US 20170115488A1

(19) United States

(12) Patent Application Publication

Ambrus et al.

(10) Pub. No.: US 2017/0115488 A1

(43) Pub. Date: Apr. 27, 2017

(54) **REMOTE RENDERING FOR VIRTUAL IMAGES**

(71) Applicant: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(72) Inventors: **Anthony Ambrus**, Seattle, WA (US); **Dan Kroymann**, Kirkland, WA (US); **Cameron Quinn Egbert**, Sammamish, WA (US); **Jeffrey Wallace McGlynn**, Seattle, WA (US); **Michael Ey**, Redmond, WA (US)

(21) Appl. No.: **14/923,082**

(22) Filed: **Oct. 26, 2015**

Publication Classification

(51) **Int. Cl.**

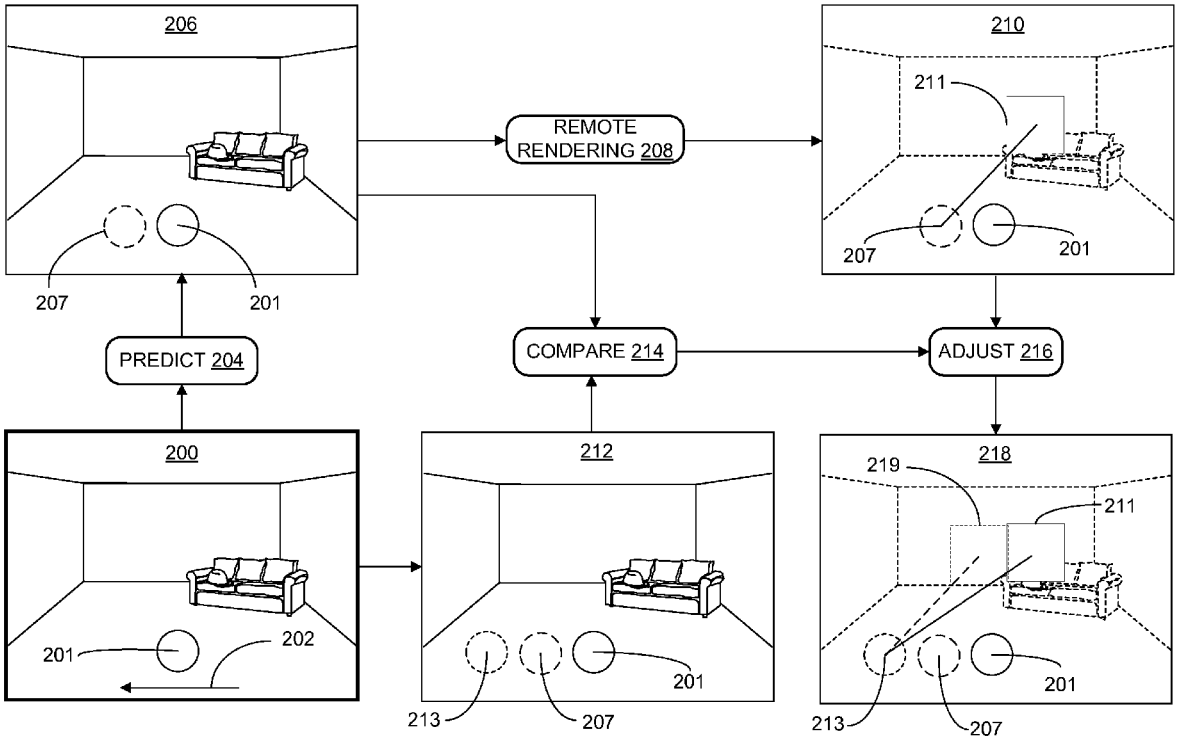
G02B 27/01	(2006.01)
G06F 3/01	(2006.01)
G06T 15/00	(2006.01)
G06T 19/00	(2006.01)
G06T 11/60	(2006.01)

(52) **U.S. Cl.**

CPC ***G02B 27/0172*** (2013.01); ***G06T 19/006*** (2013.01); ***G06T 11/60*** (2013.01); ***G06T 15/005*** (2013.01); ***G06F 3/012*** (2013.01); *G02B 2027/0187* (2013.01); *G02B 2027/014* (2013.01); *G02B 2027/0147* (2013.01)

(57) **ABSTRACT**

One or more sensors of a virtual reality device track a pose of the virtual reality device. The virtual reality device requests a virtual image having a perspective corresponding to a future pose from a remote computer. After receiving the requested virtual image, the virtual reality device adjusts the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device. Then, a virtual reality display displays the adjusted virtual image.



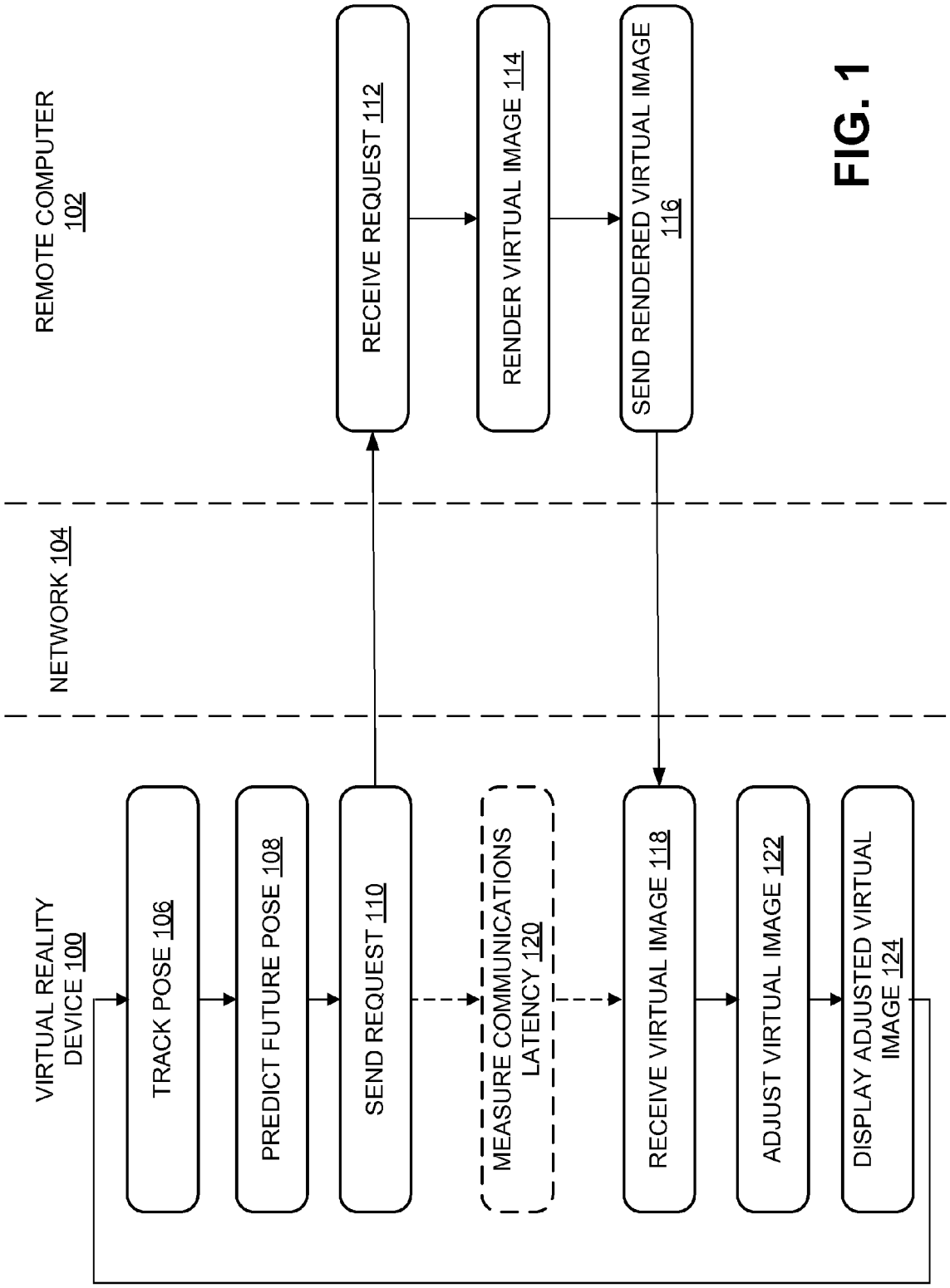

VIRTUAL REALITY DEVICE 100
NETWORK 104
REMOTE COMPUTER 102
TRACK POSE 106
PREDICT FUTURE POSE 108
SEND REQUEST 110
MEASURE COMMUNICATIONS LATENCY 120
RECEIVE REQUEST 112
RENDER VIRTUAL IMAGE 114
SEND RENDERED VIRTUAL IMAGE 116
RECEIVE VIRTUAL IMAGE 118
ADJUST VIRTUAL IMAGE 122
DISPLAY ADJUSTED VIRTUAL IMAGE 124

FIG. 1

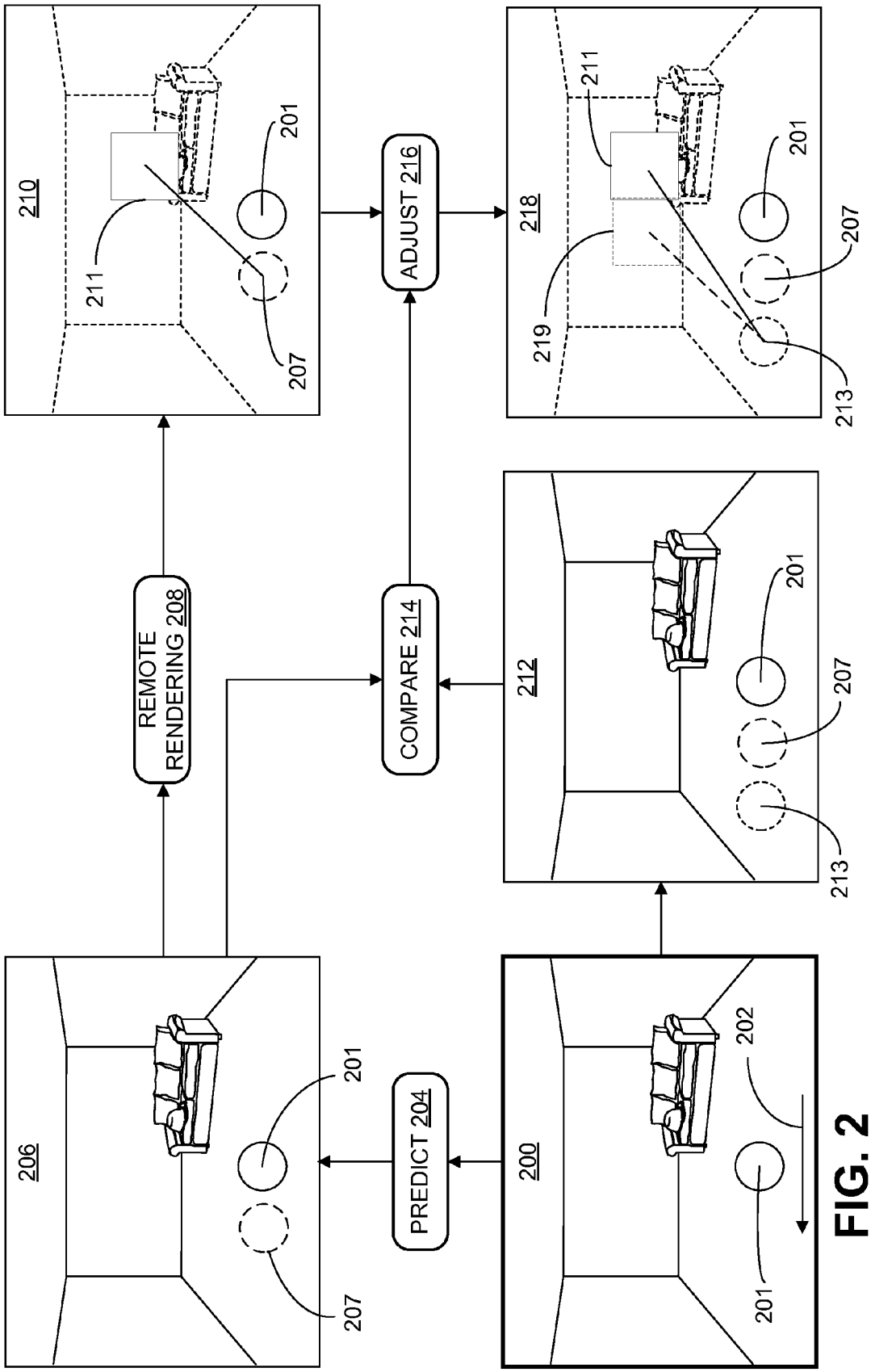

206
201
207
REMOTE RENDERING 208
210
211
201
207
PREDICT 204
200
202
201
COMPARE 214
212
201
207
213
ADJUST 216
218
211
219
201
207
213

