



(51) **International Patent Classification:**
G06T 7/292 (2017.01) *G06T 7/33* (2017.01)
G06T 7/246 (2017.01)

(21) **International Application Number:**
PCT/US2021/029515

(22) **International Filing Date:**
28 April 2021 (28.04.2021)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
16/936,377 22 July 2020 (22.07.2020) US

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

(72) **Inventors: BLEYER, Michael;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **EDMONDS, Christopher Douglas;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SAMPLES, Michael Edward;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SINHA, Sudipta Narayan;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **KARR, Matthew Beaudoin;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **PRICE, Raymond Kirk;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) **Agent: SWAIN, Cassandra T. et al.;** Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(54) **Title:** SYSTEMS AND METHODS FOR CONTINUOUS IMAGE ALIGNMENT OF SEPARATE CAMERAS

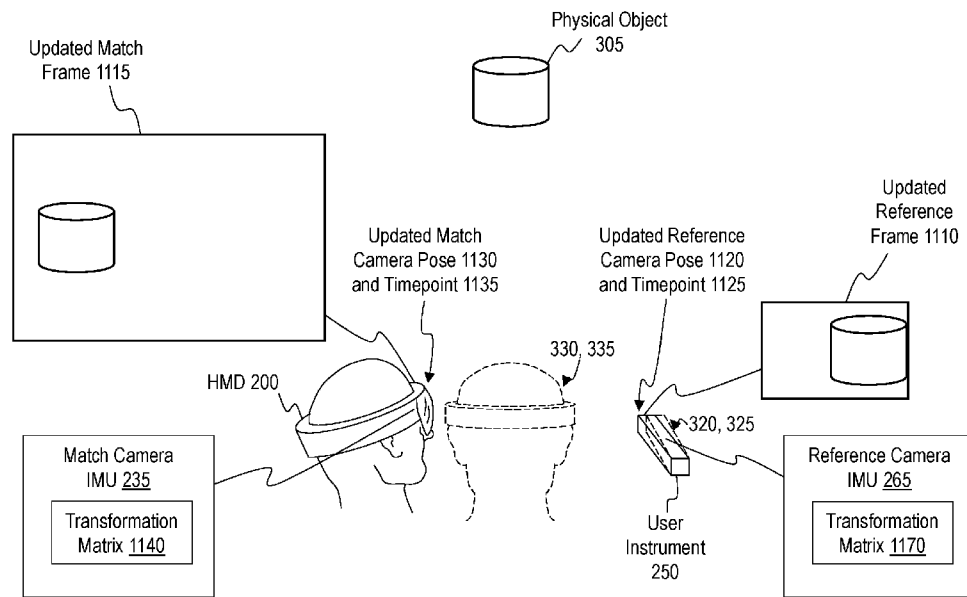
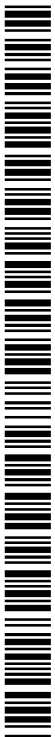


FIG. 11

(57) **Abstract:** A system for continuous image alignment of separate cameras identifies a reference camera transformation matrix between a base reference camera pose and an updated reference camera pose. The system also identifies a match camera transformation matrix between a base match camera pose and an updated match camera pose and an alignment matrix based on visual correspondences between one or more reference frames captured by the reference camera and one or more match frames captured by the match camera. The system also generates a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by the reference camera to a corresponding set of pixels of a match frame captured by the match camera based on the reference camera transformation matrix, the match camera transformation matrix, and the alignment matrix.



(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, IT, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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))*

SYSTEMS AND METHODS FOR CONTINUOUS IMAGE ALIGNMENT OF SEPARATE CAMERAS

BACKGROUND

5 [0001] Mixed-reality (MR) systems, including virtual-reality and augmented-reality systems, have received significant attention because of their ability to create truly unique experiences for their users. For reference, conventional virtual-reality (VR) systems create a completely immersive experience by restricting their users' views to only a virtual environment. This is often achieved, in VR systems, through the use of a head-mounted
10 device (HMD) that completely blocks any view of the real world. As a result, a user is entirely immersed within the virtual environment. In contrast, conventional augmented-reality (AR) systems create an augmented-reality experience by visually presenting virtual objects that are placed in or that interact with the real world.

[0002] As used herein, VR and AR systems are described and referenced
15 interchangeably. Unless stated otherwise, the descriptions herein apply equally to all types of mixed-reality systems, which (as detailed above) includes AR systems, VR reality systems, and/or any other similar system capable of displaying virtual objects.

[0003] Some MR systems include one or more cameras and utilize images and/or depth information obtained using the camera(s) to provide pass-through views of a user's
20 environment to the user. A pass-through view can aid users in avoiding disorientation and/or safety hazards when transitioning into and/or navigating within a mixed-reality environment. Pass-through views may also enhance user views in low visibility environments. For example, mixed-reality systems configured with long wavelength thermal imaging cameras may facilitate visibility in smoke, haze, fog, and/or dust. Likewise,
25 mixed-reality systems configured with low light imaging cameras facilitate visibility in dark environments where the ambient light level is below the level required for human vision.

[0004] An MR system may provide pass-through views in various ways. For example, an MR system may present raw images captured by the camera(s) of the MR system to a user. In other instances, an MR system may modify and/or reproject captured image data to
30 correspond to the perspective of a user's eye to generate pass-through views. An MR system may modify and/or reproject captured image data to generate a pass-through view using depth information for the captured environment obtained by the MR system (e.g., using a depth system of the MR system, such as a time of flight camera, a rangefinder, stereoscopic depth cameras, etc.). In some instances, an MR system utilizes one or more predefined depth

values to generate pass-through views (e.g., by performing planar reprojection).

5 [0005] In some instances, pass-through views generated by modifying and/or reprojecting captured image data may at least partially correct for differences in perspective brought about by the physical separation between a user's eyes and the camera(s) of the MR system (known as the "parallax problem," "parallax error," or, simply "parallax"). Such pass-through views/images may be referred to as "parallax-corrected pass-through" views/images. By way of illustration, parallax-corrected pass-through images may appear to a user as though they were captured by cameras that are co-located with the user's eyes.

10 [0006] MR systems are often used in combination with other devices that are physically independent from and/or untethered to the MR system (e.g., controllers, instruments, etc.). For example, a user may operate an MR system while also operating a handheld/wearable device that includes a device camera. The device camera may be configured to be directed at and/or capture portions of the environment that are within the field of view of the MR system, such that both the device camera and cameras of the MR system concurrently
15 capture portions of the environment.

[0007] User experiences may be enhanced by providing composite pass-through images on an MR system that implement camera imagery captured by a separate device camera aligned with and overlaid on camera imagery captured by the camera(s) of the MR system. However, because MR system cameras and separate device cameras move independently of
20 one another, accurately aligning the camera imagery of the separate cameras to generate such composite pass-through images is associated with many challenges, particularly in real-time and/or near-real-time implementations.

[0008] For at least the foregoing reasons, there is an ongoing need and desire for improved techniques and systems for facilitating continuous image alignment of separate
25 cameras.

[0009] The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practiced.

30 BRIEF SUMMARY

[0010] Disclosed embodiments include systems and methods for facilitating continuous image alignment of separate cameras.

[0011] Some embodiments include methods implemented by a computing system, such as a head-mounted display (HMD), in which the computing system performs various acts,

including an act of identifying a reference camera transformation matrix between a base reference camera pose and an updated reference camera pose. The base reference camera pose is associated with a base reference camera timepoint that occurs prior to an updated reference camera timepoint. The computing system also performs an act of identifying a match camera transformation matrix between a base match camera pose and an updated match camera pose. The base match camera pose is associated with a base match camera timepoint that occurs prior to an updated match camera timepoint. The computing system also performs an act of identifying an alignment matrix based on visual correspondences between one or more reference frames captured by the reference camera and one or more match frames captured by the match camera. The computing system also generates a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by the reference camera to a corresponding set of pixels of a match frame captured by the match camera. The motion model is generated based on at least the reference camera transformation matrix, the match camera transformation matrix, and the alignment matrix.

