a pixel format, determines color values for the pixels based on the colors of the virtual objects in the clip space and their z-ordering, and stores the pixel color values in a frame buffer. A display then reads the color values from the frame
5 buffer for display.

[0017] Image rendering and depth-aware late-stage reprojection examples are described herein in the context of 2D geometry for clarity. In 2D geometry, points in a 2D scene are mapped to one dimension. FIG. 2 schematically illustrates partial rendering of an example 2D scene 200 on a display device, where a portion of a frame buffer 204 is shown
10 as a one-dimensional line, and a portion of a viewing frustum 206 (the field-of-view of camera 202) is represented by a fan of virtual rays. In 3D geometry, the frame buffer would be represented by a 2D plane, while the viewing frustum would be represented by an array of virtual rays. In FIG. 2, only a portion of the viewing frustum is shown for simplicity. 2D object 208 and 2D object 210 are shown as circles, representing examples of 2D objects
15 within the 2D scene 200. The display device may render 2D objects 208 and 210 by coloring the pixels in frame buffer 204 which correspond to 2D objects 208 and 210. These pixels correspond to those that are touched by virtual rays cast from camera 202 and intersect each of 2D objects 208 and 210. For example, if 2D object 208 is a green object, then the GPU renders the pixels in frame buffer portion 212 as green, as virtual rays cast from camera 202
20 which intersect object 208 touch pixels within frame buffer portion 212. Likewise, if 2D object 210 is a red object, then the display device may render pixels in frame buffer portion 214 as red.

[0018] As mentioned above, a brief amount of time may exist between rendering of a virtual object and display of the virtual object, within which movements of the display
25 device may affect the resulting displayed content. FIG. 3 shows example effects of rotational and translational changes in camera position during rendering of image data, e.g. via a GPU. At 300, an original view space is shown as a 2D representation of the field-of-view of a rendering camera within a 2D camera coordinate system, where the camera resides at the origin O. The original view space 300 contains three objects P0, P1, and P2 along a ray P
30 cast from the camera at the origin O. Near and far clip planes, which are specified ahead of time such that only portions of virtual objects between the near and far clip planes are displayed, are shown at Z_n and Z_f respectively. The viewing frustum of the camera here is the region contained by the dotted lines of Z_n and Z_f and the two rays R_1 and R_2 extending from the camera which define the field-of-view of the camera.

[0019] Rotation of the camera is shown at 302. In the final view space after rotation, shown at 304, original ray P has been transformed to ray P', but ray P' still intersects all three points P0, P1, and P2. As such, when transformed into clip space, shown at 306, all three points maintain the same x-coordinate. The GPU may then construct a screen space representation of the scene by mapping the clip space coordinates to pixels, and determining a color for each pixel in the screen space. Assuming that each object along the original ray P is intended to be opaque, the GPU will determine the point closest to the camera as the color for that pixel. Thus, in screen space after rotation, shown at 308, P2 is determined to be the color of the pixel at the corresponding screen space location, while P0 and P1 are occluded.

[0020] In contrast, translation of the camera is shown at 310. In the final view space 312, the three points P0, P1, and P2 no longer fall on a single ray but are instead defined by new rays P0', P1', and P2' relative to the translated camera position. Thus, each point maps to a different position in clip space 314, and therefore to different pixels in screen space 316. After translational reprojection, some pixels may become occluded by the translated imagery, and some previously occluded pixels may become unoccluded. It is noted that no information regarding the color of the newly unoccluded pixels is stored in the frame buffer. However, the example depth-aware reprojection methods described below may allow colors of these pixels to be inferred more easily than when depth information is not considered.

[0021] Each pixel in the frame buffer may be identified by the angle of a ray cast from the camera extending to that pixel in the frame buffer. FIG. 4 shows an example schematic illustrating a ray cast before and after virtually translating a position of rendering camera 202 to reprojected camera 402 using non-depth-aware reprojection techniques. As in FIG. 2, where 2D object 208 is a green object in the 2D scene, the pixel at 405 in frame buffer 204 is rendered as a green pixel because ray 404 cast from rendering camera 202 to frame buffer 204 at an angle α passes through 2D object 208. This is known to the display device because both the 2D geometry of the scene as well as the camera position are available to the display device.

[0022] However, determining the correct pixel color to display after reprojection due to translation may be challenging. For example, when determining the color of a pixel based upon reprojection, the device may only have the frame buffer and a depth map for the image data available to help make the determination. In planar homography, the GPU merely translates frame buffer pixel 405 by an amount corresponding to the camera translation amount and the distance between the camera and the plane of the image, shown

in FIG. 4 as pixel at 405'. As such, the GPU samples frame buffer 204 at pixel 405' and determines the reprojected color to be black, even though ray 406 cast from reprojected camera 302 extending to frame buffer 204 at angle α actually passes through 2D object 210, which is red and thus indicates that the pixel at 405' should be red. As such, planar homography may result in incorrect display of reprojected pixels due to the limited information available.

[0023] Thus, the examples disclosed herein may help to correct the issues of planar homography by utilizing a blurred depth map to achieve depth-aware late-stage reprojection. The use of a blurred depth map to inform late-stage reprojection may help to reduce a computational burden of the reprojection process, and also may help to ensure that an iterative process to locate the intersection of the reprojected ray and blurred depth map converges to a suitable degree within a relatively small and consistent number of computation cycles. This may allow such an iterative process to be implemented deterministically.

[0024] A blurred depth map may be obtained by reducing the resolution of an original depth map for the image data by a predetermined factor, and blurring the depth map via a convolution process. As one example of reducing resolution, a device may first reduce the resolution of the original depth map by a factor of two on each axis to obtain a quarter-resolution depth map, and then further reduced by a factor of four on each axis, resulting in $1/64^{\text{th}}$ of the original resolution. In other examples, any other suitable resolution reduction process may be used. Further, in some examples, the depth map received may have previously undergone at least some reduction of resolution.

[0025] FIG. 5 shows a schematic representation of a blurred depth map 500 of 2D scene 200, and illustrates an example method of determining a color to sample for display using blurred depth map 500. Each pixel in frame buffer 204 maps to a corresponding entry in blurred depth map 500. Blurred depth map 500 is depicted as a curved line, where each point on the curved line corresponds to a distance between rendering camera 202 and the nearest 2D object on the ray cast to a corresponding pixel. For example, the square 502 represents the depth value on blurred depth map 500 that corresponds to pixel 405. In addition to blurring, further bounds may be applied on the size of the translation and the distance to the nearest object.

[0026] Prior to translational correction, a rotational correction may be applied. This may be visualized by tilting both the curved line (depth map 500) and the straight line (frame buffer 204) relative to rendering camera 202 based on detected rotational changes in the

camera position due to head and/or device movement. Because a rotational correction transforms both the depth map and the frame buffer pixel locations equally, the mapping of each pixel to its corresponding depth value is preserved.

[0027] In contrast, as described above, translational correction relies on each pixel's depth entry to determine a translated location from which the device should sample values for the outbound frame buffer. Because each blurred depth map entry may correspond to 64 or more pixels, the device interpolates the sampling address based on the pixel address for each pixel. When rendering the original frame buffer, the device relies on the geometry of the 2D scene to determine which 2D object is closest to the rendering camera and thereby contributes the most to the depth value for each pixel. Depth-aware late-stage reprojection performs a similar process, but the depth map is used to infer the color that would have been rendered based on the 2D scene geometry. The GPU or other device performing the reprojection thus determines, for each pixel, where a reprojected ray cast from the rendering camera extending to a reprojected pixel intercepts the depth map.

[0028] The location at which reprojected ray 506 intersects blurred depth map 500 may be determined using successive approximation. In FIG. 5, the squares represent depth value estimates while the circles represent locations on the reprojected ray corresponding to depth value estimates. First, an initial depth value 502 is determined which corresponds to the original frame buffer location, illustrated as source pixel 405 along original ray 504. Then, an initial location 508 on reprojected ray 506 is determined that corresponds to initial depth value 502. Next, an updated depth value 510 on blurred depth map 500 is determined which corresponds to initial location 508 on reprojected ray 506. Next, an updated location 512 on reprojected ray 506 is determined that corresponds to updated depth value 510. The steps of determining the updated depth value on blurred depth map 500 and determining the updated location on reprojected ray may be iteratively repeated any suitable number of times. In some examples, these steps are repeated a fixed number of times, such that the iterative process is deterministic.