FIG. 2

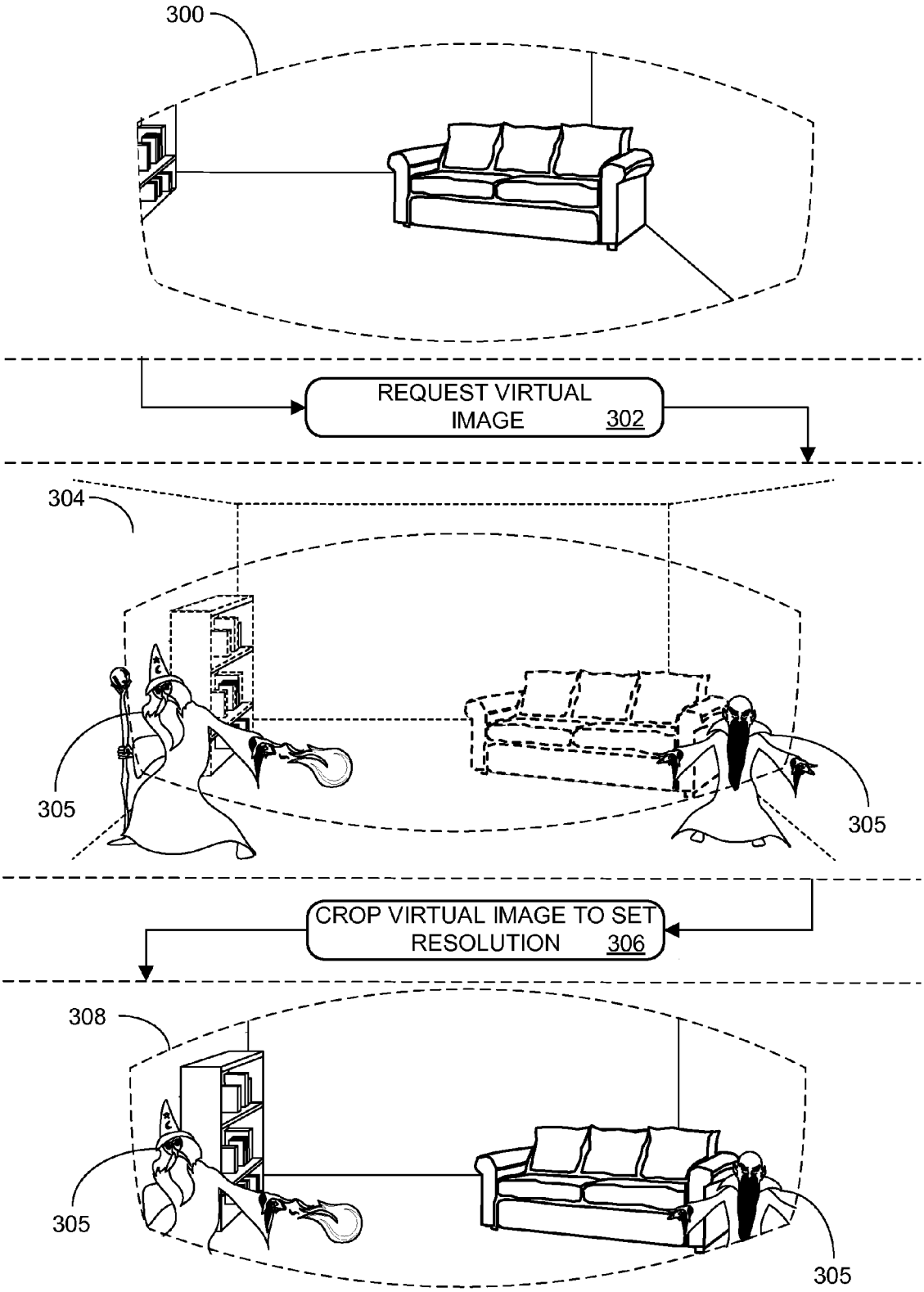

300
REQUEST VIRTUAL IMAGE 302
304
305
305
CROP VIRTUAL IMAGE TO SET RESOLUTION 306
308
305
305

FIG. 3

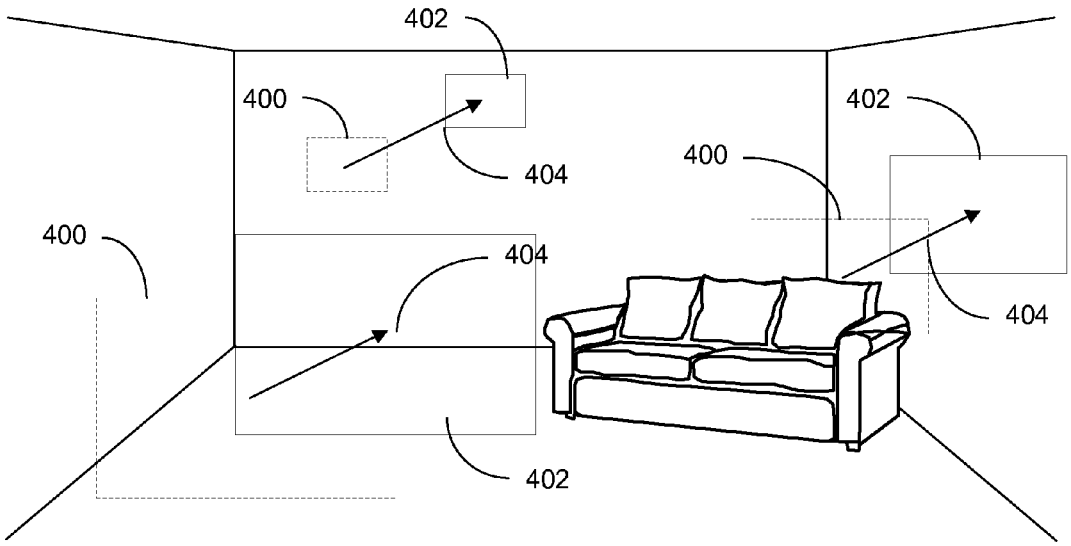

402
400
404
402
400
400
404
404
402

FIG. 4A

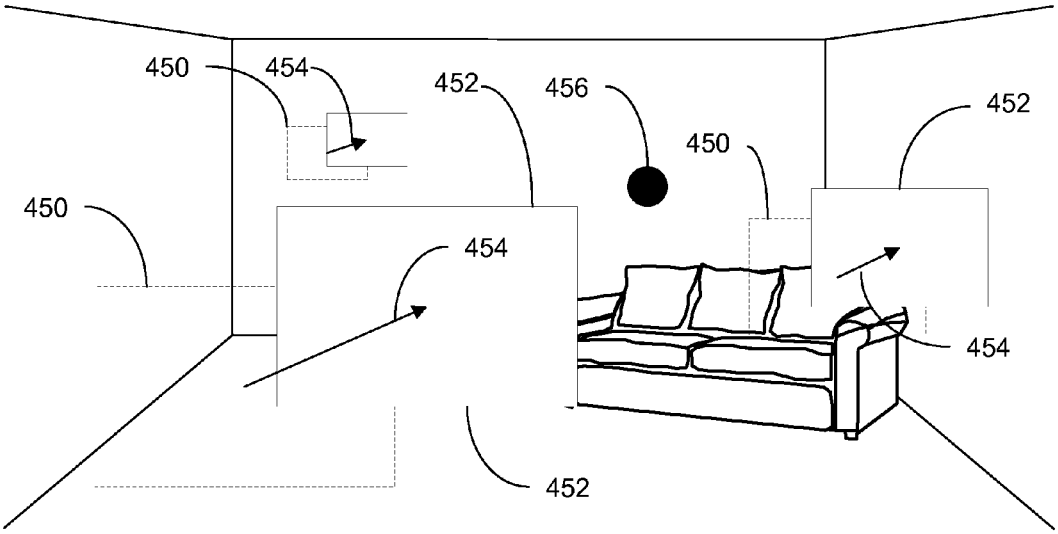

450
454
452
456
452
450
450
454
454
452

FIG. 4B

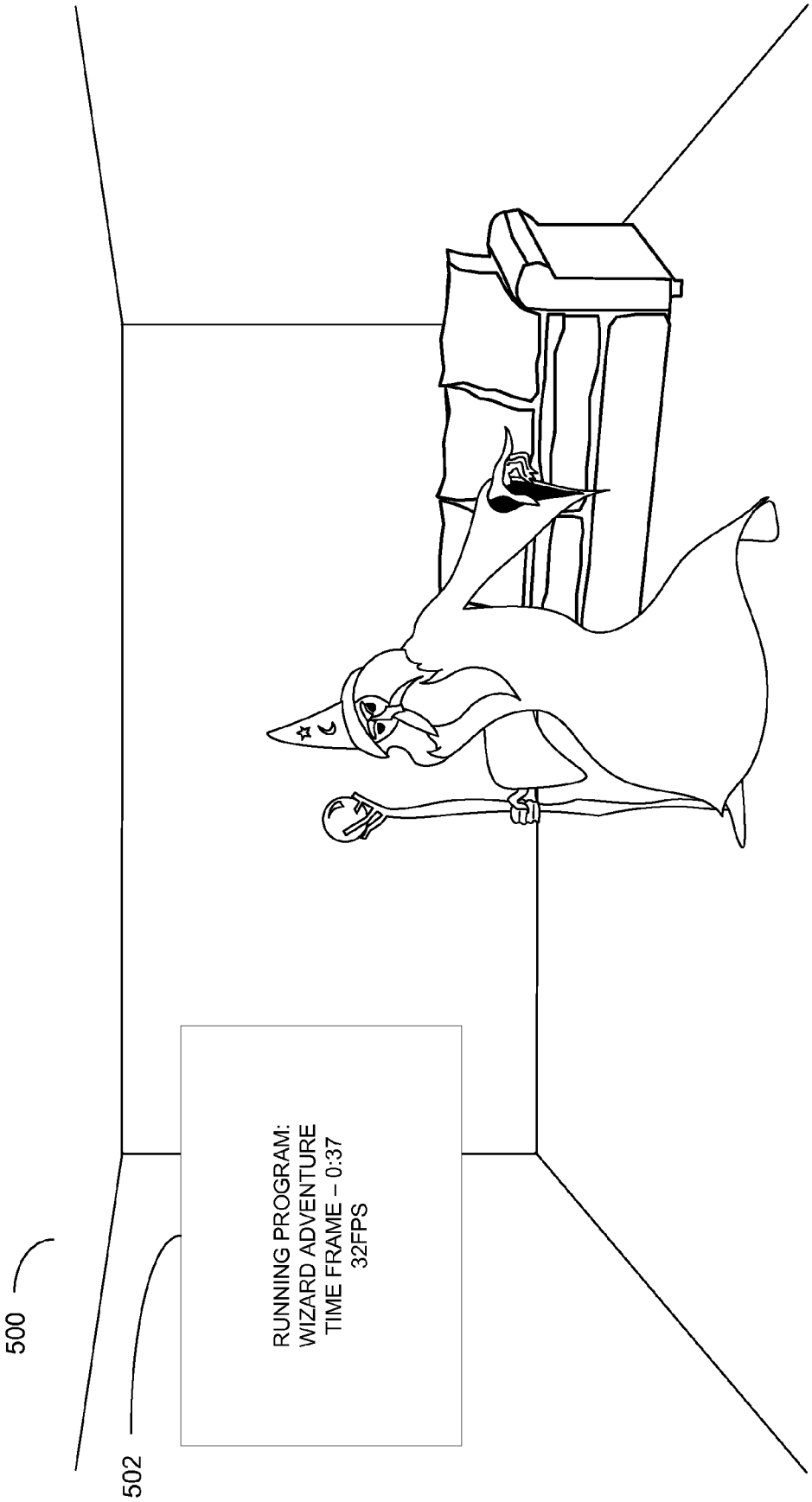

500
502
RUNNING PROGRAM:
WIZARD ADVENTURE
TIME FRAME – 0:37