[0012] 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 as an aid in determining the scope of the claimed subject matter.

[0013] Additional features and advantages will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the teachings herein. Features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. Features of the present invention will become more fully apparent from the following description and appended claims or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] In order to describe the manner in which the above-recited and other advantages and features can be obtained, a more particular description of the subject matter briefly described above will be rendered by reference to specific embodiments which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments and are not therefore to be considered to be limiting in scope, embodiments will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0015] Figure 1 illustrates an example mixed-reality system that may include or be used

to implement disclosed embodiments;

[0016] Figure 2 illustrates an example head-mounted display (HMD) and a user instrument that include various cameras that may facilitate the disclosed embodiments, including a reference camera and a match camera;

5 [0017] Figure 3 illustrates an example of capturing an environment with a reference camera and a match camera;

[0018] Figure 4 illustrates an example of feature matching between a reference frame and a match frame;

[0019] Figure 5 illustrates an example of unprojecting the feature matches identified according to Figure 4;

10 [0020] Figure 6 illustrates an example of identifying a base matrix using the unprojected feature matches of Figure 5;

[0021] Figure 7 illustrates an example of unprojecting a set of pixels of a reference frame to generated 3D points;

15 [0022] Figure 8 illustrates an example of generating modified 3D points by applying the base matrix of Figure 6 to the 3D points of Figure 7;

[0023] Figure 9 illustrates an example of projecting the modified 3D points of Figure 8;

[0024] Figure 10 illustrates an example of a composite image that includes reference frame pixels mapped to corresponding match frame pixels in an overlap region;

20 [0025] Figure 11 illustrates an example of capturing the environment from Figure 3 with the reference camera and the match camera at subsequent timepoints;

[0026] Figure 12 illustrates an example of unprojecting a set of pixels of an updated reference frame to generate 3D points;

[0027] Figure 13 illustrates an example of generating modified 3D points by applying a reference camera transformation matrix to the 3D points of Figure 12;

25 [0028] Figure 14 illustrates an example of generating modified 3D points by applying the base matrix to the modified 3D points of Figure 13;

[0029] Figure 15 illustrates an example of generating modified 3D points by applying a match camera transformation matrix to the 3D points of Figure 14;

30 [0030] Figure 16 illustrates an example of projecting the modified 3D points of Figure 15;

[0031] Figure 17 illustrates an example of a composite image that includes updated reference frame pixels mapped to updated corresponding match frame pixels in an overlap region;

[0032] Figure 18 illustrates an example of feature matching between the updated reference frame and the updated match frame;

[0033] Figure 19 illustrates an example of unprojecting the feature matches identified according to Figure 18;

5 [0034] Figure 20 illustrates an example of identifying an updated matrix using the unprojected feature matches of Figure 19;

[0035] Figures 21 and 22 illustrate an example of generating an aligned updated matrix by modifying the updated matrix from Figure 20 using inertial tracking data associated with the reference camera and the match camera;

10 [0036] Figure 23A illustrates an example of generating an alignment matrix using a base matrix and the aligned updated matrix from Figures 21 and 22;

[0037] Figure 23B illustrates an example of generating a subsequent alignment matrix using the alignment matrix from Figure 23A and a subsequent aligned updated matrix;

15 [0038] Figure 24 illustrates an example of generating a composite image using a reference frame, a match frame, and a motion model;

[0039] Figures 25 illustrates an example flow diagram depicting acts associated with generating a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by a reference camera to a corresponding set of pixels of a match frame captured by a match camera;

20 [0040] Figure 26 illustrates an example flow diagram depicting acts associated with facilitating continuous image alignment of two cameras;

[0041] Figure 27 illustrates an example of a reference camera and a match camera capturing an environment at different timepoints;

25 [0042] Figure 28 illustrates example alignment matrices that may be associated with frame pairs captured by the reference camera and the match camera;

[0043] Figure 29 illustrates an example of difference values associated with the reference camera and the match camera at different timepoints;

30 [0044] Figure 30 illustrates an example of generating an updated alignment matrix using an aligned current matrix, a previous alignment matrix, and one or more difference values as inputs;

[0045] Figure 31 illustrates examples of difference values that may be associated with the reference camera and the match camera at different timepoints;

[0046] Figure 32 illustrates an example plot of a smoothness function according to an implementation of the present disclosure;

[0047] Figure 33 illustrates an example flow diagram depicting acts associated with updating continuous image alignment of a reference camera and a match camera; and

[0048] Figure 34 illustrates an example computer system that may include and/or be used to perform disclosed embodiments.

5

DETAILED DESCRIPTION

[0049] Disclosed embodiments include systems and methods for facilitating continuous image alignment of separate cameras.

[0050] Some embodiments include methods implemented by a computing system, such as a head-mounted display (HMD), in which the computing system performs various acts, including an act of identifying a reference camera transformation matrix between a base reference camera pose and an updated reference camera pose. The base reference camera pose is associated with a base reference camera timepoint that occurs prior to an updated reference camera timepoint.

[0051] The computing system also performs an act of identifying a match camera transformation matrix between a base match camera pose and an updated match camera pose. The base match camera pose is associated with a base match camera timepoint that occurs prior to an updated match camera timepoint.

[0052] The computing system also performs an act of identifying an alignment matrix based on visual correspondences between one or more reference frames captured by the reference camera and one or more match frames captured by the match camera.

[0053] The computing system also generates a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by the reference camera to a corresponding set of pixels of a match frame captured by the match camera. The motion model is generated based on at least the reference camera transformation matrix, the match camera transformation matrix, and the alignment matrix.

Examples of Technical Benefits, Improvements, and Practical Applications

[0054] Those skilled in the art will recognize, in view of the present disclosure, that at least some of the disclosed embodiments may address various shortcomings associated with conventional approaches for facilitating continuous image alignment of separate cameras. The following section outlines some example improvements and/or practical applications provided by the disclosed embodiments. It will be appreciated, however, that the following are examples only and that the embodiments described herein are in no way limited to the example improvements discussed herein.

[0055] In some implementations, providing a motion model based on reference camera

and match camera transformation matrices (for poses) and an alignment matrix enables a system to map pixels of a reference frame to a match frame (and/or vice versa) as the positions of the reference camera and the match camera change relative to one another over time. For example, the reference camera and match camera transformation matrices may be updated/determined as the poses of the reference camera and the match camera change, which enables the motion model to account for updates in the positions of the cameras.

[0056] In some implementations, the alignment matrix is a 3D rotational matrix, which may correspond to reference camera and match camera transformation matrices based on inertial tracking data obtained by inertial measurement units (IMU) associated with the reference camera and the match camera, respectively. Accordingly, at least some implementations of the present disclosure facilitate simple concatenation of the alignment matrix with the camera transformation matrices based on IMU data to generate a motion model.

[0057] Furthermore, in some instances, the alignment matrix may be updated/determined based on visual correspondences between images captured by the reference camera and the match camera, which may enable the motion model to ameliorate the effects of IMU drift and/or parallax.

[0058] In addition, an alignment matrix may be generated by fusing (e.g., blending) a base matrix (based on visual correspondences for a base frame pair) with an updated matrix (based on visual correspondences for an updated frame pair obtained subsequent to the base frame pair), which may address noise/imprecision that may be associated with using visual correspondences to determine an alignment matrix.

[0059] Still furthermore, a system may intelligently determine whether or how to blend a previous alignment matrix with an updated/current matrix to generate an updated alignment matrix based on difference values associated with the reference camera and/or the match camera relative to the timepoints at which a previous frame pair and an updated frame pair were obtained. Such implementations may enable systems to selectively ignore past alignment matrices upon indications that they have become unreliable (e.g., where sufficient motion has occurred or time has elapsed since the previous alignment matrix was computed).