[0029] After a suitable or fixed number of iterative cycles, a location 514 at which reprojected ray 506 intersects blurred depth map 500 is determined. The device then samples frame buffer 500 location that corresponds to location 514, at 516 for display. Convergence to location 514 may occur fairly quickly, at least partially as a result of the blurring of the depth map. Where a fixed number of cycles is used, the number of cycles selected may depend upon how quickly the successive approximation is determined to converge to a suitable accuracy. The use of a deterministic fixed number of iterations may be a

programmed parameter, and may provide various advantages. For example, the use of a fixed number of cycles helps to control an amount of processing power used for each reprojected frame of image data, and to maintain a consistent data flow through the graphics processing pipeline. In other implementations, the process may utilize a convergence
5 threshold rather than a fixed number of iterations per pixel.

[0030] FIG. 6A and 6B show an example method 600 of performing depth-aware late-stage reprojection. Method 600 may be enacted on a computing system having a display, such as a head-mounted display device, or on any other suitable type of display device. While described in the context of a single frame buffer, in some examples more than
10 one frame buffer may be sampled (e.g., one frame buffer for each eye) to produce a more accurate color sample for display, as such a multiple frame buffer method may generate fewer disocclusions along the axis of the multiple samples.

[0031] Method 600 includes, at 602, receiving a depth map for the image data, and at 604, receiving image data to be displayed via a display. Such image data may include
15 virtual objects and/or images to be displayed for a virtual or augmented reality experience. The depth map and image data each may be retrieved from local or remote storage. In various embodiments, the depth map may be received before the image data, or after the image data. Receiving the depth data first may allow translational corrections to be “streamed” out (e.g. output as the corrections are determined, rather than waiting for all
20 corrections of the image to be completed prior to output) as the image data is received, which may help to reduce latency.

[0032] Method 600 further includes, at 606 storing the image data in a frame buffer. In parallel, method 600 includes, at 608, processing the depth map. For example, the resolution of the depth map may be reduced, as shown at 610, and the depth map may be
25 blurred, as shown at 612. In some examples, background values and weights below threshold values may be omitted during the depth map blur convolution, as indicated at 613, which may increase accuracy in augmented reality scenarios along silhouettes of objects. In other examples, such background values and weights may be utilized instead of omitted.

[0033] Next, method 600 includes, at 614, receiving motion data from a sensor
30 subsystem. The motion data may comprise any suitable sensor data indicative of motion, including but not limited to data received from or derived from one or more depth sensors, two-dimensional image sensors, inertial measurement units, and/or eye position sensors. Such motion data may be utilized to determine any adjustment(s) to be made to the rendered image data via late-stage reprojection, as shown at 616, which may include determining a

rotation at 617 and/or determining a translation at 618. If applicable, method 600 may include, at 619, making a rotational adjustment to the rendered image data based upon the rotation determined at 617.

[0034] Continuing with FIG. 6B, method 600 includes, at 620, for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined, translating an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location. In some examples, this may be performed for all pixels in the frame buffer, while in other examples, this may be performed on a basis other than per-pixel. For example, the translation may be performed at the resolution of the blurred depth map, and then interpolated to full resolution. FIG. 4, described above, shows an original ray 404 translated to reprojected ray 406.

[0035] Method 600 further includes, at 624, determining a location at which the reprojected ray intersects the blurred depth map. As described herein, this location may be determined via successive approximation. As an example, method 600 may include, at 626, determining an initial depth value on the blurred depth map corresponding to the original frame buffer location, and at 628, determining an initial location on the reprojected ray corresponding to the initial depth value. Then, method 600 includes, at 630, determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and at 632, determining an updated location on the reprojected ray corresponding to the updated depth value. As indicated at 634, method 600 may comprise iteratively repeating steps 630 and 632 until a specified condition is met. For example, steps 630 and 632 may be iteratively repeated for a fixed number of times for each pixel in a deterministic manner. In other examples, successive approximation may be carried out in a non-deterministic manner, for example by repeating iterations until a convergence threshold is reached.

[0036] For each pixel in the frame buffer, method 600 further includes, at 638, sampling a color of a pixel for the reprojected frame buffer location based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map. Method 600 then includes, at 640, displaying the reprojected image data. It will be understood that the depth-aware late-stage reprojection process as disclosed herein may be implemented for each eye of the user using data from the perspective of each eye, in the case of a stereoscopic HMD.

[0037] FIG. 7 shows a block diagram of an example depth-aware late-stage

reprojection system 700 that may be used to implement the example depth-aware late-stage reprojection process described above. System 700 may be implemented on any suitable computing system, including but not limited to a HMD. System 700 includes a renderer 701 for rendering image data, e.g. via a host GPU or CPU, which outputs an image data superframe 702 comprising a depth sub-region 704 and color sub-regions 706 for virtual images to be displayed. The depicted superframe 702 illustrates an example of an arrangement of depth and color sub-regions. In other examples, these subregions may be arranged in any other suitable manner. In this example, depth sub-region 704 has been downsampled, and thus occupies a smaller portion of superframe 702 than the color image data. In other examples, the depth sub-region 704 may be sent to the input stage 708 as full fidelity depth data without initial downsampling.

[0038] The superframe 702 is received at input stage 708. Input stage 708 includes a depth map blurring module 710 that further downsamples and also blurs the depth data, and subsequently outputs blurred depth data and color pixel data to a frame buffer shown as dynamic random access memory (DRAM) 712. Blurred depth data is further provided to depth random access memory (RAM) 714, from which depth correction module 716 may retrieve the blurred depth data for performing depth-aware late-stage reprojection (DLSR). The DLSR process is shown as utilizing parameters input to the depth correction module 716. Examples of such parameters may include unprojected sampling coordinates (unprojected u, v coordinates), dimensions of the depth map, rotation and translation values as determined from motion tracking, scale factors between the frame buffer and the blurred depth data, and scale factors between the blurred depth data and output display dimensions. As an example, pixel 405 of FIG. 5 corresponds to the unprojected sampling coordinates in the color plane, which are used as input to depth correction module 716. Depth correction module 716 further may apply suitable corrections to the blurred depth data, such as a bilateral filter and/or a Gaussian filter.

[0039] Depth correction module 716 outputs the corrected, reprojected sampling coordinates (shown as u, v indices) to pixel sampler 718, which informs the pixel sampler 718 of what colors to use for the pixels. As an example, pixel 516 of FIG. 5 corresponds to the reprojected sampling coordinates in the color plane, which are output to pixel sampler 718. The colored pixels are then output to display 722.

[0040] In some embodiments, the methods and processes described herein may be tied to a computing system of one or more computing devices. In particular, such methods and processes may be implemented as a computer-application program or service, an

application-programming interface (API), a library, and/or other computer-program product.

[0041] FIG. 8 shows a block diagram of an example head-mounted display system 800. The head-mounted display device 102 represents an example implementation of head-mounted display system 800 in which the depth-aware reprojection examples described herein may be implemented. Display system 800 includes a sensor subsystem 806, which may include one or more outward-facing image sensors 808 configured to acquire images of a real-world environment, such as to help determine the position of the display system 800 relative to the real-world environment, and one or more microphones 810 configured to detect sounds, such as user speech inputs. Outward-facing image sensors 808 may include one or more depth sensor(s) and/or one or more two-dimensional image sensor(s) (e.g. RGB image sensors). Head-mounted display system 800 may display completely virtual images, may display video-based augmented reality images via a viewfinder mode using data from an outward-facing image sensor, or may display augmented reality images (e.g. mixed reality images) via a see-through display subsystem.

[0042] Sensor subsystem 806 may further include a gaze detection subsystem 812 configured to detect a gaze direction of a user, e.g. as user input for computing device actions. Gaze detection subsystem 812 may be configured to determine gaze directions of each of a user's eyes in any suitable manner. For example, in the depicted embodiment, gaze detection subsystem 812 comprises one or more glint sources 814, such as infrared light sources configured to cause a glint of light to reflect from each eyeball of a user, and one or more image sensor(s) 816, such as inward-facing sensors, configured to capture an image of each eyeball of the user. Changes in the glints from the user's eyeballs and/or a location of a user's pupil as determined from image data gathered via the image sensor(s) 816 may be used to determine a direction of gaze. In other examples, gaze detection subsystem 812 may be omitted.