32FPS

FIG. 5

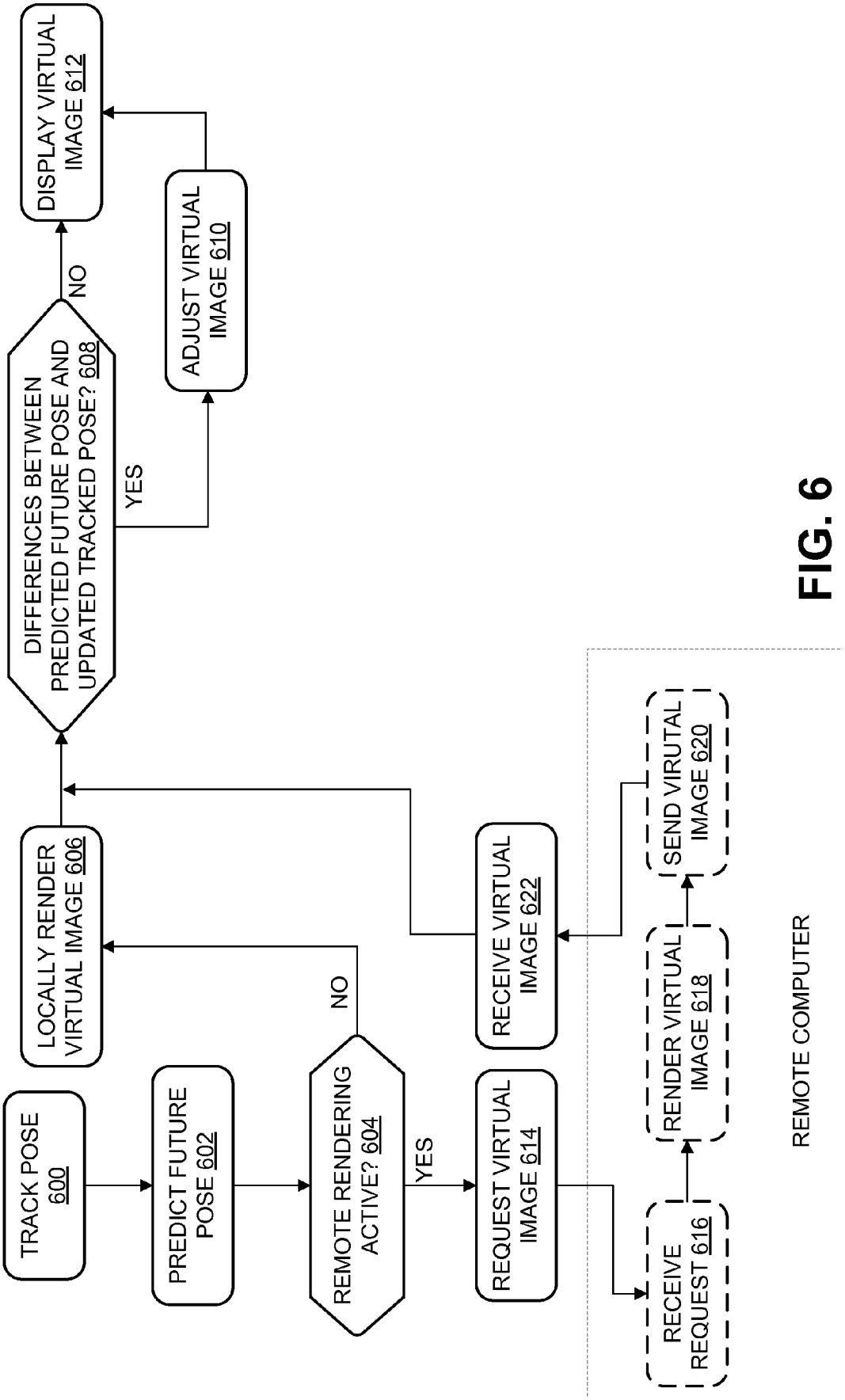

TRACK POSE 600
PREDICT FUTURE POSE 602
REMOTE RENDERING ACTIVE? 604
NO
YES
LOCALLY RENDER VIRTUAL IMAGE 606
REQUEST VIRTUAL IMAGE 614
RECEIVE VIRTUAL IMAGE 622
DIFFERENCES BETWEEN PREDICTED FUTURE POSE AND UPDATED TRACKED POSE? 608
NO
YES
DISPLAY VIRTUAL IMAGE 612
ADJUST VIRTUAL IMAGE 610
RECEIVE REQUEST 616
RENDER VIRTUAL IMAGE 618
SEND VIRUTAL IMAGE 620
REMOTE COMPUTER