[0060] A motion model of the present disclosure may enable a system to continuously map pixels of a reference frame captured by a reference camera to corresponding pixels of a match frame captured by a match camera. Thus, a motion model may enable a system to generate composite pass-through images that include imagery of the reference camera (e.g.,

a separate device camera) aligned with and overlaid on imagery of the match camera (e.g., one or more head-mounted display (HMD) camera). Providing such composite pass-through images may enable users to readily identify which portion of an environment a separate device (and/or device camera) is directed toward, which may enable users to direct input and/or actions toward portions of the environment in an accurate and/or rapid manner. Such composite pass-through images may be beneficial in various applications, such as, for example, gaming environments, medical/dental operations/training, first responder training/activities, and/or others.

[0061] Although the present disclosure focuses, in some respects, on facilitating continuous image alignment of a match camera of an HMD and a reference camera of a user instrument (e.g., separate user device) to provide composite pass-through views for display on the HMD, it will be appreciated, in view of the present disclosure, that the principles disclosed herein are applicable to any implementation that involves providing continuous image alignment between any number of separate cameras.

[0062] Having just described some of the various high-level features and benefits of the disclosed embodiments, attention will now be directed to Figures 1 through 33. These Figures illustrate various conceptual representations, architectures, methods, and supporting illustrations related to the disclosed embodiments. The disclosure will then turn to Figure 34, which presents an example computer system that may include and/or be used to facilitate the disclosed principles.

Example Mixed-Reality Systems and HMDs

[0063] Attention will now be directed to Figure 1, which illustrates an example of a head-mounted device (HMD) 100. HMD 100 can be any type of mixed-reality system 100A (MR system), including a VR system 100B or an AR system 100C. It should be noted that while a substantial portion of this disclosure is focused, in some respects, on the use of an HMD, the embodiments are not limited to being practiced using only an HMD. That is, any type of system can be used, even systems entirely removed or separate from an HMD. As such, the disclosed principles should be interpreted broadly to encompass any type of scanning scenario or device. Some embodiments may even refrain from actively using a scanning device themselves and may simply use the data generated by the scanning device. For instance, some embodiments may at least be partially practiced in a cloud computing environment.

[0064] Figure 1 illustrates HMD 100 as including sensor(s) 150, including scanning sensor(s) 105 and other sensors, such as accelerometer(s) 155, gyroscope(s) 160,

compass(es) 165. The ellipsis 170 conveys that the sensor(s) 150 depicted in Figure 1 are illustrative only and non-limiting. For instance, in some implementations, an HMD 100 includes other interoceptive and/or exteroceptive sensors not explicitly illustrated in Figure 1, such as eye tracking systems, radio-based navigation systems, microphones, and/or other sensing apparatuses. In some implementations, an HMD 100 includes fewer sensors than those depicted in Figure 1.

[0065] The accelerometer(s) 155, gyroscope(s) 160, and compass(es) 165 are configured to measure inertial tracking data. Specifically, the accelerometer(s) 155 is/are configured to measure acceleration, the gyroscope(s) 160 is/are configured to measure angular velocity data, and the compass(es) 165 is/are configured to measure heading data. In some instances, an HMD 100 utilizes the inertial tracking components thereof to obtain three degree of freedom (3DOF) pose data associated with the HMD (e.g., where visual tracking data, described below, is unavailable, unreliable, and/or undesired). As used herein, 3DOF refers to position (e.g., rotation) information associated with rotational axes about three perpendicular directional axes (e.g., pitch, yaw, and roll).

[0066] The inertial tracking components/system of the HMD 100 (i.e., the accelerometer(s) 155, gyroscope(s) 160, and compass(es) 165) may operate in concert with a visual tracking system to form a head tracking system that generates pose data for the HMD 100. In some instances, a visual tracking system includes one or more cameras (e.g., head tracking cameras) that capture image data of an environment (e.g., environment 175). In some instances, the HMD 100 obtains visual tracking data based on the images captured by the visual tracking system, such as objects within the environment that may provide an anchor for determining movement of the HMD 100 relative to the environment.

[0067] For example, visual-inertial Simultaneous Location and Mapping (SLAM) in an HMD 100 fuses (e.g., with a pose filter) visual tracking data obtained by one or more cameras (e.g., head tracking cameras) with inertial tracking data obtained by the accelerometer(s) 155, gyroscope(s) 160, and compass(es) 165 to estimate six degree of freedom (6DOF) positioning (i.e., pose) of the HMD 100 in space and in real time. 6DOF refers to positioning/velocity information associated with three perpendicular directional axes and the three rotational axes (often referred to as pitch, yaw, and roll) about each of the three perpendicular directional axes (often referred to as x, y, and z).

[0068] Unless otherwise specified, any reference herein to a “pose” or a related term describing positioning and/or orientation may refer to 3DOF or 6DOF pose.

[0069] The visual tracking system of an HMD 100, in some instances, includes a stereo

pair of head tracking images that is configured to obtain depth maps of the user's environment (e.g., environment 175) to provide visual mapping of the user's environment (e.g., by maintaining a surface mesh of the environment, or any other 3D representation of the environment). The HMD 100 may utilize the visual mapping data of the environment to accurately display virtual content with respect to the user's environment. Visual mapping data may also enable location sharing between users in a shared mixed-reality environment.

5 [0070] In some instances, the visual tracking system(s) of an HMD 100 (e.g., head tracking cameras) is/are implemented as one or more dedicated cameras. In other instances, the visual tracking system(s) is/are implemented as part of a camera system that performs other functions (e.g., as part of one or more cameras of the scanning sensor(s) 105, described hereinbelow).

10 [0071] The scanning sensor(s) 105 comprise any type of scanning or camera system, and the HMD 100 can employ the scanning sensor(s) 105 to scan environments, map environments, capture environmental data, and/or generate any kind of images of the environment. For example, in some instances, the HMD 100 is configured to generate a 3D representation of the real-world environment or generate a "pass-through" visualization. Scanning sensor(s) 105 may comprise any number or any type of scanning devices, without limit.

15 [0072] In accordance with the disclosed embodiments, the HMD 100 may be used to generate a parallax-corrected pass-through visualization of the user's environment. A "pass-through" visualization refers to a visualization that presents one or more images captured by cameras to a user, regardless of whether the HMD 100 is included as a part of an AR system or a VR system. To generate this passthrough visualization, the HMD 100 may use its scanning sensor(s) 105 to scan, map, or otherwise record its surrounding environment, including any objects in the environment, and to pass that data on to the user to view. In many cases, the passed-through data is modified to reflect or to correspond to a perspective of the user's pupils. The perspective may be determined by any type of eye tracking technique. In some instances, as the camera modules are not telecentric with the user's eyes, the perspective difference between the user's eyes and the camera modules may be corrected to provide parallax-corrected pass-through visualizations.

20 [0073] To generate a parallax-corrected passthrough image, the scanning sensor(s) 105 may rely on its cameras (e.g., visible light camera(s) 110, low light camera(s) 115, thermal imaging camera(s) 120, UV camera(s) 125, or any other type of camera) to obtain one or more raw images of the environment (e.g., environment 175). In some instances, these raw

images may also be used to determine depth data detailing the distance from the sensor to any objects captured by the raw images (e.g., a z-axis range or measurement). Once these raw images are obtained, then a depth map can be computed from the depth data embedded or included within the raw images, and passthrough images can be generated (e.g., one for
5 each pupil) using the depth map for any reprojections.

[0074] As used herein, a “depth map” details the positional relationship and depths relative to objects in the environment. Consequently, the positional arrangement, location, geometries, contours, and depths of objects relative to one another can be determined. From the depth maps (and possibly the raw images), a 3D representation of the environment can
10 be generated. Unless otherwise specified, the terms “depth map” and “disparity map” are used interchangeably herein.

[0075] Relatedly, from the passthrough visualizations, a user will be able to perceive what is currently in the user’s environment without having to remove or reposition the HMD 100. Furthermore, the disclosed passthrough visualizations may also enhance the user’s
15 ability to view objects within his/her environment (e.g., by displaying additional environmental conditions that may not have been detectable by a human eye).