[0043] Display system 800 also may include additional sensors, as mentioned above. For example, display system 800 may include non-imaging sensor(s) 818, examples of which may include but are not limited to a global positioning system (GPS) sensor, and one or more motion sensor(s) 819, such as an accelerometer, a gyroscopic sensor, and/or an inertial measurement unit (IMU). Such sensor(s) may be used in determining image display adjustments to make via late-stage reprojection.

[0044] Motion sensors 819, as well as microphone(s) 810 and gaze detection subsystem 812, also may be employed as user input devices, such that a user may interact

with the display system 800 via gestures of the eye, neck and/or head, as well as via verbal commands. It will be understood that sensors illustrated in FIG. 8 are shown for the purpose of example and are not intended to be limiting in any manner, as any other suitable sensors and/or combination of sensors may be utilized.

5 **[0045]** Display system 800 further includes one or more speaker(s) 820, for example, to provide audio outputs to a user for user interactions. Display system 800 further includes a controller 822 having a logic subsystem 824 and a storage subsystem 826 in communication with the sensors, a communication subsystem 828, and/or other components.

10 **[0046]** Logic subsystem 824 includes one or more physical devices configured to execute instructions. For example, logic subsystem 824 may be configured to execute instructions that are part of one or more applications, services, programs, routines, libraries, objects, components, data structures, or other logical constructs. Such instructions may be implemented to perform a task, implement a data type, transform the state of one or more
15 components, achieve a technical effect, or otherwise arrive at a desired result.

[0047] Logic subsystem 824 may include one or more processors configured to execute software instructions. Additionally or alternatively, logic subsystem 824 may include one or more hardware or firmware logic subsystems configured to execute hardware or firmware instructions. Processors of logic subsystem 824 may be single-core or multi-
20 core, and the instructions executed thereon may be configured for sequential, parallel, and/or distributed processing. Individual components of logic subsystem 824 optionally may be distributed among two or more separate devices, which may be remotely located and/or configured for coordinated processing. Aspects of logic subsystem 824 may be virtualized and executed by remotely accessible, networked computing devices configured in a cloud-
25 computing configuration.

[0048] Storage subsystem 826 includes one or more physical devices configured to hold instructions executable by the logic subsystem to implement the methods and processes described herein, e.g. to perform various tasks related to depth-aware late-stage reprojection. When such methods and processes are implemented, the state of storage subsystem 826 may
30 be transformed—e.g., to hold different data.

[0049] Storage subsystem 826 may include removable and/or built-in devices. Storage subsystem 826 may include optical memory (e.g., CD, DVD, HD-DVD, Blu-Ray Disc, etc.), semiconductor memory (e.g., RAM, EPROM, EEPROM, etc.), and/or magnetic memory (e.g., hard-disk drive, floppy-disk drive, tape drive, MRAM, etc.), among others.

Storage subsystem 826 may include volatile, nonvolatile, dynamic, static, read/write, read-only, random-access, sequential-access, location-addressable, file-addressable, and/or content-addressable devices.

5 [0050] It will be appreciated that storage subsystem 826 includes one or more physical devices. However, aspects of the instructions described herein alternatively may be propagated by a communication medium (e.g., an electromagnetic signal, an optical signal, etc.) that is not held by a physical device for a finite duration.

10 [0051] Aspects of logic subsystem 824 and storage subsystem 826 may be integrated together into one or more hardware-logic components. Such hardware-logic components may include field-programmable gate arrays (FPGAs), program- and application-specific integrated circuits (PASIC / ASICs), program- and application-specific standard products (PSSP / ASSPs), system-on-a-chip (SOC), and complex programmable logic devices (CPLDs), for example.

15 [0052] Communication subsystem 828 may be configured to communicatively couple the display system 800 with one or more other computing devices. Communication subsystem 828 may include wired and/or wireless communication devices compatible with one or more different communication protocols. As non-limiting examples, communication subsystem 828 may be configured for communication via a wireless telephone network, or a wired or wireless local- or wide-area network. In some embodiments, communication
20 subsystem 828 may allow display system 800 to send and/or receive messages to and/or from other devices via a network such as the Internet.

[0053] Display system 800 further includes reprojection subsystem 830 to implement depth-aware late-stage reprojection as described in the examples above. For example, reprojection subsystem 830 may receive image data and a depth map for the image
25 data, process the depth map to obtain a blurred depth map, and output the blurred depth map and color pixel data to a frame buffer. Further, reprojection subsystem 830 may determine a translation to make to the image data based upon motion data, and for each pixel of a plurality of pixels in the frame buffer, translate an original ray cast from the original rendering camera's position to an original frame buffer location, e.g. at unprojected
30 sampling coordinates, to a reprojected ray cast from the translated camera location to a reprojected frame buffer location, e.g. at reprojected sampling coordinates. Reprojection subsystem 830 may then sample the color of the pixel at the reprojected sampling coordinates (which corresponds to the location at which the reprojected ray intersects the blurred depth map) for display by display subsystem 834. The reprojection subsystem may

be implemented in any suitable manner, such as by GPU or CPU. While shown in FIG. 8 as being separate from controller 822, in some implementations the reprojection subsystem may be a part of a controller.

[0054] Display subsystem 834 may include one or more lenses 832, such that images may be displayed via lenses 832 (e.g. via projection onto lenses 832, waveguide system(s) incorporated into lenses 832, and/or in any other suitable manner). Display subsystem 834 may include a backlight and a microdisplay, such as liquid-crystal display (LCD) or liquid crystal on silicon (LCOS) display, in combination with one or more light-emitting diodes (LEDs), laser diodes, and/or other light sources. In other examples, the display subsystem 834 may utilize quantum-dot display technologies, active-matrix organic LED (OLED) technology, and/or any other suitable display technologies.

[0055] Head-mounted display system 800 is described for the purpose of example, and thus is not meant to be limiting. It is to be understood that head-mounted display system 800 may include additional and/or alternative sensors, cameras, microphones, input devices, output devices, etc. than those shown without departing from the scope of this disclosure. Head-mounted display system 800 may be implemented as a virtual reality system, or a video or optical augmented reality system. Additionally, the physical configuration of a display device and its various sensors and subcomponents may take a variety of different forms without departing from the scope of this disclosure. Further, it will be understood that the methods and processes described herein may be implemented as a computer-application program or service, an application-programming interface (API), a library, and/or other computer program product. Such computer program products may be executable locally on head-mounted display system 800 or other suitable display system, or may be executable remotely on a computing system in communication with head-mounted display system 800. It will further be understood that the depth-aware late-stage reprojection method as disclosed herein may be implemented on any other suitable display devices, as latency issues may arise in other display devices configured to display virtual content.

[0056] Another example provides a computing system comprising a display, a sensor subsystem, and a reprojection subsystem configured to receive image data to be displayed via the display, receive a depth map for the image data, store the image data in a frame buffer, processing the depth map to obtain a blurred depth map, based upon motion data received from the sensor subsystem, determine a translation to be made to the image data for display, and for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined, translate an original ray extending from an original virtual

camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location, determine a location at which the reprojected ray intersects the blurred depth map, and sample a color of a pixel for display based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map. The computing system may additionally or alternatively include a head-mounted display device. The reprojection subsystem may additionally or alternatively be configured to display via the display the image data after translating the original ray to the reprojected ray. The reprojection subsystem may additionally or alternatively be configured to make a rotation to the image data based upon motion data received from the sensor subsystem. The reprojection subsystem may additionally or alternatively be configured to process the depth map to obtain the blurred depth map by omitting background values and weights beyond threshold values during a depth map blur convolution. The reprojection subsystem may additionally or alternatively be configured to determine the location at which the reprojected ray intersects the blurred depth map by determining an initial depth value on the blurred depth map corresponding to the original frame buffer location, then determining an initial location on the reprojected ray corresponding to the initial depth value, then determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and then determining an updated location on the reprojected ray corresponding to the updated depth value. The reprojection subsystem may additionally or alternatively be configured to determine the location at which the reprojected ray intersects the blurred depth map by iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray. The reprojection subsystem may additionally or alternatively be configured to iteratively repeat the determining of the updated depth value and the determining of the updated location on the reprojected ray a fixed number of times for each pixel in the frame buffer. The reprojection subsystem may additionally or alternatively be configured to reduce a resolution of the depth map. Any of the above-described examples may be combined in any suitable manner in various implementations.