FIG. 6

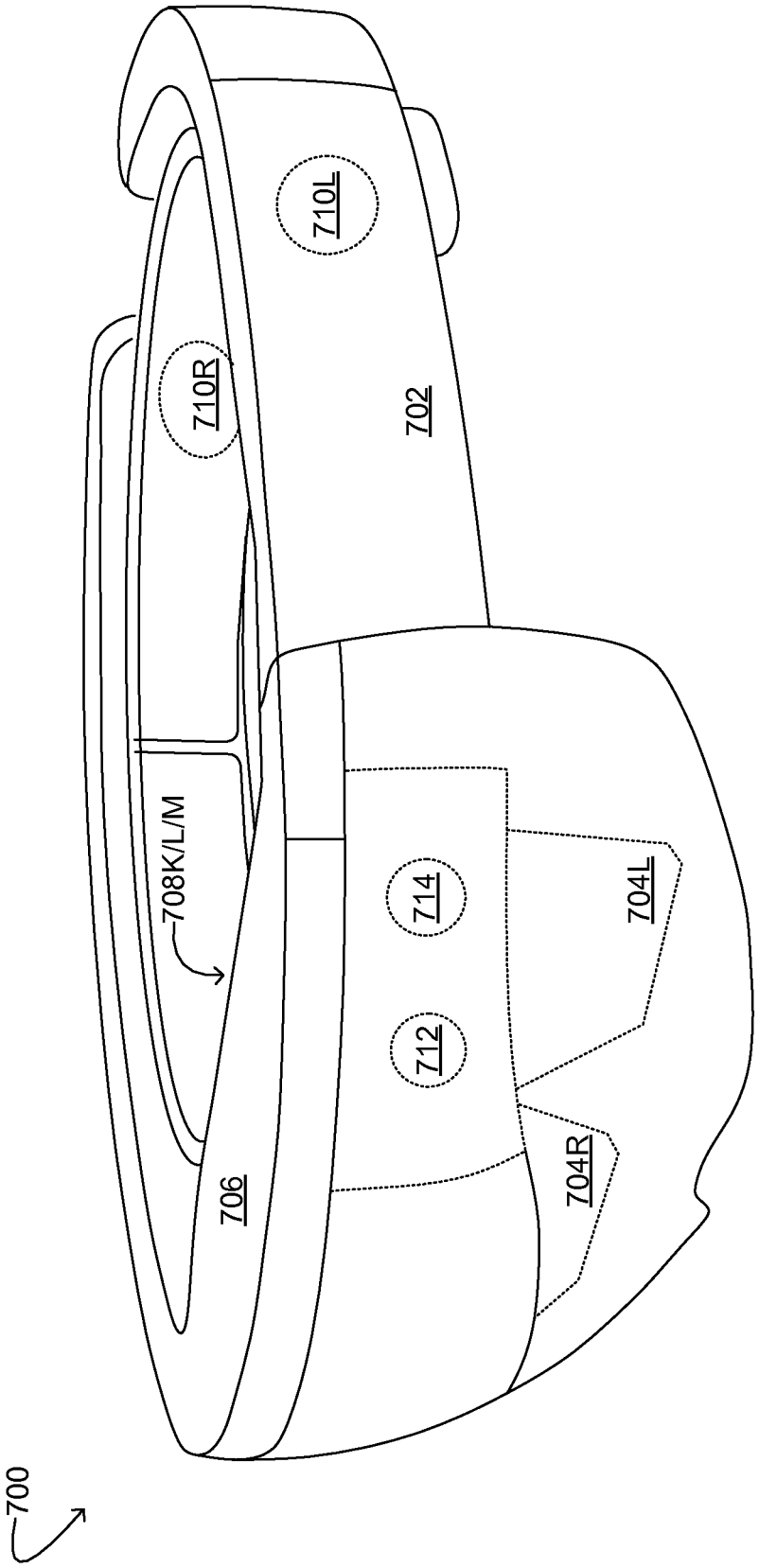


FIG. 7

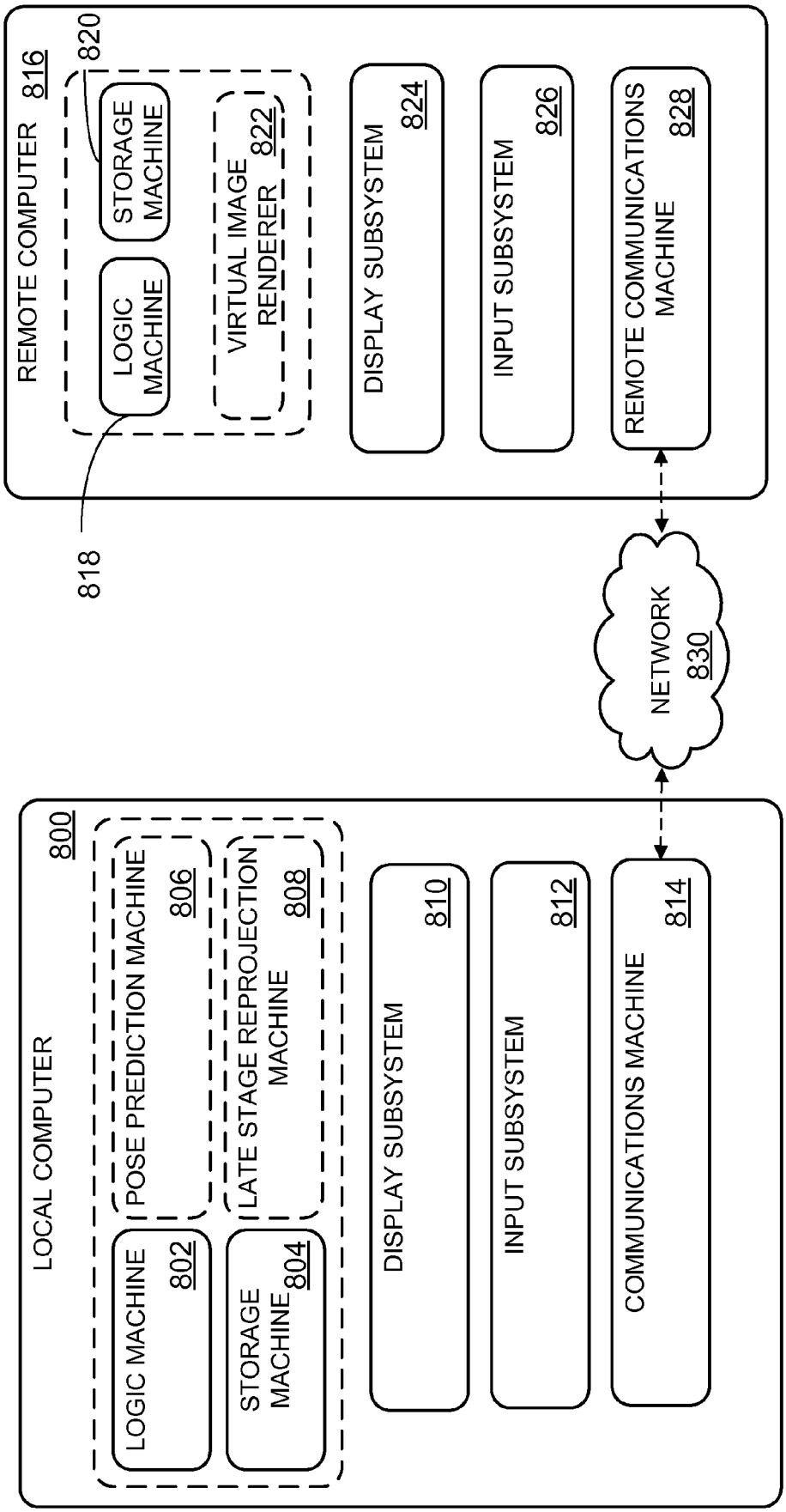

REMOTE COMPUTER 816
820
LOGIC MACHINE
STORAGE MACHINE
VIRTUAL IMAGE RENDERER 822
818
DISPLAY SUBSYSTEM 824
INPUT SUBSYSTEM 826
REMOTE COMMUNICATIONS MACHINE 828
NETWORK 830
LOCAL COMPUTER 800
POSE PREDICTION MACHINE 806
LOGIC MACHINE 802
LATE STAGE REPROJECTION MACHINE 808
STORAGE MACHINE 804
DISPLAY SUBSYSTEM 810
INPUT SUBSYSTEM 812
COMMUNICATIONS MACHINE 814

FIG. 8

REMOTE RENDERING FOR VIRTUAL IMAGES

BACKGROUND

[0001] A variety of augmented and virtual reality devices exist which are usable for generating and viewing interactive virtual imagery. Such virtual imagery may include information displays, content windows, user interfaces, artificially intelligent virtual entities, 3D models, and other virtual objects. Virtual imagery may be world-locked so as to make it appear that virtual objects remain in a fixed location in the real world.

SUMMARY

[0002] 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.

[0003] One or more sensors of a virtual reality device track a pose of the virtual reality device. The virtual reality device requests a virtual image having a perspective corresponding to a future pose from a remote computer. After receiving the requested virtual image, the virtual reality device adjusts the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device. Then, a virtual reality display displays the adjusted virtual image.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. **1** is a flowchart of a process for locally requesting and displaying remotely rendered virtual images.

[0005] FIG. **2** illustrates an example process for locally requesting and displaying remotely rendered virtual images.

[0006] FIG. **3** illustrates an example method of oversizing and cropping rendered virtual images.

[0007] FIGS. **4**A and **4**B illustrate shifting of virtual images responsive to detected motion.

[0008] FIG. **5** schematically shows an example of a field of view of a virtual reality device.

[0009] FIG. **6** illustrates an example flow diagram for a method of rendering and displaying virtual images.

[0010] FIG. **7** schematically shows an example of a wearable computing device.

[0011] FIG. **8** schematically shows an example computing system.

DETAILED DESCRIPTION

[0012] When using a virtual reality device, it may sometimes be desirable to outsource some or all rendering of virtual images to a remote computer. In such cases, a virtual reality device may send to a remote computer a request for a virtual image. The remote computer may then render the virtual image, and send it to the virtual reality device for display. Remote rendering of virtual images may present several advantages over local rendering, as will be explained in more detail below. However, remote rendering may also introduce delay between when a virtual image is requested and when the virtual image is actually displayed, potentially resulting in a less satisfactory user experience. Virtual images displayed by a virtual reality device may be substantially "world-locked," such that the virtual images appear to occupy specific positions in physical space which do not change as the perspective of the virtual reality device changes. In order to maintain this effect, rendered virtual imagery must appear to update substantially simultaneously as a user moves. For example, as a user moves to the left, a rendered virtual image may need to simultaneously shift a proportional distance to the right, so as to appear not to move relative to the real world. Delays in image rendering introduced by remote rendering can compromise this effect, resulting in "jittery" or "swimmy" images. As discussed below, these effects can be lessened by recognizing the time necessary to request and receive a remotely-rendered virtual image, and requesting the virtual image for a corresponding time in the future.

[0013] Remote rendering of virtual images may be useful in any number of potential scenarios. For example, the virtual images to be rendered may be complex, and require substantial computer resources for rendering and/or manipulation. In such cases, rendering of virtual images may be outsourced to a remote computer with greater image rendering and/or general processing capabilities than the virtual reality device.

[0014] In another example, remote rendering may be useful for reducing iteration time and/or testing code changes during software development. An application developer may use a remote computer to write a software application executable by a virtual reality device. During application development, the developer may test aspects of the software application by executing it on a remote computer configured to send rendered imagery to a virtual reality device. The remote computer may have one or more tools/features useful for application development which are not available/easily accessible on the virtual reality device. The developer may use the virtual reality device during this process, evaluating and/or interacting with the rendered virtual imagery. This may allow the application developer to test software changes substantially in real time, eliminating the need to compile and install new software builds on the virtual reality device before testing can occur. In some embodiments, the developer may be able to make changes to the software application by interacting with one or more virtual elements via the virtual reality device, thereby affecting the software running on the remote computer.

[0015] FIG. **1** illustrates a flowchart of an example process for remotely rendering virtual images while lessening lag/latency perceived by a user. Virtual reality device **100** may be any device suitable for requesting and displaying virtual images, including but not limited to: virtual reality and/or augmented reality-glasses, -goggles, -contact lenses, -helmets, -heads-up displays (HUDs), -hologram generators, -projectors, and/or any other suitable devices. Virtual reality device **100** may utilize an opaque display, thereby obscuring a wearer's view of their real-world environment in favor of a rendered virtual reality (VR) environment. Alternately, virtual reality device **100** may utilize a transparent or see-thru display, configured to present an augmented reality (AR) environment, in which virtual objects are viewable along with real-world objects. Remote computer **102** may include any suitable computer system, including but not limited to one or more a: desktop computer, laptop computer, smartphone, tablet computer, wearable computer, and/

or server computer. Further, network **104** may be any suitable computer network including, for example, the Internet, and/or any local wired/wireless computer networks with bandwidth sufficient for transmitting rendered virtual images. Virtual reality device **100** and remote computer **102** may occupy separate network nodes on network **104**. Virtual reality device **100** and/or remote computer **102** may be implemented as one or more computer systems as described below with respect to FIG. **8**.

[0016] At **106**, one or more sensors track a pose of the virtual reality device. A pose may be defined as a set of variables establishing the virtual reality device's position and/or orientation at a specific moment in time. Such variables may include the device's current three-dimensional coordinates (e.g., X, Y, and Z coordinates), comprising three degrees of freedom (3DOF). Further, the one or more sensors may track the virtual reality device's current pitch, roll, and yaw, providing an additional three degrees of freedom (3DOF), for a combined six degrees of freedom (6DOF). In some embodiments, the one or more sensors are configured to detect a motion of the virtual reality device in one or more dimensions. For example, if a user of a wearable virtual reality device walks forward, moves his head, leans backward, and/or moves in any other way, the one or more sensors may detect the motion and continuously update the tracked pose. The one or more sensors may comprise one or more gyroscopes, accelerometers, proximity sensors, optical sensors, infrared sensors, ultrasonic sensors, microphones, global positioning system (GPS) receivers, cameras, thermometers, barometers, and/or any other suitable sensors.

[0017] At **108**, a pose prediction machine of virtual reality device **100** predicts a future pose of the virtual reality device at a future time. The future time may be any suitable time in the future, and may be established according to one or more criteria, as described in further detail below. In particular, the future time may be separated from the current time by a buffer period, and the size of the buffer period may depend upon one or more factors. The future pose is the pose prediction machine's prediction of what the pose of virtual reality device **100** will be at or near the future time, according to the device's current pose, movement, and/or other factors, as will be explained below.

[0018] In order to predict the future pose, the pose prediction machine may make use of the virtual reality device's current pose and movement, as tracked by the one or more sensors. Then, the pose prediction machine may extrapolate the detected motion forward from the current time to the future time, thus generating a reasonable approximation of the position and/or orientation the virtual reality device **100** is likely to have at or near the future time. For example, if the one or more sensors determine that, at a current time, virtual reality device **100** is moving slowly to the left, then the pose prediction machine may predict a future pose of the virtual reality device by extrapolating the device's leftward motion until the future time.

[0019] In some embodiments, other factors may be considered when predicting the future pose of virtual reality device **100**. For example, when virtual reality device **100** is worn by a user, the future pose may be predicted by extrapolating the detected motion forward with biomechanical constraints. This may include determining that a detected motion is consistent with a predictable human movement such as nodding, head turning, bending at the waist, jumping, etc., and using this information to more accurately predict the future motion and resulting pose. For example, the pose prediction machine may determine that a detected motion is consistent with a user turning his head. Based on the duration and speed of the detected head turn, the pose prediction machine may determine a likely future pose extrapolated from the current pose and detected head-turning motion.