[0076] It should be noted that while a portion of this disclosure focuses on generating “a” passthrough image, the implementations described herein may generate a separate passthrough image for each one of the user’s eyes. That is, two passthrough images may be
20 generated concurrently with one another. Therefore, while frequent reference is made to generating what seems to be a single passthrough image, the implementations described herein are actually able to simultaneously generate multiple passthrough images.

[0077] In some embodiments, scanning sensor(s) 105 include cameras of various modalities, such as visible light camera(s) 110, low light camera(s) 115, thermal imaging camera(s) 120, Near Infrared (NIR) Cameras (in the 800nm to 2um range), and/or
25 potentially (though not necessarily) ultraviolet (UV) cameras 125. The ellipsis 130 demonstrates how any other type of camera or camera system (e.g., depth cameras, time of flight cameras, etc.) may be included among the scanning sensor(s) 105. As an example, a camera structured to detect mid-infrared wavelengths may be included within the scanning
30 sensor(s) 105.

[0078] Generally, a human eye is able to perceive light within the so-called “visible spectrum,” which includes light (or rather, electromagnetic radiation) having wavelengths ranging from about 380 nanometers (nm) up to about 740 nm. In some instances, the visible light camera(s) 110 include monochrome cameras structured to capture light photons within

the visible spectrum (and/or, in some instances, the infrared spectrum). In some instances, the visible light camera(s) 110 include red, green, blue (RGB) cameras structured to capture light photons within the visible spectrum (and/or, in some instances, the infrared spectrum). In some implementations, visible light camera(s) 110 are complementary metal-oxide-semiconductor (CMOS) type cameras, though other camera types may be used as well (e.g., charge coupled devices, CCD).

[0079] Visible light camera(s) 110 may be implemented as stereoscopic cameras, meaning that the fields of view of two or more visible light cameras 110 at least partially overlap with one another. With this overlapping region, images generated by the visible light camera(s) 110 can be used to identify disparities between certain pixels that commonly represent an object captured by both images. Disparities are typically measured after applying rectification to the stereo pair of images such that corresponding pixels in the images that commonly represent an object in the environment are aligned along scanlines. After rectification, corresponding pixels in the different images that commonly represent an object in the environment only differ in one dimension (e.g., the direction of the scanlines, such as the horizontal direction). The one-dimensional difference between the coordinates of corresponding pixels in their respective images of the stereo pair of images represents the disparity value for the object represented by the corresponding pixels.

[0080] Based on these pixel disparities, the embodiments are able to determine depths for objects located within the overlapping region (i.e. “stereoscopic depth matching,” “stereo depth matching,” or simply “stereo matching”). The depths for the objects/3D points of the environment located within the overlapping region may be represented as pixels of a depth map. As such, the visible light camera(s) 110 can be used to not only generate passthrough visualizations, but they can also be used to determine object depth. depth information about the real-world environment surrounding the mixed-reality system may enable the system to accurately present mixed-reality content (e.g., holograms) with respect to real-world objects. As an illustrative example, a depth system may obtain depth information for a real-world table positioned within a real-world environment. The mixed-reality system is then able to render and display a virtual figurine accurately positioned on the real-world table such that the user perceives the virtual figurine as though it were part of the user’s real-world environment.

[0081] Those skilled in the art will recognize, in view of the present disclosure, that stereo matching may be performed on a stereo pair of images obtained by any type and/or combination of cameras. For example, an HMD 100 or other system may comprise any

combination of visible light camera(s) 110, low light camera(s) 115, thermal imaging camera(s) 120, UV camera(s) 125, Near Infrared Red, and/or other cameras to capture a stereo pair of images upon which to perform stereo matching (e.g., for the overlapping region of the stereo pair of images).

5 **[0082]** In some instances, the low light camera(s) 115 are structured to capture visible light and IR light. IR light is often segmented into three different classifications, including near-IR, mid-IR, and far-IR (e.g., thermal-IR). The classifications are determined based on the energy of the IR light. By way of example, near-IR has relatively higher energy as a result of having relatively shorter wavelengths (e.g., between about 750 nm and about 1,000
10 nm). In contrast, far-IR has relatively less energy as a result of having relatively longer wavelengths (e.g., up to about 30,000 nm). Mid-IR has energy values in between or in the middle of the near-IR and far-IR ranges. In some instances, the low light camera(s) 115 are structured to detect or be sensitive to IR light in at least the near-IR range.

[0083] In some embodiments, the visible light camera(s) 110 and the low light
15 camera(s) 115 operate in approximately the same overlapping wavelength range. In some cases, this overlapping wavelength range is between about 400 nanometers and about 1,000 nanometers. Additionally, in some embodiments these two types of cameras are both silicon detectors.

[0084] In some instances, one distinguishing feature between these two types of cameras
20 is related to the illuminance conditions or illuminance range(s) in which they actively operate. In some cases, the visible light camera(s) 110 are low power cameras and operate in environments where the illuminance is between about 10 lux and about 100,000 lux (e.g., for an example commercial visible light camera), or rather, the illuminance range begins at about 10 lux and increases beyond 10 lux. In contrast, the low light camera(s) 115 consume
25 more power and operate in environments where the illuminance range is between overcast starlight and dusk lighting levels. In some instances, the device operates in environments between starlight conditions (e.g., about 1 milli-lux, for a typical commercial low light camera) and dusk conditions (e.g., about 10 lux, for a typical commercial low light camera).

[0085] The thermal imaging camera(s) 120, in some instances, are structured to detect
30 electromagnetic radiation or IR light in the far-IR (i.e. thermal-IR) range, though some implementations also enable the thermal imaging camera(s) 120 to detect radiation in the mid-IR range. To clarify, the thermal imaging camera(s) 120 may be a long wave infrared imaging camera structured to detect electromagnetic radiation by measuring long wave infrared wavelengths. Often, the thermal imaging camera(s) 120 detect IR radiation having

wavelengths between about 8 microns and 14 microns. Because the thermal imaging camera(s) 120 detect far-IR radiation, the thermal imaging camera(s) 120 can operate, in some instances, in any illuminance condition.

[0086] In some cases (though not necessarily all), the thermal imaging camera(s) 120 include an uncooled thermal imaging sensor. An uncooled thermal imaging sensor uses a specific type of detector design that is based on a bolometer, which is a device that measures the magnitude or power of an incident electromagnetic wave / radiation. To measure the radiation, the bolometer uses a thin layer of absorptive material (e.g., metal) connected to a thermal reservoir through a thermal link. The incident wave strikes and heats the material. In response to the material being heated, the bolometer detects a temperature-dependent electrical resistance. Changes to environmental temperature cause changes to the bolometer's temperature, and these changes can be converted into an electrical signal to thereby produce a thermal image of the environment. In accordance with at least some of the disclosed embodiments, the uncooled thermal imaging sensor is used to generate any number of thermal images. The bolometer of the uncooled thermal imaging sensor can detect electromagnetic radiation across a wide spectrum, spanning the mid-IR spectrum, the far-IR spectrum, and even up to millimeter-sized waves.

[0087] The UV camera(s) 125 are structured to capture light in the UV range. The UV range includes electromagnetic radiation having wavelengths between about 10 nm and about 400 nm. The disclosed UV camera(s) 125 should be interpreted broadly and may be operated in a manner that includes both reflected UV photography and UV induced fluorescence photography.

[0088] In some instances, visible light cameras are cameras that are used for computer vision to perform head tracking (e.g., as described hereinabove). These cameras can detect visible light, or even a combination of visible and IR light (e.g., a range of IR light). In some cases, these cameras are global shutter devices with pixels being about 3 μm in size.

[0089] Low light cameras, in some instances, are cameras that are sensitive to visible light and near-IR. These cameras are larger and may have pixels that are about 5 μm in size or larger. These cameras are also sensitive to wavelengths that silicon sensors are sensitive to, which wavelengths are between about 350 nm to 1100 nm.