[0057] Another examples provides, on a computing system comprising a display and a sensor subsystem, a method comprising receiving image data to be displayed via the display, receiving a depth map for the image data, storing the image data in a frame buffer, processing the depth map to obtain a blurred depth map, based upon motion data received from the sensor subsystem, determining a translation to be made to the image data prior to

display, and for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined, translating an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location, determining a location at which the reprojected ray intersects the blurred depth map, and sampling a color of a pixel in the frame buffer for display at the reprojected frame buffer location based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map. The method may additionally or alternatively include displaying via the display the image data after translating the original ray to the reprojected ray for each pixel in the frame buffer. The plurality of pixels may additionally or alternatively include each pixel in the frame buffer. The method may additionally or alternatively include determining the location at which the reprojected ray intersects the blurred depth map by determining an initial depth value on the blurred depth map corresponding to the original frame buffer location, then determining an initial location on the reprojected ray corresponding to the initial depth value, then determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and then determining an updated location on the reprojected ray corresponding to the updated depth value. The method may additionally or alternatively include determining the location at which the reprojected ray intersects the blurred depth map by iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray. The method may additionally or alternatively include iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray a same number of times for each pixel in the frame buffer. The method may additionally or alternatively include reducing a resolution of the depth map. Any of the above-described examples may be combined in any suitable manner in various implementations.

[0058] Another example provides a head-mounted display device comprising a display, a sensor subsystem, and a reprojection subsystem configured to receive image data to be displayed via the display, receive depth data for the image data, store the image data in a frame buffer, process the depth map to obtain a blurred depth map, based upon motion data received from the sensor subsystem, determine a translation to be made to the image data prior to display, and for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined, translate an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray

extending from a translated camera location to a reprojected frame buffer location, determine a location at which the reprojected ray intersects the blurred depth map using successive approximation, and sample a color of a pixel in the frame buffer for display at the reprojected frame buffer location based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map. The reprojection subsystem may additionally or alternatively be configured to determine the location at which the reprojected ray intersects the blurred depth map using successive approximation by determining an initial depth value on the blurred depth map corresponding to the original frame buffer location, then determining an initial location on the reprojected ray corresponding to the initial depth value, then determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and then determining an updated location on the reprojected ray corresponding to the updated depth value. The reprojection subsystem may additionally or alternatively be configured to iteratively repeat the determining of the updated depth value and the determining of the updated location on the reprojected ray a fixed number of times for each pixel in the frame buffer. The reprojection subsystem may additionally or alternatively be configured to determine the translation to be made to the image data based on one or more additional frame buffers and corresponding blurred depth maps. Any of the above-described examples may be combined in any suitable manner in various implementations.

[0059] It will be understood that the configurations and/or approaches described herein are exemplary in nature, and that these specific embodiments or examples are not to be considered in a limiting sense, because numerous variations are possible. The specific routines or methods described herein may represent one or more of any number of processing strategies. As such, various acts illustrated and/or described may be performed in the sequence illustrated and/or described, in other sequences, in parallel, or omitted. Likewise, the order of the above-described processes may be changed.

[0060] The subject matter of the present disclosure includes all novel and nonobvious combinations and subcombinations of the various processes, systems and configurations, and other features, functions, acts, and/or properties disclosed herein, as well as any and all equivalents thereof.

CLAIMS

1. A computing system comprising a display, a sensor subsystem, and a reprojection subsystem configured to:

receive image data to be displayed via the display;

receive a depth map for the image data;

store the image data in a frame buffer;

process the depth map to obtain a blurred depth map;

based upon motion data received from the sensor subsystem, determine a translation to be made to the image data for display; and

for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined,

translate an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location,

determine a location at which the reprojected ray intersects the blurred depth map, and

sample a color of a pixel for display based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map.

2. The computing system of claim 1, wherein the computing system comprises a head-mounted display device.

3. The computing system of claim 1, wherein the reprojection subsystem is further configured to display via the display the image data after translating the original ray to the reprojected ray.

4. The computing system of claim 1, wherein the reprojection subsystem is further configured to make a rotation to the image data based upon motion data received from the sensor subsystem.

5. The computing system of claim 1, wherein the reprojection subsystem is configured to process the depth map to obtain the blurred depth map by omitting background values and weights beyond threshold values during a depth map blur convolution.

6. The computing system of claim 1, wherein the reprojection subsystem is further configured to determine the location at which the reprojected ray intersects the blurred depth map by

determining an initial depth value on the blurred depth map corresponding to the original frame buffer location,

then determining an initial location on the reprojected ray corresponding to the initial depth value,

then determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and

then determining an updated location on the reprojected ray corresponding to the updated depth value.

7. The computing system of claim 6, wherein the reprojection subsystem is further configured to determine the location at which the reprojected ray intersects the blurred depth map by iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray.

8. The computing system of claim 6, wherein the reprojection subsystem is further configured to iteratively repeat the determining of the updated depth value and the determining of the updated location on the reprojected ray a fixed number of times for each pixel in the frame buffer.

9. The computing system of claim 1, wherein the reprojection subsystem is further configured to reduce a resolution of the depth map.

10. On a computing system comprising a display and a sensor subsystem, a method comprising:

receiving image data to be displayed via the display;

receiving a depth map for the image data;

storing the image data in a frame buffer;

processing the depth map to obtain a blurred depth map;

based upon motion data received from the sensor subsystem, determining a translation to be made to the image data prior to display; and

for each pixel of a plurality of pixels in the frame buffer, based upon the translation determined,

translating an original ray extending from an original virtual camera location to a corresponding original frame buffer location to a reprojected ray extending from a translated camera location to a reprojected frame buffer location,

determining a location at which the reprojected ray intersects the blurred depth map, and

sampling a color of a pixel in the frame buffer for display at the reprojected frame buffer location based upon a color in the frame buffer corresponding to the location at which the reprojected ray intersects the blurred depth map.

11. The method of claim 10, further comprising displaying via the display the image data after translating the original ray to the reprojected ray for each pixel in the frame buffer.
12. The method of claim 10, wherein the plurality of pixels comprises each pixel in the frame buffer.
13. The method of claim 10, wherein determining the location at which the reprojected ray intersects the blurred depth map comprises
 - determining an initial depth value on the blurred depth map corresponding to the original frame buffer location,
 - then determining an initial location on the reprojected ray corresponding to the initial depth value,
 - then determining an updated depth value on the blurred depth map corresponding to the initial location on the reprojected ray, and
 - then determining an updated location on the reprojected ray corresponding to the updated depth value.
14. The method of claim 13, wherein determining the location at which the reprojected ray intersects the blurred depth map comprises iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray.
15. The method of claim 14, further comprising iteratively repeating the determining of the updated depth value and the determining of the updated location on the reprojected ray a same number of times for each pixel in the frame buffer.