[0020] Further, the pose prediction machine may predict the future pose by extrapolating the detected motion forward with environmental constraints. This may include identifying one or more real-world/virtual objects, features, elements, and/or other points of interest that a user is likely to focus on. For example, if the pose prediction machine determines that a user is turning his head to look at a nearby television set, the pose prediction machine may predict that the user will focus on the television set, and therefore predict the future pose according to an assumption that the detected motion will stop once the television set is in the center of the user's field of view. Similar predictions can be made for virtual objects. For example, the pose prediction machine may determine that a user is tilting his head upward toward a world-locked virtual monitor, and predict the future pose accordingly. Alternatively, the pose prediction machine may determine that there are no notable points of interest in the path of the detected motion, and use this information when predicting the future pose.

[0021] In the illustrated example, the pose prediction machine is shown as being part of the virtual reality device. However, in some embodiments, the pose prediction machine may be part of the remote computer. For example, the virtual reality device may send to the remote computer data from the one or more sensors, and the pose prediction machine of the remote computer may predict the future pose of the virtual reality device based on the sensor data, as described above.

[0022] At **110**, a communications machine of the virtual reality device sends to the remote computer a request for a virtual image. Specifically, the virtual reality device requests a virtual image having a perspective corresponding to the future pose of the virtual reality device. In other words, after the pose prediction machine predicts the future pose, the communications machine requests that the remote computer render one or more virtual images from the predicted perspective of the virtual reality device.

[0023] In some embodiments, the request may specify one or more virtual images consistent with one or more points of view. For example, the virtual reality device may request two virtual images with slightly different perspectives, one to be displayed for a user's left eye, and one to be displayed for a user's right eye. Additional perspectives may also be requested, for example from the point of view of one or more cameras attached to the virtual reality device and/or present in the real-world environment. Further, the request may specify that rendered virtual images have a certain resolution and/or size. This resolution may match a resolution of a display device by which the virtual image will be displayed, or the requested resolution may be different from (e.g., greater or less than) the resolution of the display device, as will be explained in further detail below with respect to FIG. **3**.

[0024] Virtual reality device **100** sends the request to remote computer **102** over network **104**. As described above, network **104** may be virtually any suitable computer network including, for example, the Internet and/or a local area

network (LAN). As such, when communicating over the computer network, each of the virtual reality device and the remote computer occupy a different network node. The virtual reality device and the remote computer may be separated by virtually any physical distance, provided they are able to communicate over the computer network.

[0025] At **112**, a remote communications machine of the remote computer receives from a virtual reality device a request for a virtual image. As described above, the request is for one or more virtual images having a perspective corresponding to a predicted future pose of the virtual reality device.

[0026] At **114**, a virtual image renderer of the remote computer renders the virtual image for the virtual reality device as specified by the request.

[0027] At **116**, the remote communications machine of the remote computer sends to the virtual reality device the rendered virtual image. In some embodiments, the virtual image may be compressed and/or otherwise modified prior to sending. This may be done, for example, responsive to a transmission speed and/or maximum bandwidth of the computer network.

[0028] In some embodiments, information pertaining to the predicted future pose may be attached to the rendered virtual image before it is sent to the virtual reality device. This may permit a virtual reality device to determine which rendered virtual images correspond to which predicted future poses. Further, in the case where the pose prediction machine is located at the remote computer, sending pose information with the rendered virtual image may allow the virtual reality device to determine the future pose for which the virtual image was rendered. This information may be linked to the rendered virtual images in any suitable way. For example, the pose information may simply be appended to each network packet including rendered virtual imagery. Alternately, pose information may be encoded within each network packet as pixel data associated with the rendered virtual imagery. Pixels including pose information may be cleared to black once decoded by the virtual reality device. Further, pose information may be indexed according to integer identifiers and tracked by one or more computer devices via, for example, a send and lookup method.

[0029] At **118**, the communications machine of the virtual reality device **100** receives from the remote computer the virtual image. Upon receipt of the virtual image, the virtual reality device may perform one or more processing/analysis steps in order to, for example, decompress the received virtual image and/or otherwise modify it before additional operations are performed.

[0030] At **120**, after the rendered virtual image has been received, the communications machine of the virtual reality device **100** determines a communications latency of the computer network by measuring an elapsed time between the request being sent and the virtual image being received. This communications latency may be used in future decisions regarding how virtual images should be rendered. For example, as described above, the pose prediction machine predicts the future pose of the virtual reality device at a future time, and this future time is separated from the current time by a buffer period. In some embodiments, the pose prediction machine may be configured to dynamically adjust the buffer period by dynamically increasing the buffer period with increasing communications latency and dynamically decreasing the buffer period with decreasing communications latency. A high communications latency may indicate relatively slow communication between the virtual reality device and the remote computer over the computer network. As such, the pose prediction machine may need to predict the future pose for a future time which is relatively far in the future, in order to account for the time it will take before a requested virtual image has been remotely rendered and received. However, when the communications latency is low, the pose prediction machine may be able to predict the future pose for a future time which is relatively closer to the current time, thereby increasing the accuracy of the prediction. Further, the current framerate for display of virtual images may be adjusted based on communications latency, and the communications latency may be taken into account when establishing a resolution of virtual images. For example, in cases of high communications latency, framerate and virtual image resolution may be dynamically reduced, in order to minimize network congestion related latency/delays.

[0031] At **122**, a late stage reprojection machine of the virtual reality device **100** adjusts the virtual image before it is displayed. Such late stage reprojection can lessen any noticeable effects resulting from the pose prediction engine predicting the wrong pose. Specifically, the rendered virtual image, which has a perspective corresponding to the predicted future pose, may be adjusted to have an updated perspective corresponding to an updated tracked pose of the virtual reality device. For example, before a virtual image is requested, the one or more sensors may determine that the virtual reality device is moving to the left, and the pose prediction machine may use this information to predict a future pose where the virtual reality device has moved somewhat to the left relative to the current pose. Then, a virtual image having this perspective may be requested. However, after requesting the virtual image, the one or more sensors may determine that the leftward motion has stopped sooner than expected, and as a result, the rendered virtual image will not match the actual pose of the device at the future time. As such, if the rendered virtual image was displayed to a user unmodified, virtual objects within the virtual image may appear to suddenly move away from their world-locked positions. To solve this problem, the late stage reprojection machine may adjust the image by shifting the virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose. Given the above example, the late stage reprojection machine may slightly shift the virtual image in order to compensate for the change in the virtual reality device's motion, ensuring that, from the user's perspective, world-locked virtual objects remain in their designated positions.

[0032] In some embodiments, the updated tracked pose used by the late stage reprojection machine will be the last possible pose which can be used before the virtual image is displayed, so as to ensure that the virtual image is adjusted according to the most recent possible information. The updated tracked pose used to adjust the rendered virtual image may represent the virtual reality device's position and/or orientation at a time either before the virtual image has been received, or after the virtual image has been received and before it is displayed. In some embodiments, the pose prediction machine may begin continuously comparing the most recent updated tracked pose to the predicted future pose as soon as the virtual image is requested,

dynamically determining which adjustments need to be made to the rendered virtual image before it is displayed.

[0033] At **124**, a virtual reality display of the virtual reality device **100** displays the adjusted virtual image at or near the future time. Once the rendered virtual image has been received and adjusted by the late stage reprojection machine, it is displayed at or near the future time using the virtual reality display. Virtual reality device **100** may incorporate virtually any type of display usable for displaying rendered virtual imagery.

[0034] Once the adjusted virtual image is displayed, the process described above repeats. The pose prediction machine uses the new current pose to predict a new future pose, and the communications machine requests a new rendered image for the new future pose. In some embodiments, the process described above is continuously repeating, such that at any given moment in time, multiple steps of the process are being carried out simultaneously. For example, at the same time a virtual image is being displayed by the virtual reality display, new future poses are being predicted, and new virtual images are being requested, rendered, and adjusted.

[0035] FIG. **2** illustrates aspects of the process outlined above with respect to FIG. **1**. First, at **200**, the one or more sensors determine a current pose **201** of a virtual reality device. The current 6DOF pose **201** is simplified as a circle in FIG. **2**. The one or more sensors further detect a motion of the virtual reality device. In FIG. **2**, the virtual reality device is moving to the left, as indicated by movement arrow **202**.

[0036] At **204**, the pose prediction machine uses current pose **201** and the detected movement indicated by arrow **202** to predict a future pose of the virtual reality device. Such a prediction may be made as described above with respect to FIG. **1**. For example, the pose prediction machine may predict the future pose by extrapolating the detected motion forward from the current time to the future time. This extrapolation may further comprise extrapolating the detected motion forward with biomechanical and/or environmental constraints.

[0037] Next, at **206**, FIG. **2** shows current pose **201** of the virtual reality device, and also shows a predicted future 6DOF pose **207** of the virtual reality device, simplified as a dashed circle. Specifically, predicted future pose **207** is shown to the left of current pose **201**, given that the motion detected by the one or more sensors was to the left.

[0038] At **208**, a virtual image is rendered at a remote computing device according to predicted future pose **207**. Virtual image rendering may occur at the remote computing device according to the process described above with respect to FIG. **1**.

[0039] Next, at **210**, FIG. **2** shows the predicted future pose **207** of the virtual reality device, and also shows a world-locked virtual object **211** present in the rendered virtual image. The real-world environment of the virtual reality device is shown in dashed lines at **210**, so as to differentiate real-world objects from world-locked virtual object **211**. A solid line connects predicted future pose **207** to virtual object **211**, indicating the angle at which the world-locked virtual object will appear relative to the virtual reality device. The angle between the virtual reality device and the world-locked virtual object is dynamically set such that the position of the world-locked virtual object appears to be the same, even as the virtual reality device moves.

[0040] Advancing in time from **200**, at **212**, FIG. **2** shows an updated tracked pose **213** of the virtual reality device, simplified as a dashed circle. Updated tracked pose **213** indicates the most recently updated position and/or orientation of the virtual reality device. Also at **212**, FIG. **2** shows pose **201**, indicating the former position and/or orientation of the virtual reality device, and predicted future pose **207**. Notably, updated tracked pose **213** is shown to the left of predicted future pose **207**, indicating that the virtual reality device has moved farther to the left than was predicted at **204**.

[0041] At **214**, after the future pose has been predicted and updated tracked pose **213** has been determined, the late stage reprojection machine compares the predicted future pose to the updated tracked pose. In this case, the late stage reprojection machine will identify differences in position and/or orientation between predicted future pose **207** and updated tracked pose **213**. As described above, in some embodiments, once the rendered virtual image has been requested, the late stage reprojection machine may continuously compare the updated tracked pose to the predicted future pose, in order to determine which if any adjustments should be made to the received virtual image before it is displayed.

[0042] Next, at **216**, the late stage reprojection machine uses the results of the comparison performed at **214** to adjust the rendered virtual image to have an updated perspective corresponding to the updated tracked pose **213** of the virtual reality device. Such adjustment may comprise shifting the virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose. In the illustrated example, the late stage reprojection machine may shift the rendered virtual image in order to account for the differences in position and/or orientation of the virtual reality device between the predicted future pose **207** and the updated tracked pose **213**.