[0090] In some implementations, thermal / long wavelength IR devices (i.e. thermal imaging cameras) have pixel sizes that are about 10 μm or larger and detect heat radiated from the environment. These cameras may be sensitive to wavelengths in the 8 μm to 14 μm range. Some embodiments also include mid-IR cameras configured to detect at least

mid-IR light. These cameras often comprise non-silicon materials (e.g., InP-based InGaAs cameras) that detect light in the 800 nm to 2 μ m wavelength range.

[0091] Generally, the low light camera(s) 115, the thermal imaging camera(s) 120, and the UV camera(s) 125 (if present) consume relatively more power than the visible light camera(s) 110. Therefore, when not in use, the low light camera(s) 115, the thermal imaging camera(s) 120, and/or the UV camera(s) 125 are typically in the powered-down state in which those cameras are either turned off (and thus consuming no power) or in a reduced operability mode (and thus consuming substantially less power than if those cameras were fully operational). In contrast, the visible light camera(s) 110 are typically in the powered-up state in which those cameras are by default fully operational.

[0092] It should be noted that any number of cameras may be provided on the HMD 100 for each of the different camera types. That is, the visible light camera(s) 110 may include 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, or more than 10 cameras. Often, however, the number of cameras is at least 2 so the HMD 100 can perform stereoscopic depth matching, as described earlier. Similarly, the low light camera(s) 115, the thermal imaging camera(s) 120, and the UV camera(s) 125 may each respectively include 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, or more than 10 corresponding cameras.

Continuous Image Alignment of Separate Cameras

[0093] Attention is now directed to Figure 2, which illustrates an example head-mounted display (HMD) 200 and a user instrument 250 that include various cameras that may facilitate the disclosed embodiments. The HMD 200 may correspond, in at least some respects, to the HMD 100 described hereinabove with reference to Figure 1. The user instrument 250 may comprise any type of handheld and/or wearable device that is usable in conjunction with the HMD 200 (or another system associated with match camera(s) 215). For example, in some instances, a user instrument 250 is a controller, a medical/dental instrument, a first responder tool, etc.

[0094] The HMD 200 includes match camera(s) 215, and the user instrument 250 includes a reference camera 260. The match camera(s) 215 and/or the reference camera 260 may be implemented as cameras of any modality (e.g., any combination of visible light camera(s) 110, low light camera(s) 115, thermal imaging camera(s) 120, UV camera(s) 125, Near Infrared Red, and/or other cameras). In some implementations, the match camera(s) 215 and the reference camera 260 share the same camera modality, whereas in other implementations, the match camera(s) 215 and the reference camera 260 have different camera modalities.

[0095] Figure 2 depicts the HMD 200 with two match cameras 215. In some implementations, the two match cameras 215 are substantially vertically aligned the eyes of a user operating the HMD 200. In some instances, an HMD 200 with two match cameras 215 may enable the HMD 200 to provide per-eye parallax-corrected images (e.g., based on composite images, as described hereinafter with reference to at least Figures 10 and 17).
5 However, those skilled in the art will recognize, in view of the present disclosure, that the particular configuration of the HMD 200 and/or the user instrument 250 depicted in Figure 2 is illustrative only and non-limiting. For example, in some instances, an HMD 200 includes one or more than two match cameras 215, and/or a user instrument 250 includes
10 more than one reference camera 260.

[0096] Figure 2 also illustrates that, in some instances, the HMD 200 includes other cameras 220 that may serve various functions, such as head tracking, hand/object tracking, video capture, etc. The HMD 200 also includes displays 225A and 225B for displaying virtual content (e.g., holograms, composite images, etc.) to a user wearing the HMD 200.

[0097] The HMD 200 of Figure 2 includes a match camera inertial measurement unit (IMU) 235, and the user instrument 250 of Figure 2 includes a reference camera IMU 265. The match camera IMU 235 includes any combination of accelerometer(s) 155, gyroscope(s) 160, and/or compass(es) for generating inertial tracking data 240 (as described hereinabove). In some instances, the match camera IMU 235 is mounted to the HMD 200
20 at a fixed position relative to the match camera(s) 215, such that the inertial tracking data 240 may be associated with the match camera(s) 215.

[0098] Similarly, the reference camera IMU 265 includes any combination of accelerometer(s), gyroscope(s), and/or compass(es) for generating inertial tracking data 270 that may be associated with the reference camera 260 (e.g., where the reference camera IMU
25 265 is mounted to the user instrument 250 at a fixed position relative to the reference camera 260).

[0099] Figure 2 also illustrates that, in some implementations, the HMD 200 and the user instrument 250 are configured to share data through a wireless link 290. In one example, the user instrument 250 may transmit image data for image frames captured by the reference
30 camera 260 to the HMD 200 through a wireless channel. It should be noted that the wireless link 290 may implement various wireless communication technologies, such as ultra-wideband, WLAN, infrared communication, Bluetooth, and/or others.

[00100] Figure 3 illustrates an example of capturing an environment that includes a physical object 305 with a reference camera 260 and a match camera 215. In particular,

Figure 3 illustrates the reference camera 260 of the user instrument 250 capturing a base reference frame 310 at a base reference camera pose 320 and base reference camera timepoint 325. As illustrated in Figure 3, the base reference frame 310 includes a representation of the physical object 305. Similarly, Figure 3 illustrates the match camera 215 of the HMD 200 capturing a base match frame 315 at a base match camera pose 330 and base match camera timepoint 335. The base match frame 315 also includes a representation of the physical object 305.

[00101] As will be described hereinbelow, the base reference frame 310 and the base match frame 315 may provide a basis for generating a motion model for facilitating mapping of imagery captured by the reference camera onto imagery captured by the match camera (or vice versa).

[00102] Figure 4 illustrates an example of performing feature matching between the base reference frame 310 and the base match frame 315. In some implementations, performing feature matching involves identifying feature points and feature descriptors within the base reference frame 310 and the base match frame 315. In some instances, a feature point (sometimes referred to as “keypoints,” “points of interest,” or simply “features”) refers to a pixel within an image that comprises rich texture information, such as edges, corners, and/or other readily identifiable structures. In some instances, a feature descriptor (also referred to as a “feature vector”) results from extracting image data/statistics from a local image/pixel patch around an identified feature point. A feature descriptor may operate as an identifier for the feature point about which the feature descriptor is centered. Various approaches exist for extracting feature descriptors, such as local histogram approaches, N-jets approaches, and/or others. For example, a feature descriptor may be identified based on a histogram of gradient magnitudes (e.g., changes in intensity and/or color) and/or orientations (e.g., edge orientations) for pixels within an image patch centered on a feature point.

[00103] A system may employ various techniques for identifying feature points and/or feature descriptors, such as, by way of non-limiting example, scale-invariant feature transform (SIFT), speeded up robust features (SURF), Canny operator, Kayyali operator, Moravec algorithm, Harris & Stephens / Shi-Tomasi algorithms, Förstner detector, smallest univalue segment assimilating nucleus (SUSAN) detector, level curve curvature approach, DAISY algorithms, and/or others.

[00104] Figure 4 illustrates various feature points associated with the physical object 305 as represented in the base reference frame 310 and the base match frame 315. For example, Figure 4 illustrates feature points 410A, 410B, and 410C on the edges of the depiction of

the physical object 305 within the base match frame 315. Figure 4 also depicts feature descriptors 415A, 415B, and 415C that are associated, respectively, with feature points 410A, 410B, and 410C.

5 [00105] Figure 4 also illustrates various feature points associated with the depiction of the physical object 305 within the base reference frame 310. For example, Figure 4 illustrates feature points 420A, 420B, and 420C on the edges of the depiction of the physical object 305 within the base reference frame 310. Figure 4 also depicts feature descriptors 425A, 425B, and 425C that are associated, respectively, with feature points 420A, 420B, and 420C.

10 [00106] As illustrated in Figure 4, the feature points 410A, 410B, and 410C within the base match frame 315 correspond to the feature points 420A, 420B, and 420C within the base reference frame 310. As used herein, feature points “correspond” to one another when they represent the same 3D point within a captured environment. For example, feature point 410A within the base match frame 315 and feature point 420A within the base reference
15 frame 310 both represent the same 3D point along the left edge of the physical object 305.