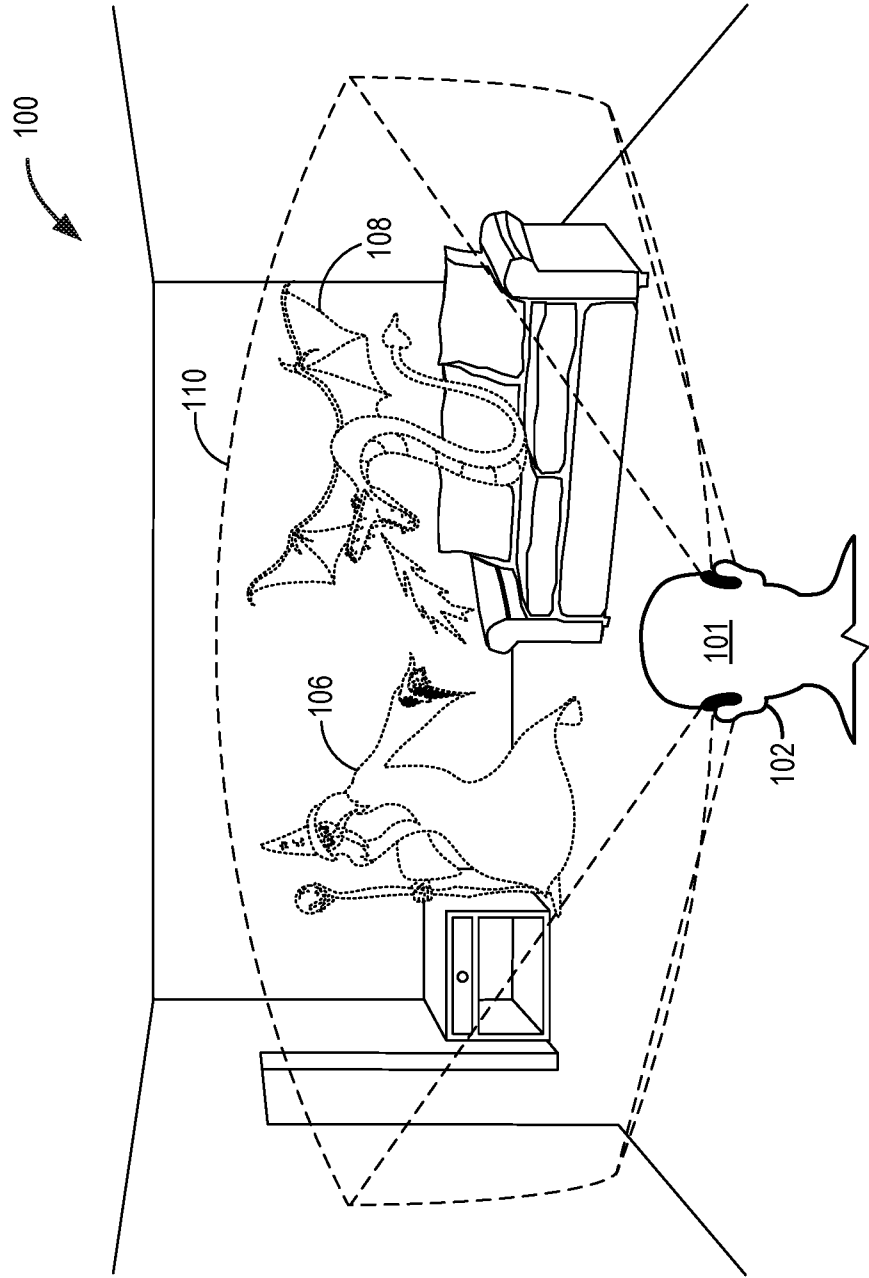


FIG. 1

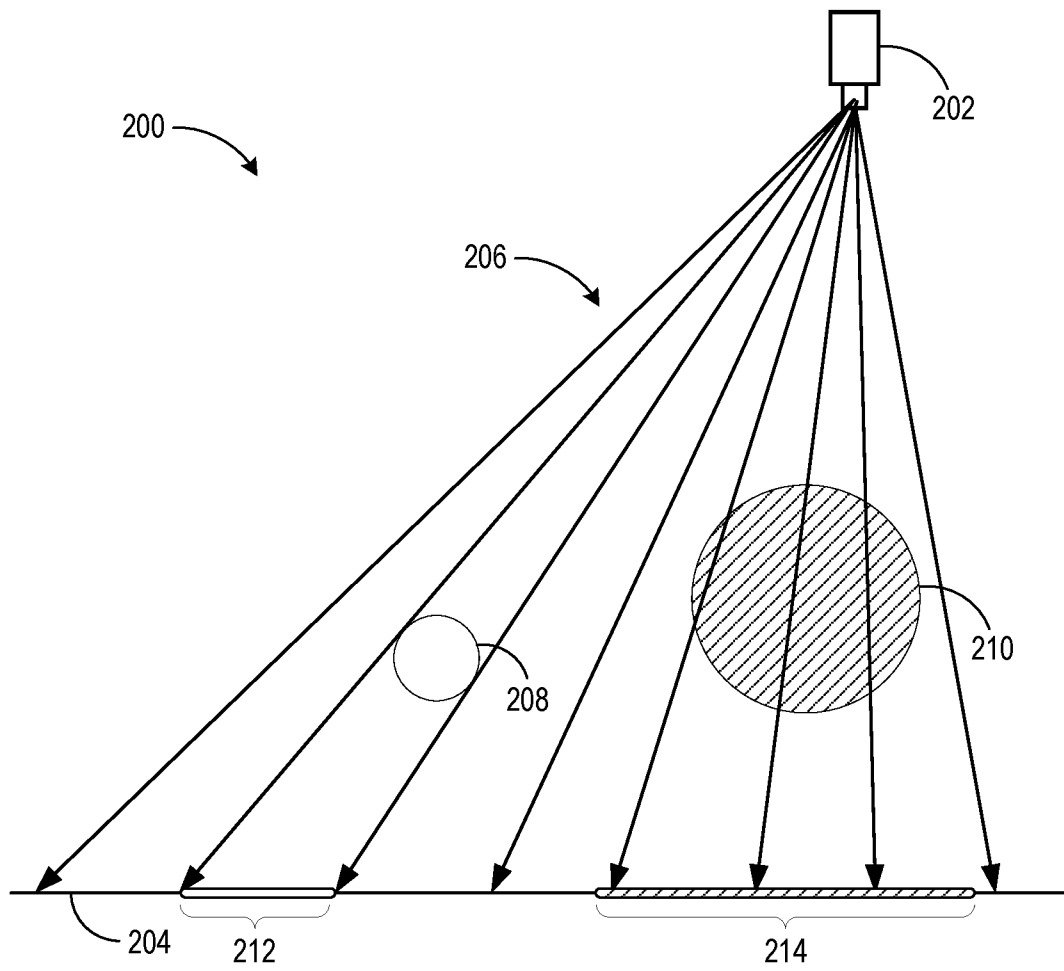


FIG. 2

3/9

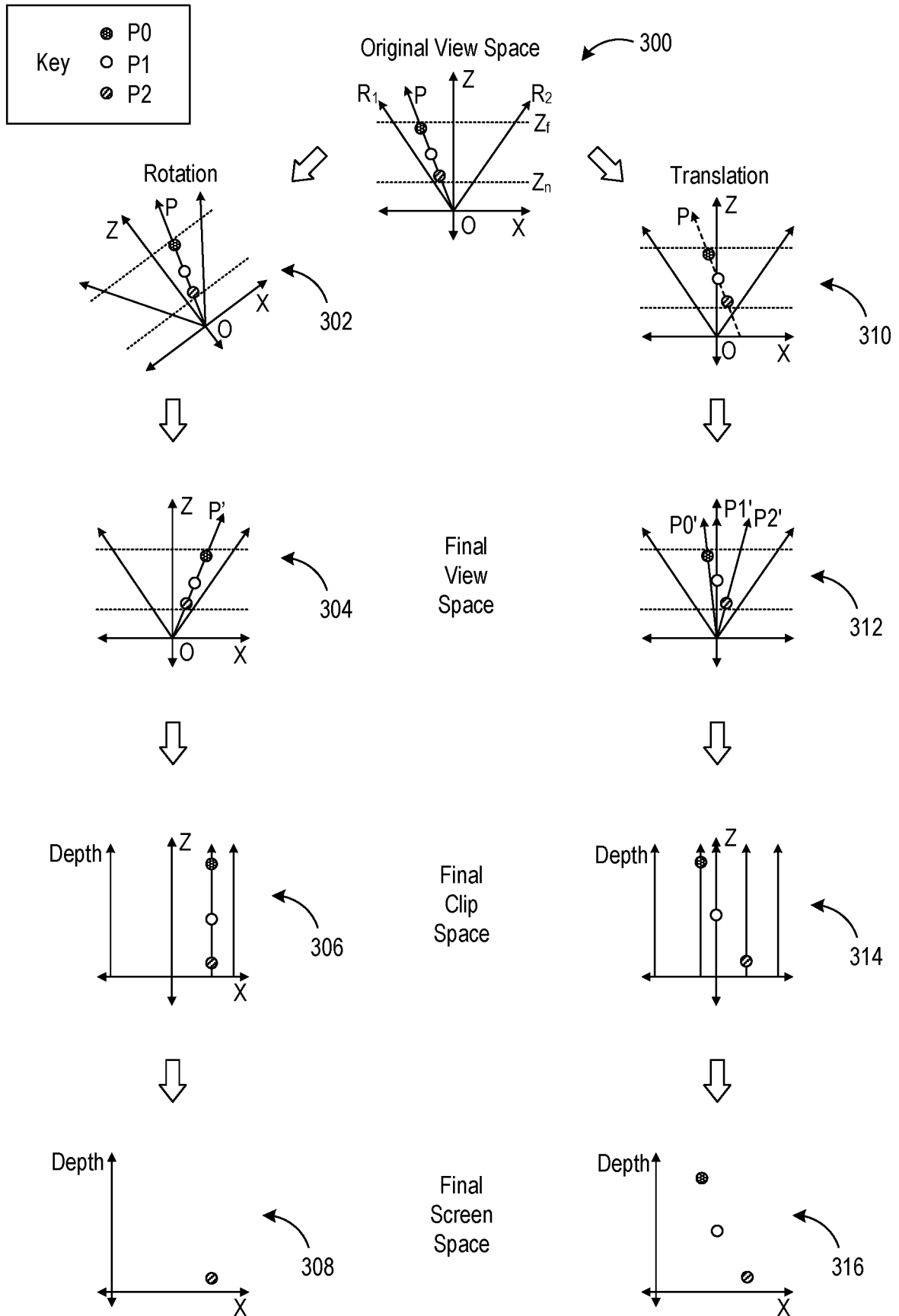


FIG. 3

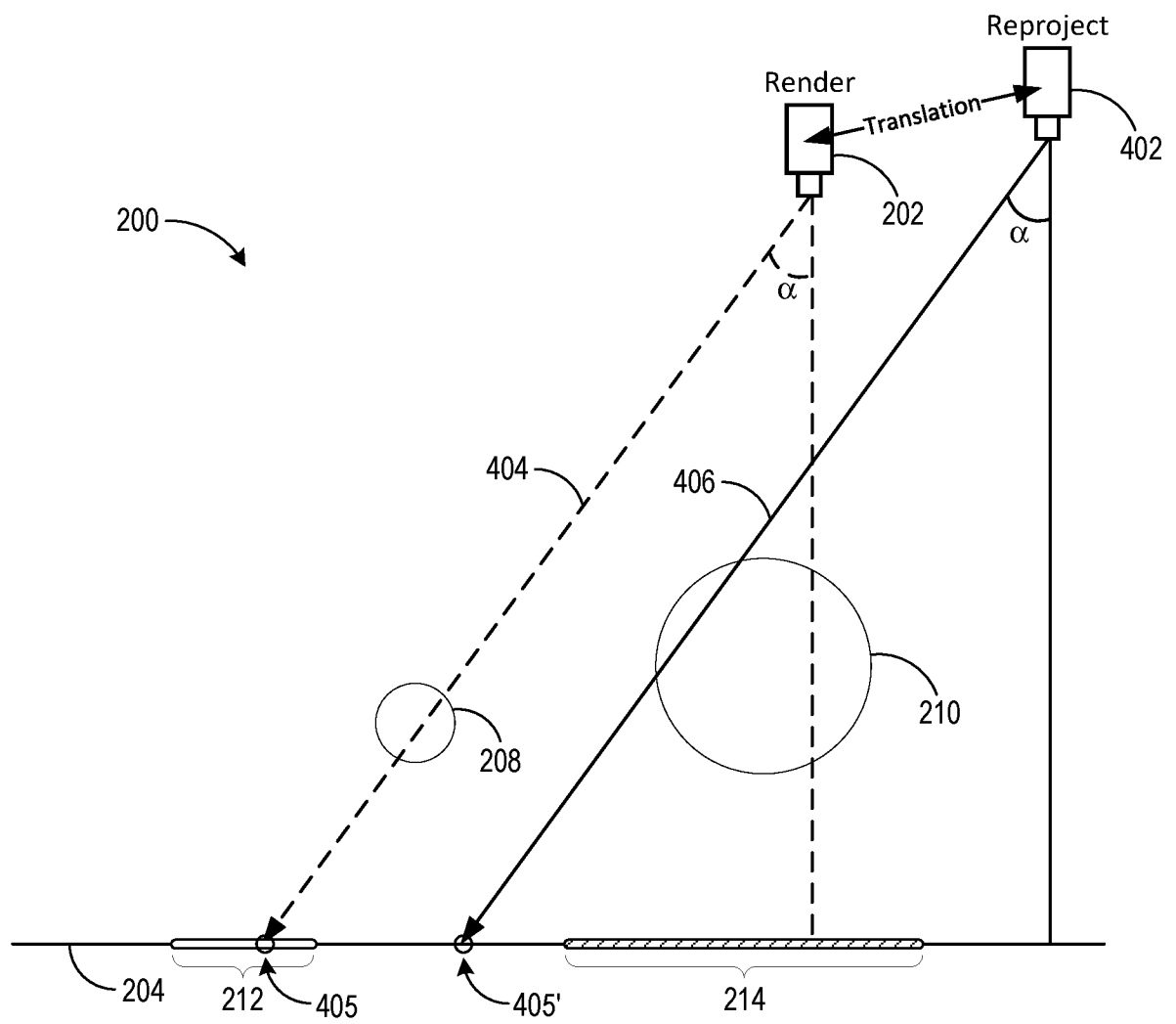


FIG. 4

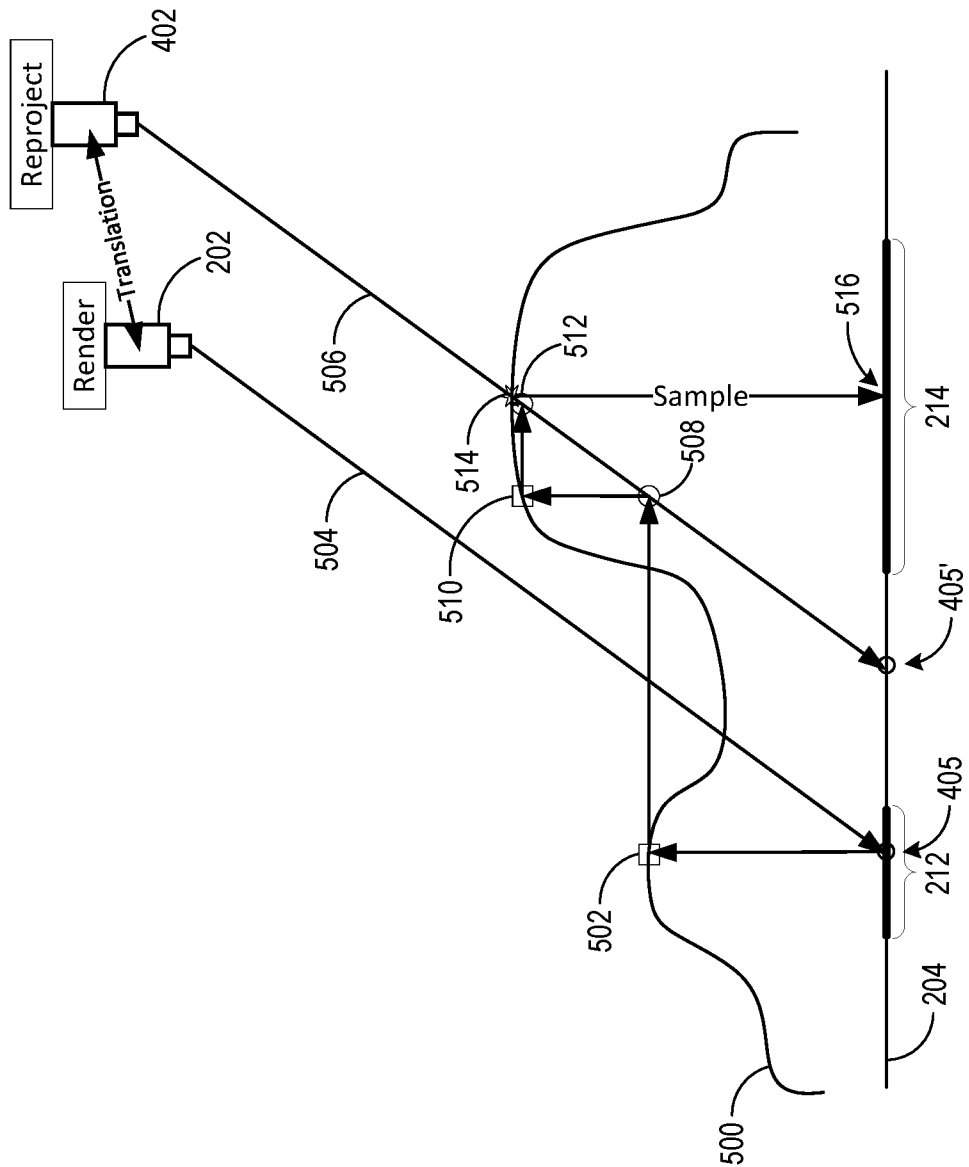


FIG. 5

6/9

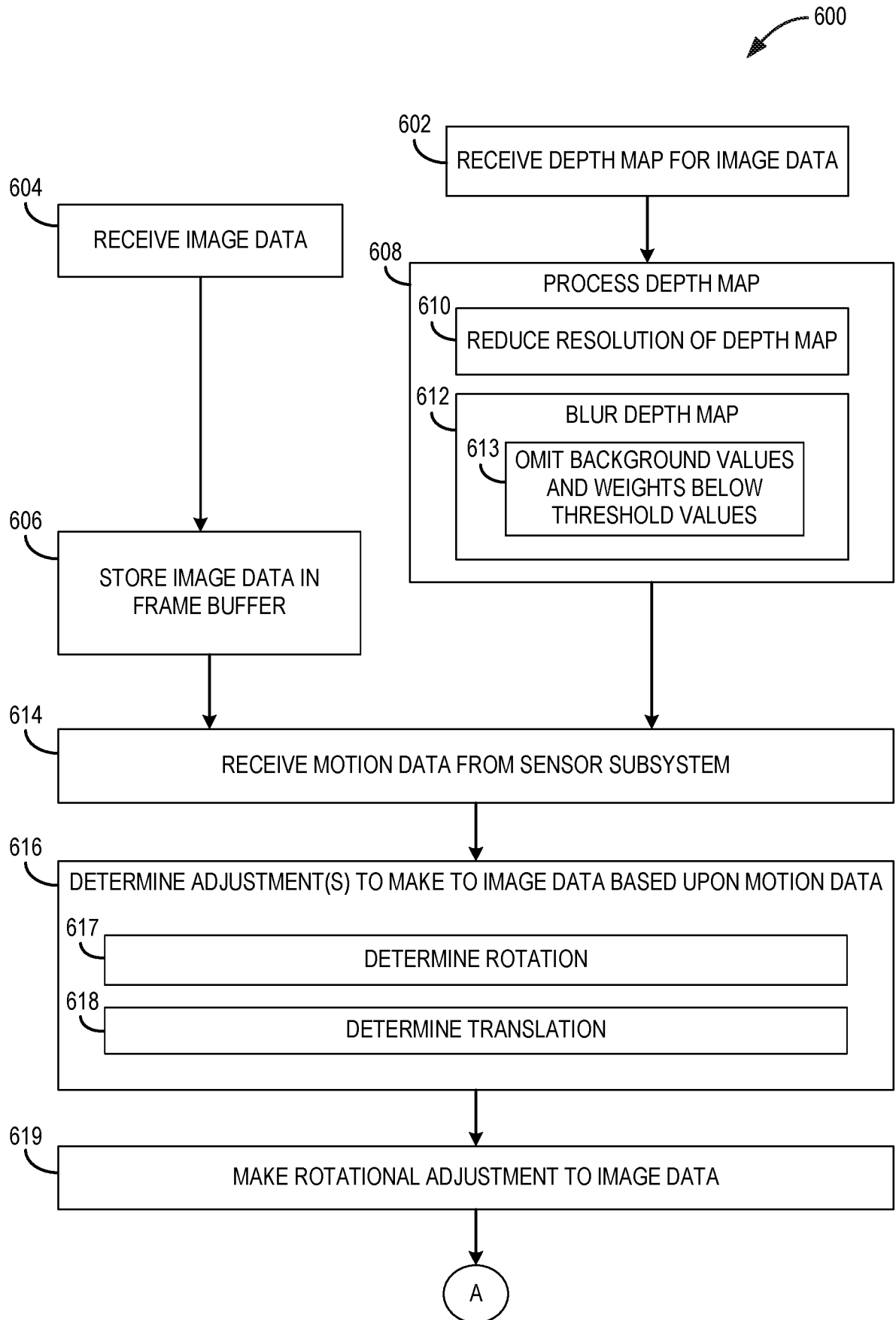


FIG. 6A

7/9

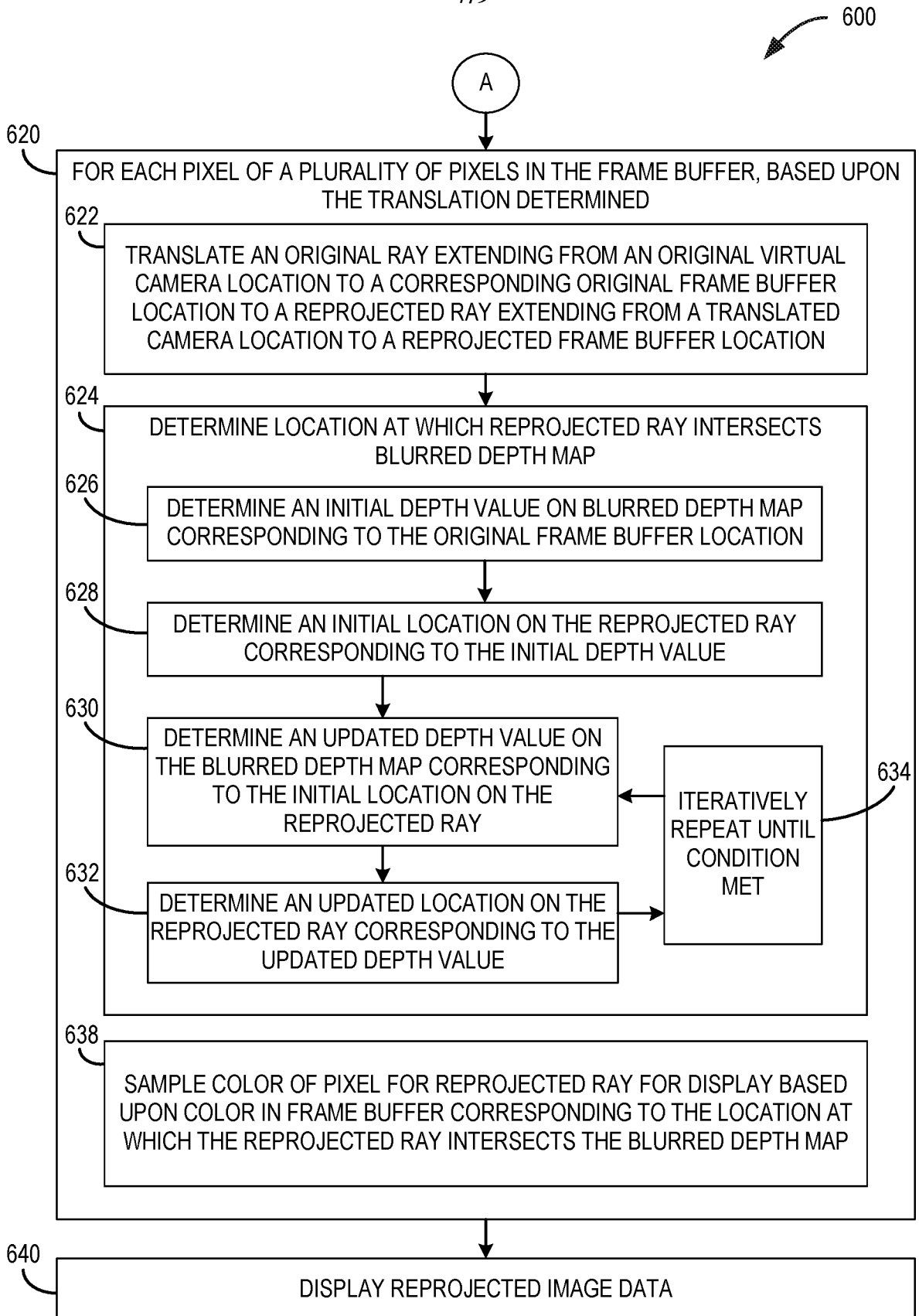


FIG. 6B

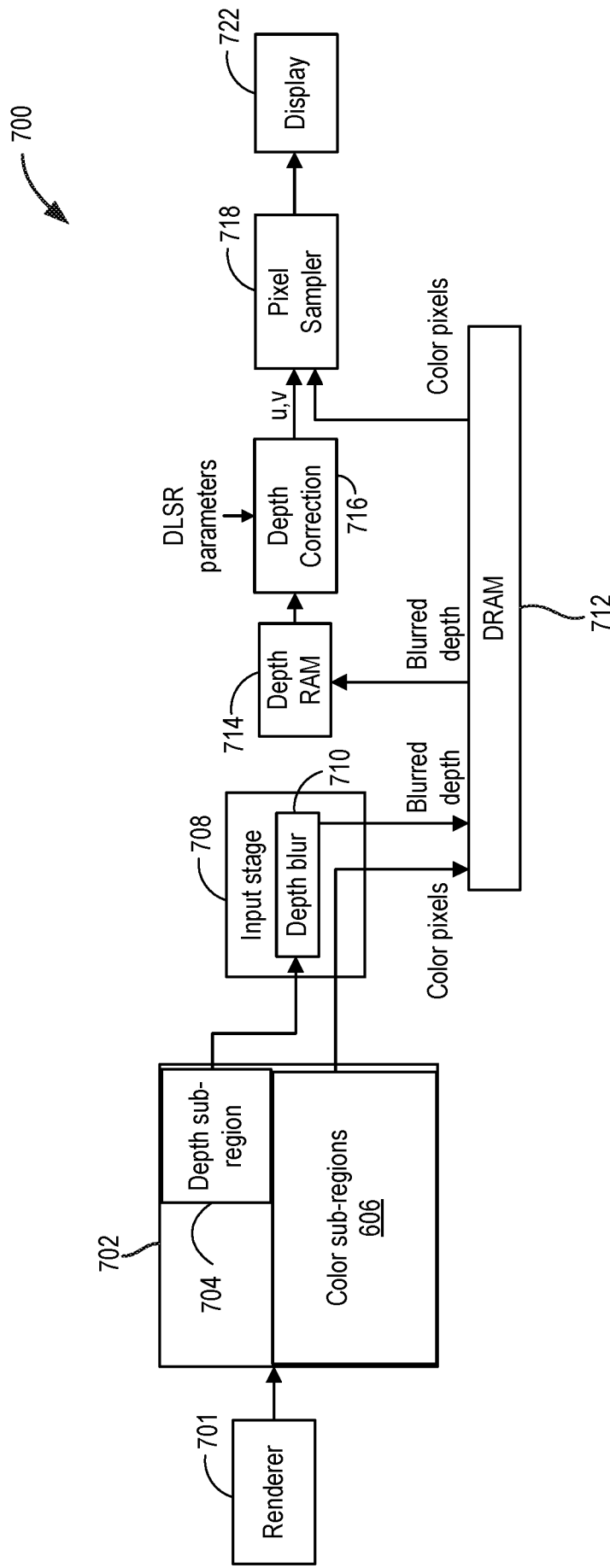


FIG. 7

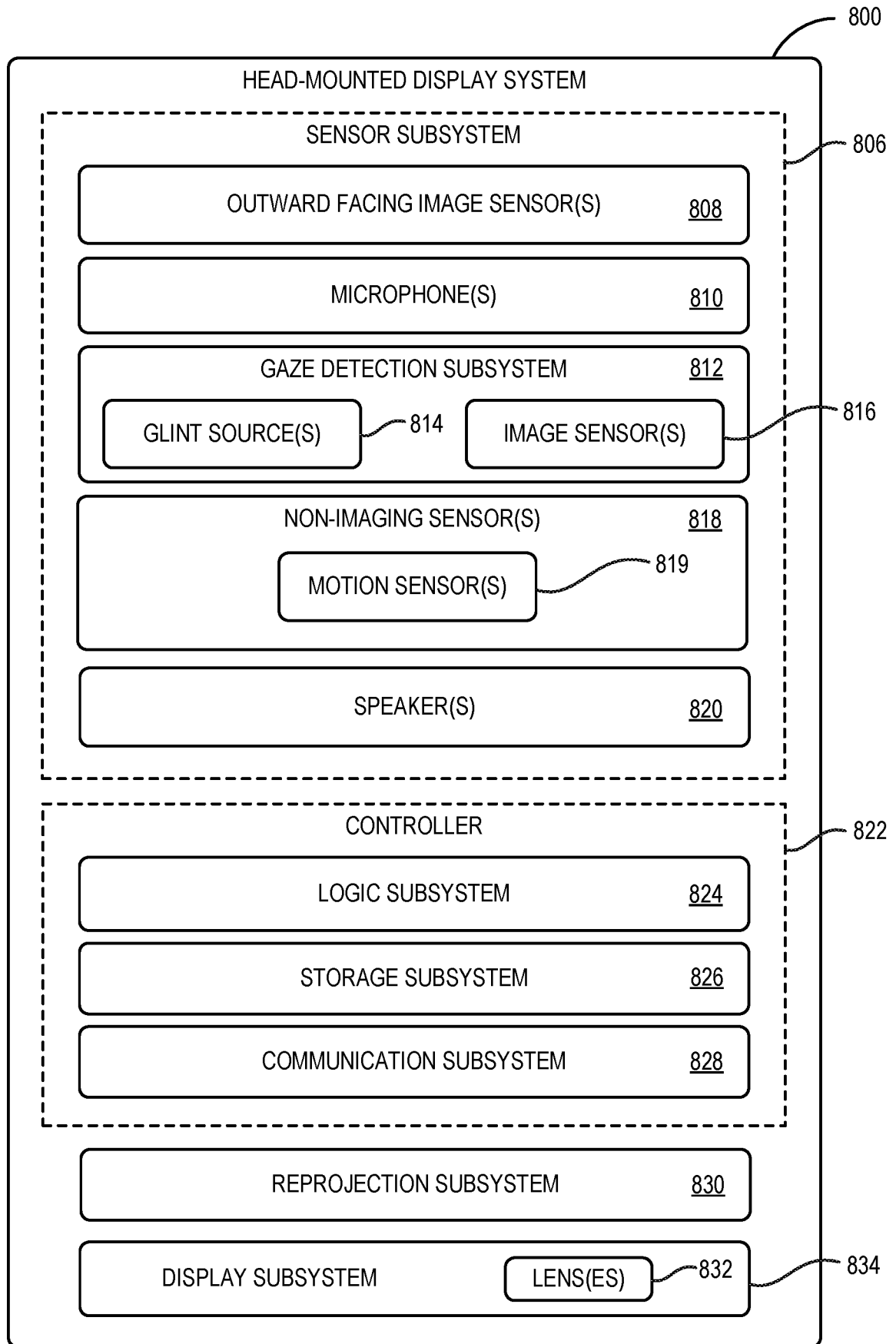


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2017/036927

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06T19/00

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2015/029218 A1 (WILLIAMS OLIVER MICHAEL CHRISTIAN [US] ET AL) 29 January 2015 (2015-01-29) abstract	1-15
A,P	----- EP 3 051 525 A1 (SONY COMP ENTERTAINMENT EUROPE [GB]) 3 August 2016 (2016-08-03) abstract	1-15
A,P	----- WO 2017/087088 A1 (QUALCOMM INC [US]) 26 May 2017 (2017-05-26) abstract	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

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

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

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

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

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

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

22 August 2017

Date of mailing of the international search report

29/08/2017

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

dos Santos, Luís

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/036927

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015029218 A1	29-01-2015	CN 105474273 A	06-04-2016
		EP 3025312 A1	01-06-2016
		KR 20160033763 A	28-03-2016
		US 2015029218 A1	29-01-2015
		US 2016343172 A1	24-11-2016
		WO 2015013347 A1	29-01-2015

EP 3051525 A1	03-08-2016	EP 3051525 A1	03-08-2016
		GB 2534847 A	10-08-2016
		US 2016216518 A1	28-07-2016

WO 2017087088 A1	26-05-2017	US 2017148168 A1	25-05-2017
		WO 2017087088 A1	26-05-2017