[0043] Finally, at **218**, FIG. **2** shows the world-locked virtual object **211**, updated tracked pose **213**, former pose **201**, and predicted future pose **207**. The real-world environment of the virtual reality device is shown in dashed lines, so as to differentiate between the real-world objects and world-locked virtual object **211**. A solid line connects the updated tracked pose **213** to the world-locked virtual object **211**, indicating the angle at which the world-locked virtual object appears relative to the virtual reality device. As can be seen by comparing **218** to **210**, the angle between the virtual reality device and the world-locked virtual object **211** has changed such that the world-locked virtual object still appears to occupy the same position, even though the virtual reality device has a different pose than the one for which the virtual image was rendered.

[0044] Also shown at **218** is a non-adjusted position **219** of the world-locked virtual object, shown as a dashed box. The virtual reality device does not display the world-locked virtual object at position **219** in the adjusted virtual image. Rather, it is included in FIG. **2** in order to indicate the position at which the world-locked virtual object would appear, if the rendered virtual image was not adjusted by the late stage reprojection machine before display. Position **219** is shown to the left of world locked virtual image **211**, indicating that the late stage reprojection machine adjusted the virtual image by shifting it to the right, in response to the difference in position and/or orientation between the predicted future pose **207** and the updated tracked pose **213**.

[0045] In some embodiments, an environment of a virtual reality device may include virtual objects which are too large to fit within a field of view (FOV) of the virtual reality device, or are located entirely outside of the current FOV. For example, a user of a virtual reality device may play a game requiring the user to interact with a large virtual dragon. The dragon may be too large to fit entirely within the FOV of the virtual reality device, meaning a virtual image rendered for a given pose of the virtual reality device may not include the entire dragon, and instead only include the portions of the dragon which are currently visible in the FOV. Similarly, a user may interact with a virtual object, anchoring it to a particular location in the physical world, then move such that the virtual object is no longer visible within the virtual reality device's current FOV. As such, a virtual image rendered for a given pose of the virtual reality device may only include the virtual object when it is within the device's current FOV.

[0046] During remote rendering, virtual images may be rendered based on which virtual objects would be visible to a virtual reality device occupying a predicted future pose at a future time. For example, at a predicted future pose, an FOV of a virtual reality device may include only the head of a virtual dragon. As a result, a virtual image rendered for the predicted future pose may include only the dragon's head. However, this may pose a problem if, at an updated tracked pose, the FOV of the virtual reality device includes an area which should be occupied by a neck/body of the virtual dragon. The late stage reprojection machine could adjust the rendered virtual image in order to compensate for any differences between the predicted future pose and the updated tracked pose. However, this may not be possible if the rendered virtual image only includes the dragon's head, for example. Adjusting the virtual image may move the head of the dragon to a position consistent with the updated tracked pose, though leave blank space where the dragon's neck and body should be.

[0047] In order to address this problem, a virtual image may be rendered which is oversized relative to a set resolution of a virtual reality display, then cropped to match the set resolution before the virtual image is displayed. FIG. **3** illustrates this process. FIG. **3** shows a current FOV **300** of a virtual reality device. The dashed line indicates the edge of the virtual reality device's current FOV. In the illustrated embodiment, the FOV of the virtual reality device is identical to the FOV of a user of the virtual reality device. However, in some embodiments, the FOV of the virtual reality device may be smaller than the FOV of the user. In such embodiments, real-world objects may extend beyond the dashed line boundary of the virtual reality device FOV. In other embodiments, the FOV of the virtual reality device may be larger than the FOV of the user. Additional real-world and/or virtual objects may be present in the current environment of the virtual reality device which are not currently visible in the FOV.

[0048] At **302**, the virtual reality device requests a rendered virtual image. The rendered virtual image may be oversized relative to the set resolution of the virtual reality display. In some embodiments, virtual images may be automatically oversized at least to a degree, in order to ensure that adjustments can be made in the event that the predicted future pose does not match the updated tracked pose. Alternately, images may only be oversized when a threshold motion is detected by the one or more sensors. Further, an amount of oversizing may be proportional to a speed of the detected motion, the current framerate, and/or an average variation in network latency.

[0049] At **304**, FIG. **3** shows an oversized virtual image, as rendered by a remote computer. Objects in the real-world environment of the virtual reality device are drawn with fine dashed lines, so as to differentiate real-world objects from virtual objects. As shown, the virtual reality device is configured to display an image having a set resolution, and the virtual image is oversized relative to the set resolution.

[0050] The oversized virtual image includes virtual wizards **305**. Each virtual wizard is fully rendered by the remote computer, despite the fact that only part of each virtual wizard may be visible in the virtual reality device's FOV. Further, the virtual image is shown as being oversized in two dimensions—a horizontal dimension and a vertical dimension. However, in some embodiments, a degree of oversizing may be different in different dimension. For example, an amount of oversizing in a first dimension of the virtual image may be proportional to a speed of the detected motion in the first dimension, and an amount of oversizing in a second dimension of the virtual image may be independently proportional to a speed of the detected motion in the second dimension. For example, if a virtual reality device is moving rapidly in a horizontal direction, and slowly in a vertical dimension, a rendered virtual image may be greatly oversized in a horizontal dimension, and only slightly oversized in a vertical dimension. Dynamically setting the amount of oversizing according to detected motion may allow a virtual reality device to shift virtual images as necessary for proper display, without wasting resources by rendering unnecessarily large virtual images.

[0051] At **306**, the virtual reality device adjusts the rendered virtual image before it is displayed. In this case, adjusting the virtual image comprises cropping the virtual image to the set resolution of the virtual reality display. At this time, the late stage reprojection machine may perform any other adjustment operations before the virtual image is displayed. For example, the late stage reprojection machine may shift the virtual image in order to account for differences between the predicted future pose and the updated tracked pose.

[0052] Next, FIG. **3** shows an updated FOV **308** of the virtual reality device. The late stage reprojection machine has cropped the rendered virtual image **304** to the set resolution of the virtual reality display. As illustrated, only those portions of virtual wizards **305** which are visible within the FOV are displayed, even though each virtual wizard was rendered in its entirety as part of the oversized virtual image. After adjustment, the cropped virtual image will be displayed by the virtual reality display at or near the future time corresponding to the previously predicted pose.

[0053] Alternatively, a virtual image may be rendered which is undersized relative to a set resolution of a virtual reality display. Such images may be rendered and displayed in a similar manner to the oversized virtual images described above. For example, an undersized virtual image may be rendered which has a resolution which is less than the set resolution of the virtual reality display, and then stretched to fit the size of the virtual reality display. This may help maintain a constant framerate during periods of network congestion and high communications latency. An amount of

undersizing may be proportional to a speed of the detected motion, the current framerate, and/or an average variation in network latency.

[0054] In some embodiments, a rendered virtual image may be oversized relative to a size of a virtual reality display, though use the same resolution as the virtual reality display. In such a scenario, each pixel of the oversized virtual image will cover a larger area than a pixel from a rendered virtual image which is the same size as the virtual reality display. In some embodiments, such blurring may be desirable. For example, in the event that a virtual reality device is moving, a blurred virtual image may be rendered, simulating a motion blur effect. In some embodiments, additional operations may be performed in order to perform smoothing of a blurred virtual image before it is displayed.

[0055] FIG. **4**A illustrates shifting of virtual objects. In this example, the virtual image is a two-dimensional image including color information (an RGB image). Further, the virtual image includes virtual objects that are intended to be world-locked, such that they remain in approximately the same position relative to each other and to the real-world environment even as a virtual reality device moves. Specifically, FIG. **4**A includes virtual object original positions **400**, updated virtual object positions **402**, and virtual object shift indicators **404**. In the illustrated example, a virtual reality device has recently moved slightly downward and to the left, and as a result, the virtual image has shifted slightly up and to the right in the virtual reality device's FOV, thereby compensating for differences between the predicted future pose and the updated tracked pose. Virtual objects may be rendered to have a virtual depth (i.e., simulated distance from the virtual reality device). However, because the virtual image is a two-dimensional image, individual virtual objects are not shifted independently. Rather, the entire virtual image is shifted, meaning each virtual object is shifted the same display distance (i.e., number of pixels), regardless of virtual depth.

[0056] In some embodiments, the virtual image is a three-dimensional image, and includes a number of virtual objects, each rendered to appear at a different distance away from the virtual reality device. The rendered virtual image may include color and depth information (an RGBD image). For example, in addition to one or more color values used to illuminate the display (e.g., RGB values), each pixel may also include underlying depth information (e.g., D value) which is not displayed, but which can be used to position the individual virtual objects on the display if the predicted pose differs from an updated tracked pose.

[0057] FIG. **4**B illustrates shifting of virtual objects consistent with adjusting a three-dimensional RGBD image. Similar to FIG. **4**A, FIG. **4**B includes virtual object original positions **450**, virtual object updated positions **452**, and virtual object shift indicators **454**. However, in this example, adjusting the virtual image comprises independently shifting different aspects of the three-dimensional virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose. Adjusting the virtual image may further comprise independently shifting each particular aspect of the three-dimensional virtual image by an amount that is inversely proportional to a virtual depth of the particular aspect. This can be seen in FIG. **4**B, as each virtual object has shifted a different distance responsive to movement of the virtual reality device. Specifically, virtual objects which have a smaller virtual depth have shifted a greater distance than virtual objects which have a greater virtual depth, consistent with a parallax effect.

[0058] One or more virtual objects optionally may be locally rendered on the virtual reality device while other virtual objects are remotely rendered on the remote computer. In such embodiments, the virtual reality device may include a local virtual image renderer configured to locally render one or more virtual objects. Further, in such embodiments, the virtual reality display may be configured to display the one or more virtual objects rendered by the local virtual image renderer with the adjusted virtual image. This may be done for a variety of reasons. For example, relatively simple virtual objects may be rendered by a virtual reality device, while more complicated objects may be rendered remotely. Further, virtual objects with relatively low latency tolerance including, for example, cursors, may be rendered locally, where rendering can be done without network associated communications latency. For example, FIG. **4**B includes cursor **456**. In some embodiments, cursor **456**, and/or other virtual objects, may be rendered locally by a local image renderer, while other virtual objects are rendered remotely. The bifurcation of local and remote rendering can be performed with 2D and/or 3D virtual images.

[0059] In some embodiments, the remote computer may render a three-dimensional image and transmit it to the virtual reality device for adjustment and display. However, in the event of low network bandwidth/other network delays, or if the remote computer is unable to render virtual images at a framerate defined by the virtual reality device, the remote computer may render a two-dimensional RGB virtual image and send this virtual image to the virtual reality device along with an associated depth map. The late stage reprojection machine may then adjust the received virtual image according to differences between the predicted future pose and the updated tracked pose, and further according to information included in the received depth map. Adjusting the two-dimensional virtual image to include information from the depth map may be done through any suitable technique including, for example, rendering the image on a tessellated plane as a heightmap, or building a voxel representation of the virtual environment. Further, virtual images may be rendered and/or modified by the virtual reality device via ray tracing, constructed as a virtual mesh, or via virtually any other suitable techniques. Further, the virtual reality device may locally render a depth map, and use this locally rendered depth map to occlude portions of a received virtual image, thereby reducing the latency of any occlusion operations. The techniques described above may allow virtual image rendering and display to be performed efficiently, independent of network transmission speed and/or remote computer rendering speed.