[00107] As used herein, a “feature match” comprises a feature point in one image (e.g., the base reference frame 310) and a corresponding feature point in another image (e.g., the base match frame 315). In some instances, a system identifies feature matches by comparing feature descriptors of the features identified in the images. A system may employ various
20 techniques to identify feature matches between the features of the base match frame 315 and the base reference frame 310, such as a brute-force matcher, a fast library for approximate nearest neighbors (FLANN) matcher, and/or others. Figure 4 illustrates three feature matches (e.g., feature points 410A and 420A, feature points 410B and 420B, and feature points 410C and 420C), and the ellipses 450 and 460 indicate that a system may identify
25 any number of feature matches within the base reference frame 310 and the base match frame 315 (e.g., in some instances, on the order of 100 matches).

[00108] In some instances, a system utilizes feature matches identified within the base match frame 315 and the base reference frame 310 to determine a motion model for facilitating mapping of imagery captured by the reference camera onto imagery captured by
30 the match camera (or vice versa). Figure 5 illustrates an example of unprojecting the feature matches identified within the base match frame 315 and the base reference frame 310.

[00109] Initially, Figure 5 illustrates a colocation point 500, which illustrates an assumption that the reference camera 260 and the match camera 215 were at a same location while capturing the base match frame 315 and the base reference frame 310 (e.g., the

separation distance between the reference camera 260 and the match camera 215 is set to zero). However, while the colocation point 500 illustrates an assumption that the reference camera 260 and the match camera 215 shared a same location while capturing images, the colocation point 500 does not constrain the orientation of the reference camera 260 and the match camera 215. For example, Figure 5 depicts the reference camera 260 at the colocation point 500 with its orientation determined by the base reference camera pose 320, and Figure 5 also depicts the match camera 215 at the colocation point 500 with its orientation determined by the base match camera pose 330.

[00110] Figure 5 illustrates generating unprojected feature points 520A, 520B, and 520C by performing unprojection 530 on the feature points 420A, 420B, and 420C of the base reference frame 310. Figure 5 also illustrates generating unprojected feature points 510A, 510B, and 510C by performing unprojection 540 on the feature points 410A, 410B, and 410C of the base match frame 315. The unprojected feature points are 3D points that are generated based on the 2D feature points.

[00111] In some instances, generating 3D unprojected feature point (e.g., 510A–510C, 520A–520C) by performing unprojection (e.g., unprojection 540, 530) on a 2D feature point (e.g., feature points 410A–410C, 420A–420C) may be conceptualized as extending a ray from a camera center or optical center using a pixel location of the feature point to be unprojected. In one example for generating unprojected feature point 520A, using pinhole camera terminology for illustrative purposes, a system may define the colocation point 500 as an optical center or camera center of the reference camera 260 while the reference camera captured the base reference frame 310 at the base reference camera pose 320. Continuing with the example, the system may then cast a ray from the colocation point 500 using the 2D pixel coordinates of feature point 420A as feature point 420A lies on a front image plane positioned about the colocation point 500. The front image plane may be positioned about the colocation point 500 according to the base reference camera pose 320 (e.g., wherein the base reference camera pose 320 provides the orientation of the front image plane with respect to the colocation point 500). The system may extend the ray to a particular depth value to provide the 3D unprojected feature point 520A. A system may use any depth value for the unprojection 530, such as a uniform depth value for generating all unprojected feature points.

[00112] The ellipses 550 and 560 of Figure 5 indicate that a system may perform unprojection 530, 540 to generate any number of 3D unprojected feature points based on the 2D feature points of the base reference frame 310 and the base match frame 315. As

noted hereinabove, at least some of the feature points of the base reference frame 310 (e.g., feature points 420A–420C) may correspond to feature points of the base match frame 315 (e.g., feature points 410A–410C), providing feature matches between the base reference frame 310 and the base match frame 315. Therefore, Figure 5 illustrates an instance in which performing unprojection 530, 540 on the feature points of the base reference frame 310 and the base match frame 315 provides unprojected feature matches 570 of corresponding unprojected feature points. For example, unprojected feature points 510A and 520A may correspond to one another, being generated by unprojecting feature points 410A and 420A, respectively.

10 **[00113]** Figure 6 illustrates an example of identifying a base matrix 610 using the unprojected feature matches 570. In some implementations, a base matrix 610 represents a 3D rotation matrix, or a 3x3 matrix that describes rotation about 3 perpendicular axes to rotate a set of points from one coordinate system into a different coordinate system. In some instances, as indicated by Figure 6, the base matrix 610 is a 3D rotation matrix that facilitates rotation of the unprojected feature points 520A–520C into the coordinate system of the corresponding unprojected feature points 510A–510C (the corresponding unprojected feature points 510A–510C having been unprojected using the base match camera pose 330). Put differently, the base matrix 610 is a 3D rotation matrix that rotates a set of unprojected feature points (e.g., 520A–520C) onto a set of corresponding unprojected feature points (e.g., 510A–510C).

20 **[00114]** By way of illustration, Figure 6 depicts the reference camera 260 positioned at the colocation point 500 with an orientation that corresponds to the base reference camera pose 320. It should be noted that the base reference camera pose 320 is the pose from which the reference camera 260 captured the base reference frame 310 that provided the 2D feature points 420A–420C for generating unprojected feature points 520A–520C. Figure 6 illustrates that the base matrix 610 may be described as a 3D rotation matrix that facilitates rotation of the reference camera 260 to correspond to the base match camera pose 330, which is the pose from which the match camera 215 captured the base match frame 315 that provided the 2D feature points 410A–410C for generating unprojected feature points 510A–510C.

25 **[00115]** A system may identify a base matrix 610 using unprojected feature matches 570 in a variety of ways, such as by minimizing a cost function (e.g., according to the Wahba method), utilizing machine learning techniques, and/or other approaches.

30 **[00116]** Using a base matrix 610, a system may map pixels from the base reference frame

310 onto corresponding pixels of the base match frame 315, as discussed in more detail with reference to Figures 7–10. Figure 7 illustrates generating 3D points 740 by performing unprojection 730 on a set of pixels of the base reference frame 310 captured by the reference camera 260 at the base reference camera pose 320. Unprojection 730 corresponds to
5 unprojection 530, 540 described hereinabove with reference to Figure 5. In some instances, a system performs unprojection 730 on pixels of the base reference frame 310 using the intrinsic matrix (e.g., intrinsic calibration) of the reference camera 260 and a uniform depth value. Ellipsis 750 indicates that a system may perform unprojection 730 on any number of pixels of the base reference frame 310 to generate the 3D points 740.

10 **[00117]** Figure 8 illustrates an example of generating modified 3D points 860 by applying the base matrix 610 to the 3D points 740. In some instances, applying the base matrix 610 to the 3D points 740 rotates the 3D points 740 into a different coordinate system, thereby providing modified 3D points 860. By way of illustration, applying the base matrix 610 to the 3D points 740 may be thought of as rotating the 3D points 740 into a coordinate system
15 that would exist for 3D points unprojected using the base match frame 315 and the base match camera pose 330. Ellipsis 850 indicates that a system may apply the base matrix 610 to any number of the 3D points 740 to generate the modified 3D points 860.

[00118] Figure 9 illustrates an example of performing projection 930 on the modified 3D points 860. In some instances, projection 930 is an operation that is inverse to the
20 unprojection operations described herein (e.g., unprojection 530, 540, 730). Projection 930 may be conceptualized as extending a ray from a 3D point (e.g., one of the modified 3D points 860) toward a camera center or optical center. For instance, again using pinhole camera terminology for illustrative purposes, a system may define the colocation point 500 as an optical center or camera center of the match camera 215 while the match camera
25 captured the base match frame 315 at the base match camera pose 330. A system may then cast a ray from a 3D point (e.g., one of the modified 3D points 860) toward the colocation point 500, and the ray may identify a 2D pixel as the pixel lies on a front image plane positioned about the colocation point. The front image plane may be positioned about the colocation point according to an intrinsic matrix (e.g., intrinsic calibration) of the match
30 camera 215 and according to the base match camera pose 330 (e.g., wherein the base match camera pose 330 provides the orientation of the front image plane with respect to the colocation point 500). The system may then associate the projected 3D point with the 2D pixel coordinates identified according to the projection 930 (and thereby associate 2D pixel coordinates with the pixels of the base reference frame 310 that were originally unprojected

according to unprojection 730). The ellipsis 950 indicates that a system may perform projection 930 on any number of the modified 3D points 860 to associate the modified 3D points with 2D pixel coordinates. The 2D pixel coordinates may identify a set of corresponding pixels in the base match frame 315 to which a system may map of pixels from the base reference frame 310.

[00119] In some instances, a system uses the 2D pixel coordinates associated with the modified 3D points 860 to identify a set of corresponding pixels in the base match frame 315 to which the system may map the set of pixels of the base reference frame 310 that were originally unprojected according to unprojection 730. Figure 10 illustrates an example of a composite frame 1010 that includes base reference frame pixels 1030 mapped to corresponding base match frame pixels 1040 in an overlap region 1020 of the composite frame 1010. For example, in some instances, the system applies texture information from the base reference frame pixels 1030 to the corresponding base match frame pixels 1040 that were identified according to the projections 930 of the modified 3D points 860.

[00120] Accordingly, Figures 7–9 illustrate facilitating image alignment of a base reference frame and a base match frame using a 3D rotational matrix (i.e., base matrix 610), which is a simplification made possible by the colocation assumption described hereinabove and exemplified by colocation point 500 from Figures 5–9. The colocation assumption may reduce the computation burden associated with continuously mapping imagery from one camera onto imagery of another camera. The colocation assumption ignores real-world parallax that may exist between the perspective of the match camera and the perspective of the reference camera. However, in some implementations, the effects of parallax are only observable when the cameras capture objects that are relatively close to the cameras, and/or when the separation distance between the cameras is relatively large.

[00121] Although the foregoing description focuses, in some respects, on implementations that use the colocation assumption, those skilled in the art will recognize, in view of the present disclosure, that at least some of the principles described herein are applicable in implementations that omit the colocation assumption.

[00122] In some instances, the operations of generating a set of 3D points 740 by unprojecting a set of pixels of the base reference frame 310, generating modified 3D points 860 by applying the base matrix 610 to the 3D points 740, and projecting the modified 3D points 860 as described hereinabove with reference to Figures 7–9 map the set of pixels of the base reference frame 310 onto a set of corresponding pixels in the base match frame 315. These operations may be thought of as a model that captures the relative positioning of

the reference camera 260 and the match camera 215 to enable mapping of pixels from a base reference frame to a base match frame. The model may be represented by the following equation:

$$f(p) = K_{match} * R_{ref \rightarrow match} * K_{ref}^{-1} \quad (1)$$

[00123] Where $f(p)$ is a function that maps pixels p of the reference frame to pixels p' of the match frame. K_{ref} represents the intrinsic matrix of the reference camera, and K_{match} represents the intrinsic matrix of the match camera. $R_{ref \rightarrow match}$ represents an alignment matrix, which may be implemented as the base matrix described hereinabove (or other transformations, such as homography, similarity transforms, affine motion models, etc.). In some instances, Equation 1 represents unprojection of a pixel p of the reference frame using the intrinsic matrix of the reference camera, K_{ref} , rotating the unprojected point using the base matrix, $R_{ref \rightarrow match}$, and projecting the rotated point onto the match frame using the intrinsic matrix of the match camera, K_{match} .

[00124] Attention is now directed to Figure 11, which illustrates an example of the reference camera 260 of the user instrument 250 and the match camera of the HMD 200 continuously capturing the environment that includes the physical object 305 as the poses of the user instrument 250 and the HMD 200 change over time. Specifically, Figure 11 illustrates the reference camera 260 of the user instrument 250 capturing an updated reference frame 1110 at an updated reference camera pose 1120 and at an updated reference camera timepoint 1125. The updated reference camera pose 1120 is different than the base reference camera pose 320, and the updated reference camera timepoint 1125 is subsequent to the base reference camera timepoint 325. Similarly, Figure 11 illustrates the match camera 215 of the HMD 200 capturing an updated match frame 1115 at an updated match camera pose 1130 and at an updated match camera timepoint 1135. The updated match camera pose 1130 is different than the base match camera pose 330, and the updated match camera timepoint 1135 is subsequent to the base match camera timepoint 335.

[00125] In some instances, the difference between the base reference camera pose 320 and the updated reference camera pose 1120 is captured by the reference camera IMU 265 in the form of a reference camera transformation matrix 1170. In some implementations, the reference camera transformation matrix 1170 is a 3D rotational matrix identified based on inertial tracking data 270 obtained by the reference camera IMU 265. Similarly, the difference between the base match camera pose 330 and the updated match camera pose 1130 is captured by the match camera IMU 235 in the form of a match camera

transformation matrix 1140. In some implementations, the match camera transformation matrix 1140 is also a 3D rotational matrix identified based on inertial tracking data 240 obtained by the match camera IMU 235.

5 [00126] As is evident in Figure 11, because of the pose change of the reference camera 260, the position of the depiction of the physical object 305 within the updated reference frame 1110 is different when compared with the position of the depiction of the physical object 305 within the base reference frame 310. Similarly, because of the pose change of the match camera 215, the position of the depiction of the physical object 305 within the updated match frame 1115 is different when compared with the position of the depiction of the physical object 305 within the base match frame 315. Thus, the base matrix 610, if used alone, may fail to accurately map a set of pixels of the updated reference frame 1110 to a set of corresponding pixels of the updated match frame 1115 because of the pose differences of the cameras between the base timepoint(s) (e.g., the base reference camera timepoint 325 and the base match camera timepoint 335) and the updated timepoint(s) (e.g., the updated reference camera timepoint 1125 and the updated match camera timepoint 1135).

15 [00127] However, in some instances, utilizing the reference camera transformation matrix 1170 and/or the match camera transformation matrix 1140 in combination with the base matrix 610 may enable the system to map a set of pixels of the updated reference frame 1110 to a set of corresponding pixels of the updated match frame 1115.

20 [00128] For example, Figure 12 illustrates an example of generating 3D points 1240 by performing unprojection 1230 on a set of pixels of the updated reference frame 1110 captured by the reference camera 260 at the updated reference camera pose 1120. In some instances, unprojection 1230 corresponds to unprojection 730 described hereinabove with reference to Figure 7. For example, in some instances, a system performs unprojection on the set of pixels of the updated reference frame 1110 from the collocation point 1200 using the updated reference camera pose 1120 of the reference camera 260. The ellipsis 1250 indicates that a system may perform unprojection 1230 on any number of pixels of the updated reference frame 1110 to generate the 3D points 1240.

25 [00129] Figure 13 illustrates an example of generating modified 3D points 1360 by applying the reference camera transformation matrix 1170 to the 3D points 1240. In some instances, applying the reference camera transformation matrix 1170 rotates the 3D points 1240 into the coordinate system that existed for the 3D points 740 generated by performing unprojection 730 using the base reference frame 310 and the base reference camera pose 320. For illustrative purposes, Figure 13 depicts the reference camera 260 positioned at the

colocation point 1200 according to the base reference camera pose 320. Accordingly, applying the reference camera transformation matrix 1170 to the 3D points 1240 to generate the modified 3D points 1360 may be thought of as restoring the pose of the reference camera 260 that existed while the reference camera 260 captured the base reference frame 310 (e.g.,
5 base reference camera pose 320). The ellipsis 1350 indicates that a system may apply the reference camera transformation matrix 1170 to any number of 3D points 1240 to generate the modified 3D points 1360.