[0060] In some embodiments, a rendered virtual image may include additional content provided by the remote computer. For example, a remote computer may be connected to one or more local/remote content sources which provide audio and/or visual content. Further, content provided by the remote computer may include information pertaining to one or more hardware components/software operations of the remote computer. The additional content may be included in the virtual image in any suitable manner. For example, the virtual image may include visual content from a remote/local video service. This visual content may be included in the virtual image as a floating virtual monitor,

or presented to the user in any other suitable manner. Additionally, when an application developer is using remote rendering to test changes in a software application, as described above, additional content provided to the virtual reality device may include, for example, debug information, and/or any other suitable information related to the virtual reality device and/or the remote computer.

[0061] FIG. **5** shows an example virtual object in the form of a content window **502**. Content window **502** displays additional content provided by a remote computer and pertaining to a software application the remote computer is currently executing. Specifically, content window **502** indicates that a remote computer is currently executing a program called "Wizard Adventure," and sending the rendered virtual imagery to the virtual reality device. Content window **502** further includes information pertaining to the current time frame of the program, as well as the current frame rate. FOV **500** may include any number of content windows similar to content window **502**, configured to display virtually any content provided by a remote computer. Further, a user may be able to interact with the content provided by the remote computer in any suitable way, including via one or more input devices included in the virtual reality device and/or included in the local environment. For example, a user may interact with content window **502** in order to change the information displayed, modify the code of the running application, and/or perform any other suitable operations.

[0062] FIG. **6** illustrates a method for rendering, adjusting, and displaying rendered virtual images. In some embodiments, a virtual reality device may be configured to switch between local and remote rendering of virtual images according to a variety of conditions. Further, during local rendering, a virtual reality device may still utilize pose prediction and late stage reprojection, as illustrated in FIG. **6**.

[0063] The method starts at **600**, where one or more sensors track a pose of the virtual reality device. Next, at **602**, a pose prediction machine predicts a future pose of the virtual reality device at a future time.

[0064] At **604**, the method determines if remote rendering is currently active. If remote rendering is not active (NO at **604**), then the method proceeds to **606**, where a local image renderer of the virtual reality device renders a virtual image for the predicted future pose. However, if remote rendering is active (YES at **604**), then the method proceeds to **614**.

[0065] From **606**, the method proceeds to **608**, where a late stage reprojection machine of the virtual reality device determines whether there are any differences between the future pose predicted at **602**, and an updated tracked pose determined by the one or more sensors. If differences exist (YES at **608**), then the method proceeds to **610**. If no differences exist (NO at **608**), then the method proceeds to **612**.

[0066] At **610**, the late stage reprojection machine adjusts the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device. Once the virtual image has been adjusted, the method proceeds to **612**, where a virtual reality display of the virtual reality device displays the adjusted virtual image at or near the future time.

[0067] At **614**, a communications machine of the virtual reality device sends to a remote computer a request for a virtual image having a perspective corresponding to the future pose of the virtual reality device. At **616**, a remote communications machine of the remote computer receives the request from the virtual reality device.

[0068] Next, at **618**, a virtual image renderer of the remote computer renders the virtual image for the virtual reality device as specified by the request received by the remote communications machine. Once the image has been rendered, the method proceeds to **620**, at which point the remote communications machine sends to the virtual reality device the rendered virtual image.

[0069] At **622**, the communications machine of the virtual reality device receives from the remote computer the virtual image, and the method proceeds to **608**. The method of FIG. **6** may be continually performed, and one instance of the method may begin before a previous instance of the method ends.

[0070] FIG. **7** shows aspects of an example head-mounted display (HMD) **700** to be worn and used by a wearer. The illustrated display system includes a frame **702**. The frame supports stereoscopic, see-through display componentry, which is positioned close to the wearer's eyes. HMD **700** may be used in augmented-reality applications, where real-world imagery is admixed with virtual display imagery. It will be appreciated that in other examples, the HMD device **700** may take other suitable forms in which transparent, semi-transparent, and/or non-transparent displays are supported in front of a viewer's eye or eyes. It will also be appreciated that the virtual reality device **100** shown in FIG. **1** may take the form of the HMD device **700**, as described in more detail below, or any other suitable HMD device.

[0071] HMD **700** includes separate right and left display panels, **704**R and **704**L, which may be wholly or partly transparent from the perspective of the wearer, to give the wearer a clear view of his or her surroundings. For example, the appearance of the physical environment may be augmented by graphical content (e.g., one or more pixels each having a respective color and brightness) that is presented via the display panels to create an augmented reality environment.

[0072] Each display panel includes a backlight and a liquid-crystal display (LCD) type microdisplay. The backlight may include an ensemble of light-emitting diodes (LEDs)—e.g., white LEDs or a distribution of red, green, and blue LEDs. The backlight may be configured to direct its emission through the LCD microdisplay, which forms a display image based on control signals from compute system **706**. The LCD microdisplay may include numerous, individually addressable pixels arranged on a rectangular grid or other geometry. In some embodiments, pixels transmitting red light may be juxtaposed to pixels transmitting green and blue light, so that the LCD microdisplay forms a color image. In other embodiments, a reflective liquid-crystal-on-silicon (LCOS) microdisplay or a digital micromirror array may be used in lieu of the LCD microdisplay. Alternatively, an active LED, holographic, or scanned-beam microdisplay may be used to form right and left display images. Although the drawings show separate right and left display panels, a single display panel extending over both eyes may be used instead.

[0073] Compute system **706** is operatively coupled to the display panels and to other display-system componentry. The compute system may include a logic machine and a storage machine, as discussed in more detail below with respect to FIG. **8**, that are in communication with the various

sensors and systems of the HMD device and display. In one example, the storage machine may include instructions that are executable by the logic machine to receive sensor data from the sensors and predict a future pose of the HMD device, request a rendered virtual image via a communications machine, adjust a rendered virtual image, and display an adjusted virtual image via right and/or left display panels. HMD **700** may include an accelerometer **708**K, gyroscope **708**L, and magnetometer **708**M, stereo loudspeakers **710**R and **710**L, a color camera **712**, and a depth camera **714**.

[0074] HMD device **700** may also include a head tracking system that utilizes one or more motion sensors, such as sensors **708** on HMD device **700**, to capture pose data and thereby enable position tracking, direction and orientation sensing, and/or motion detection of the user's head and/or HMD device **700**.

[0075] The head tracking system may also support other suitable positioning techniques, such as GPS or other global navigation systems. Further, while specific examples of position sensor systems have been described, it will be appreciated that any other suitable position sensor systems may be used. For example, pose and/or movement data may be determined based on sensor information from any combination of sensors mounted on the wearer and/or external to the wearer including, but not limited to, any number of gyroscopes, accelerometers, inertial measurement units (IMUs), GPS devices, barometers, magnetometers, cameras (e.g., visible light cameras, infrared light cameras, time-of-flight depth cameras, structured light depth cameras, etc.), communication devices (e.g., WIFI antennas/interfaces), etc.

[0076] In some examples the HMD device **700** may also include an optical sensor system that utilizes one or more outward facing sensors, such as color camera **712** or depth camera **714** on HMD device **700**, to capture image data. The outward facing sensor(s) may detect movements within its field of view, such as gesture-based inputs or other movements performed by a user or by a person or physical object within the field of view. The outward facing sensor(s) may also capture 2D image information and depth information from the physical environment and physical objects within the environment.

[0077] The optical sensor system may include a depth tracking system that generates depth tracking data via one or more depth cameras. In one example, each depth camera may include left and right cameras of a stereoscopic vision system. Time-resolved images from one or more of these depth cameras may be registered to each other and/or to images from another optical sensor such as a visible spectrum camera, and may be combined to yield depth-resolved video.

[0078] In other examples a structured light depth camera may be configured to project a structured infrared illumination, and to image the illumination reflected from a scene onto which the illumination is projected. A depth map of the scene may be constructed based on spacings between adjacent features in the various regions of an imaged scene. In still other examples, a depth camera may take the form of a time-of-flight depth camera configured to project a pulsed infrared illumination onto a scene and detect the illumination reflected from the scene. For example, modulated illumination may be provided by an infrared light source, and a phase difference between projected and reflected illumination may be used to calculate the time between projecting and detecting the light. It will be appreciated that any other suitable depth camera may be used within the scope of the present disclosure.

[0079] The outward facing sensor(s) may capture images of the physical environment in which a user is situated. With respect to the HMD device **700**, in one example a mixed reality display program may include a 3D modeling system that uses such captured images to generate a virtual environment that models the physical environment surrounding the user.

[0080] In one example, the HMD **700** may include a gaze tracking system configured to detect a direction of gaze of each eye of a user. The gaze detection subsystem may be configured to determine gaze directions of each of a user's eyes in any suitable manner. For example, the gaze detection subsystem may comprise one or more light sources, such as infrared light sources, configured to cause a glint of light to reflect from the cornea of each eye of a user. One or more image sensors may then be configured to capture an image of the user's eyes.

[0081] Images of the glints and of the pupils as determined from image data gathered from the image sensors may be used to determine an optical axis of each eye. Using this information, the gaze tracking system may then determine a direction the user is gazing. The gaze tracking system may additionally or alternatively determine at what physical or virtual object the user is gazing. Such gaze tracking data may then be provided to the HMD device **700**.

[0082] It will also be understood that the gaze tracking system may have any suitable number and arrangement of light sources and image sensors. For example, the gaze tracking system of the HMD device **700** may utilize at least one inward facing sensor.

[0083] The HMD device **700** may also include a microphone system that includes one or more microphones that capture audio data. In other examples, audio may be presented to the user via one or more speakers, such as speakers **710** on the HMD device **700**.

[0084] In some embodiments, the methods and processes described herein may be tied to one or more computing systems 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.

[0085] FIG. **8** schematically shows a non-limiting embodiment of a local computer **800** and a remote computer **816** that can enact one or more of the methods and processes described above. Computers **800** and **816** are shown in simplified form. Computers **800** and **816** may take the form of one or more AR/VR HMD computers, personal computers, server computers, tablet computers, home-entertainment computers, network computing devices, gaming devices, mobile computing devices, mobile communication devices (e.g., smart phone), wearable computing devices, and/or other computing devices. In particular, local computer **800** and remote computer **816** may be implemented as a virtual reality device and/or a remote computer as described above.

[0086] Local computer **800** includes a logic machine **802** and a storage machine **804**. Local computer **800** may optionally include a display subsystem **810**, input subsystem **812**, communications machine **814**, and/or other components not shown in FIG. **8**. Similarly, remote computer **816** includes a logic machine **818** and a storage machine **820**. Remote

computer **816** may optionally include a display subsystem **824**, input subsystem **826**, remote communications machine **828**, and/or other components not shown in FIG. **8**.

[0087] Logic machines **802** and **818** include one or more physical devices configured to execute instructions. For example, the logic machines 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 components, achieve a technical effect, or otherwise arrive at a desired result.

[0088] The logic machines may include one or more processors configured to execute software instructions. Additionally or alternatively, the logic machines may include one or more hardware or firmware logic machines configured to execute hardware or firmware instructions. Processors of the logic machines may be single-core or multi-core, and the instructions executed thereon may be configured for sequential, parallel, and/or distributed processing. Individual components of the logic machines optionally may be distributed among two or more separate devices, which may be remotely located and/or configured for coordinated processing. Aspects of the logic machines may be virtualized and executed by remotely accessible, networked computing devices configured in a cloud-computing configuration.

[0089] Storage machines **804** and **820** include one or more physical devices configured to hold instructions executable by the logic machines to implement the methods and processes described herein. When such methods and processes are implemented, the state of storage machines **804** and **820** may be transformed—e.g., to hold different data.

[0090] Storage machines **804** and **820** may include removable and/or built-in devices. Storage machines **804** and **820** 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 machines **804** and **820** may include volatile, nonvolatile, dynamic, static, read/write, read-only, random-access, sequential-access, location-addressable, file-addressable, and/or content-addressable devices.

[0091] It will be appreciated that storage machines **804** and **820** include 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.

[0092] Aspects of logic machines **802** and **818**, and storage machines **804** and **820**, 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.

[0093] The pose prediction machine **806** and the late stage reprojection machine **808** may be implemented via logic machine **802** and storage machine **804** and/or any other suitable devices. Further, the virtual image renderer **822** may be implemented via logic machine **818** and storage machine **820** and/or any other suitable devices. In some embodiments, the local computer **800** may include a local image renderer, as described above, which may additionally be implemented via logic machine **802** and storage machine **804**. Further, the remote computer may include a pose prediction machine, as described above, which may be implemented via logic machine **818** and storage machine **820**.

[0094] The terms "module," "program," and "engine" may be used to describe an aspect of computing system **800** implemented to perform a particular function. In some cases, a module, program, or engine may be instantiated via logic machines **802** and/or **818** executing instructions held by storage machines **804** and/or **820**. It will be understood that different modules, programs, and/or engines may be instantiated from the same application, service, code block, object, library, routine, API, function, etc. Likewise, the same module, program, and/or engine may be instantiated by different applications, services, code blocks, objects, routines, APIs, functions, etc. The terms "module," "program," and "engine" may encompass individual or groups of executable files, data files, libraries, drivers, scripts, database records, etc.

[0095] It will be appreciated that a "service", as used herein, is an application program executable across multiple user sessions. A service may be available to one or more system components, programs, and/or other services. In some implementations, a service may run on one or more server-computing devices.

[0096] When included, display subsystems **810** and/or **824** may be used to present a visual representation of data held by storage machines **804** and/or **820**. This visual representation may take the form of a graphical user interface (GUI). As the herein described methods and processes change the data held by the storage machines, and thus transform the state of the storage machines, the state of display subsystems **810** and **824** may likewise be transformed to visually represent changes in the underlying data. Display subsystems **810** and **824** may include one or more display devices utilizing virtually any type of technology. Such display devices may be combined with logic machines **802** and **818** and/or storage machines **804** and **820** in a shared enclosure, or such display devices may be peripheral display devices. In particular, display subsystem **810** may in some embodiments be implemented as a virtual reality display.

[0097] When included, input subsystems **812** and **826** may comprise or interface with one or more user-input devices such as a keyboard, mouse, touch screen, or game controller. In some embodiments, the input subsystems may comprise or interface with selected natural user input (NUI) componentry. Such componentry may be integrated or peripheral, and the transduction and/or processing of input actions may be handled on- or off-board. Example NUI componentry may include a microphone for speech and/or voice recognition; an infrared, color, stereoscopic, and/or depth camera for machine vision and/or gesture recognition; a head tracker, eye tracker, accelerometer, and/or gyroscope for motion detection and/or intent recognition; as well as electric-field sensing componentry for assessing brain activity. In particular, the one or more sensors described above may in some embodiments be implemented as input subsystems **812** and/or **826**.

[0098] When included, communications machine **814** and remote communications machine **828** may be configured to

communicatively couple local computer **800** to remote computer **816** via network **830**, and may be further configured to allow the local computer and/or the remote computer to communicate with one or more other computer systems. Communications machine **814** and remote communications machine **828** may include wired and/or wireless communication devices compatible with one or more different communication protocols. As non-limiting examples, the communications machines may be configured for communication via a wireless telephone network, or a wired or wireless local- or wide-area network. In some embodiments, the communications machines may allow local computer **800** and remote computer **816** to send and/or receive messages to and/or from other devices via a network such as the Internet.

[0099] In an example, a virtual reality device comprises: one or more sensors configured to track a pose of the virtual reality device; a pose prediction machine configured to predict a future pose of the virtual reality device at a future time; a communications machine configured to send to a remote computer a request for a virtual image having a perspective corresponding to the future pose of the virtual reality device and to receive from the remote computer the virtual image; a late stage reprojection machine configured to adjust the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device; and a virtual reality display configured to display the adjusted virtual image. In this example or any other example, the communications machine is further configured to determine a communications latency by measuring an elapsed time between the request being sent and the virtual image being received. In this example or any other example, the pose prediction machine is configured to dynamically adjust a buffer period between a current time and the future time for which the future pose is predicted, the buffer period dynamically increased with increasing communications latency and dynamically decreased with decreasing communications latency. In this example or any other example, the one or more sensors are configured to detect a motion of the virtual reality device in one or more dimensions. In this example or any other example, the pose prediction machine is configured to predict the future pose by extrapolating the detected motion forward from a current time to the future time. In this example or any other example, extrapolating the detected motion comprises extrapolating the detected motion forward with biomechanical constraints. In this example or any other example, extrapolating the detected motion comprises extrapolating the detected motion forward with environmental constraints. In this example or any other example, the virtual reality device is configured to display an image having a set resolution, and the virtual image is oversized relative to the set resolution. In this example or any other example, an amount of oversizing is proportional to a speed of the detected motion. In this example or any other example, an amount of oversizing in a first dimension of the virtual image is proportional to a speed of the detected motion in the first dimension, and an amount of oversizing in a second dimension of the virtual image is independently proportional to a speed of the detected motion in the second dimension. In this example or any other example, adjusting the virtual image comprises cropping the virtual image to the set resolution. In this example or any other example, adjusting the virtual image comprises shifting the virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose. In this example or any other example, the virtual image is a three-dimensional image, adjusting the virtual image comprises independently shifting different aspects of the three-dimensional virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose, and further comprises independently shifting each particular aspect of the three-dimensional virtual image by an amount that is inversely proportional to a virtual depth of the particular aspect. In this example or any other example, the virtual reality device further comprises a local virtual image renderer configured to locally render one or more virtual objects, and the virtual reality display is configured to display the one or more virtual objects rendered by the local virtual image renderer with the adjusted virtual image. In this example or any other example, the virtual reality device and the remote computer communicate over a computer network, and each of the virtual reality device and the remote computer occupy a different network node. In this example or any other example, the virtual image received from the remote computer includes additional content provided by the remote computer.

[0100] In an example, a computing device comprises: a remote communications machine configured to receive from a virtual reality device a request for a virtual image having a perspective corresponding to a predicted future pose of the virtual reality device, and send to the virtual reality device a rendered virtual image; and a virtual image renderer configured to render the virtual image for the virtual reality device as specified by the request received by the remote communications machine.

[0101] In an example, a virtual reality device comprises: one or more sensors configured to track a pose of the virtual reality device; a communications machine configured to send to a remote computer a request for a virtual image having a perspective corresponding to a future pose of the virtual reality device and to receive from the remote computer the virtual image; a late stage reprojection machine configured to adjust the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device; and a virtual reality display configured to display the adjusted virtual image. In this example or any other example, a future time of the future pose is dynamically predicted based on communication latency. In this example or any other example, the future pose is predicted by extrapolating detected motion forward from a current time to the future time.

[0102] 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.

[0103] The subject matter of the present disclosure includes all novel and nonobvious combinations and subcombinations of the various processes, systems and configu-

rations, and other features, functions, acts, and/or properties disclosed herein, as well as any and all equivalents thereof.

1. A virtual reality device, comprising:

one or more sensors configured to track a pose of the virtual reality device;

a pose prediction machine configured to predict a future pose of the virtual reality device at a future time;

a communications machine configured to send to a remote computer a request for a virtual image having a perspective corresponding to the future pose of the virtual reality device and to receive from the remote computer the virtual image;

a late stage reprojection machine configured to adjust the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device; and

a virtual reality display configured to display the adjusted virtual image.

2. The virtual reality device of claim **1**, wherein the communications machine is further configured to determine a communications latency by measuring an elapsed time between the request being sent and the virtual image being received.

3. The virtual reality device of claim **2**, wherein the pose prediction machine is configured to dynamically adjust a buffer period between a current time and the future time for which the future pose is predicted, the buffer period dynamically increased with increasing communications latency and dynamically decreased with decreasing communications latency.

4. The virtual reality device of claim **1**, wherein the one or more sensors are configured to detect a motion of the virtual reality device in one or more dimensions.

5. The virtual reality device of claim **4**, wherein the pose prediction machine is configured to predict the future pose by extrapolating the detected motion forward from a current time to the future time.

6. The virtual reality device of claim **5**, wherein extrapolating the detected motion comprises extrapolating the detected motion forward with biomechanical constraints.

7. The virtual reality device of claim **5**, wherein extrapolating the detected motion comprises extrapolating the detected motion forward with environmental constraints.

8. The virtual reality device of claim **4**, wherein the virtual reality device is configured to display an image having a set resolution, and wherein the virtual image is oversized relative to the set resolution.

9. The virtual reality device of claim **8**, wherein an amount of oversizing is proportional to a speed of the detected motion.

10. The virtual reality device of claim **9**, wherein an amount of oversizing in a first dimension of the virtual image is proportional to a speed of the detected motion in the first dimension, and wherein an amount of oversizing in a second dimension of the virtual image is independently proportional to a speed of the detected motion in the second dimension.

11. The virtual reality device of claim **10**, wherein adjusting the virtual image comprises cropping the virtual image to the set resolution.

12. The virtual reality device of claim **1**, wherein adjusting the virtual image comprises shifting the virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose.

13. The virtual reality device of claim **12**, wherein the virtual image is a three-dimensional image, adjusting the virtual image comprises independently shifting different aspects of the three-dimensional virtual image by an amount that is proportional to a difference between the predicted future pose and the updated tracked pose, and further comprises independently shifting each particular aspect of the three-dimensional virtual image by an amount that is inversely proportional to a virtual depth of the particular aspect.

14. The virtual reality device of claim **1**, further comprising a local virtual image renderer configured to locally render one or more virtual objects, wherein the virtual reality display is configured to display the one or more virtual objects rendered by the local virtual image renderer with the adjusted virtual image.

15. The virtual reality device of claim **1**, wherein the virtual reality device and the remote computer communicate over a computer network, and each of the virtual reality device and the remote computer occupy a different network node.

16. The virtual reality device of claim **1**, wherein the virtual image received from the remote computer includes additional content provided by the remote computer.

17. A computing device, comprising:

a remote communications machine configured to receive from a virtual reality device a request for a virtual image having a perspective corresponding to a predicted future pose of the virtual reality device, and send to the virtual reality device a rendered virtual image; and

a virtual image renderer configured to render the virtual image for the virtual reality device as specified by the request received by the remote communications machine.

18. A virtual reality device, comprising:

one or more sensors configured to track a pose of the virtual reality device;

a communications machine configured to send to a remote computer a request for a virtual image having a perspective corresponding to a future pose of the virtual reality device and to receive from the remote computer the virtual image;

a late stage reprojection machine configured to adjust the virtual image to an adjusted virtual image having an updated perspective corresponding to an updated tracked pose of the virtual reality device; and

a virtual reality display configured to display the adjusted virtual image.

19. The virtual reality device of claim **18**, wherein a future time of the future pose is dynamically predicted based on communication latency.

20. The virtual reality device of claim **19**, wherein the future pose is predicted by extrapolating detected motion forward from a current time to the future time.

* * * * *