[00130] Figure 14 illustrates an example of generating modified 3D points 1460 by applying the base matrix 610 to the modified 3D points 1360. In some instances, related to
10 applying the base matrix 610 to the 3D points 740 described hereinabove with reference to Figure 7, applying the base matrix 610 to the modified 3D points 1360 rotates the modified 3D points into a coordinate system that would exist for 3D points unprojected using the base match frame 315 and the base match camera pose 330 (which may be the same coordinate system that existed for the modified 3D points 860 described hereinabove with reference to
15 Figure 8). For illustrative purposes, Figure 14 depicts the match camera 260 positioned at the colocation point 1200 according to the base match camera pose 330. The ellipsis 1450 indicates that a system may apply the base matrix 610 to any number of modified 3D points 1360 to generate the modified 3D points 1460.

[00131] Figure 15 illustrates an example of generating modified 3D points 1560 by
20 applying the match camera transformation matrix 1140 to the modified 3D points 1460. In some instances, applying the match camera transformation matrix 1140 rotates the modified 3D points 1460 into a coordinate system that would exist for 3D points unprojected using the updated match frame 1115 and the updated match camera pose 1130. The ellipsis 1550 indicates that a system may apply the match camera transformation matrix 1140 to any
25 number of modified 3D points 1460 to generate the modified 3D points 1560.

[00132] Figure 16 illustrates an example of performing projection 1630 on the modified 3D points 1560. In some instances, projection 1630 corresponds to projection 930 described hereinabove with reference to Figure 8. For example, in some instances, a system performs projection 1630 on the modified 3D points 1560 toward the colocation point 1200 using the
30 updated match camera pose 1130 of the match camera 215 to associate 2D pixel coordinates with the projected modified 3D points 1560 (and thereby associated 2D pixel coordinates with the pixels of the updated reference frame 1110 that were originally unprojected according to unprojection 1230).

[00133] As before, in some implementations, a system uses the 2D pixel coordinates

associated with the modified 3D points 1560 to identify a set of corresponding pixels in the updated match frame 1115 to which the system may map the set of pixels of the updated reference frame 1110 that was originally unprojected according to unprojection 1230. Figure 17 illustrates an example of a composite frame 1710 that includes updated reference frame pixels 1730 mapped to updated corresponding match frame pixels 1740 in an overlap region 1720 of the composite frame 1710. For example, in some instances, the system applies texture information from the updated reference frame pixels 1730 to the updated corresponding match frame pixels 1740 that were identified according to the projections 1630 of the modified 3D points 1560.

10 **[00134]** Accordingly, Figures 12–16 illustrate facilitating image alignment of an updated reference frame and an updated match frame using a combination of 3D rotational matrices (in particular, the base matrix 610, the reference camera transformation matrix 1170, and the match camera transformation matrix 1140). The operations described hereinabove with reference to Figures 12–16 may be thought of as a motion model that maps the relative
15 positioning of the reference camera 260 and the match camera 215 over time to enable continuous mapping of pixels from updated reference frames to updated match frames. The motion model may be represented by the following equation:

$$f(p) = K_{match} * P_{match_cur} * P_{match_base}^{-1} * R_{ref \rightarrow match} * P_{ref_base} * P_{ref_cur}^{-1} * K_{ref}^{-1} \quad (2)$$

[00135] Where $f(p)$ is a function that maps pixels p of the reference frame to pixels p' of the match frame. As before, K_{ref} represents the intrinsic matrix of the reference camera, and K_{match} represents the intrinsic matrix of the match camera. $R_{ref \rightarrow match}$ represents an
20 alignment matrix, which may be implemented as the base matrix described hereinabove. P_{ref_base} and P_{match_base} refer to the base reference camera pose 320 and the base match camera pose 330, respectively, and P_{ref_cur} and P_{match_cur} refer to the updated reference camera pose 1120 and the updated match camera pose 1130, respectively. Accordingly, in
25 some implementations, the combination of $P_{ref_base} * P_{ref_cur}^{-1}$ refers to the reference camera transformation matrix 1170 described hereinabove. Furthermore, in some instances, the combination of $P_{match_cur} * P_{match_base}^{-1}$ refers to the match camera transformation matrix 1140 described hereinabove. Accordingly, in some implementations, Equation 2 may be thought of as a motion model configured to facilitate continuous mapping of sets of pixels
30 of updated reference frames captured by the reference camera to corresponding sets of pixels of updated match frames captured by the match camera, with the motion model being based

on the base matrix, the reference camera transformation matrix, and the match camera transformation matrix.

5 **[00136]** However, inertial tracking data obtained by IMUs (e.g., inertial tracking data 240 obtained by the match camera IMU 235 and/or inertial tracking data 270 obtained by the reference camera IMU 265) may be prone to drift, which refers to accumulated errors brought about by continually integrating acceleration with respect to time. Accordingly, the accuracy of the motion model represented by Equation 2 and described hereinabove with reference to Figures 12–16 may degrade as poses continue to update from the time that the original base matrix 610 was computed.

10 **[00137]** Accordingly, in some implementations, a motion model utilizes an alignment matrix that is generated based on a previously computed base matrix and/or a current updated matrix (e.g., by fusing the two together). Similar to the base matrix 610 described hereinabove, in some instances, an updated matrix is also generated/identified using visual correspondences between frames captured by the reference camera 260 and the match camera 215. In addition to combatting drift, generating a motion model using matrices generated from visual correspondences of frame pairs captured at different timepoints may also ameliorate the effects that changes in the relative positioning of the cameras may have on composite images generated using the motion model.

15 **[00138]** Figure 18 illustrates an example of performing feature matching between the updated reference frame 1110 (captured by the reference camera 260 at updated reference camera pose 1120 and timepoint 1125, see Figure 11) and the updated match frame 1115 (captured by the match camera 215 at updated match camera pose 1130 and timepoint 1135, see Figure 11). Similar to Figure 4 described hereinabove, Figure 18 illustrates various feature points associated with the physical object 305 as represented in the updated reference frame 1110 and the updated match frame 1115. For example, Figure 18 illustrates feature points 1810A, 1810B, and 1810C on the edges of the depiction of the physical object 305 within the updated match frame 1115. Figure 18 also depicts feature descriptors 1815A, 1815B, and 1815C that are associated, respectively, with feature points 1810A, 1810B, and 1810C.

25 **[00139]** Figure 18 also illustrates various feature points associated with the depiction of the physical object 305 within the updated reference frame 1110. For example, Figure 18 illustrates feature points 1820A, 1820B, and 1820C on the edges of the depiction of the physical object 305 within the updated reference frame 1110. Figure 18 also depicts feature descriptors 1825A, 1825B, and 1825C that are associated, respectively, with feature points

1820A, 1820B, and 1820C.

[00140] As illustrated in Figure 18, the feature points 1810A, 1810B, and 1810C within the updated match frame 1115 correspond to the feature points 1820A, 1820B, and 1820C within the updated reference frame 1110. For example, feature point 1810A within the
5 updated match frame 1115 and feature point 1820A within the updated reference frame 1110 both represent the same 3D point along the left edge of the physical object 305. Each pair of feature points that correspond to one another form feature matches. The ellipses 1850 and 1860 indicate that a system may identify any number of feature matches within the updated reference frame 1110 and the updated match frame 1115.

[00141] Figure 19 illustrates an example of unprojecting the feature matches identified within the updated reference frame 1110 and the updated match frame 1115. In some instances, using the colocation point 1900 and the updated reference camera pose 1120 of the reference camera 260, a system performs unprojection 1930 on the various 2D feature points identified in the updated reference frame 1110 (e.g., feature points 1820A–1820C) to
15 generate 3D unprojected feature points (e.g., unprojected feature points 1920A–1920C). Similarly, in some instances, using the colocation point 1900 and the updated match camera pose 1130 of the match camera 215, a system performs unprojection 1940 on the various 2D feature points identified in the updated match frame 1115 (e.g., feature points 1810A–1810C) to generate 3D unprojected feature points (e.g., unprojected feature points 1910A–
20 1910C). The unprojections 1930, 1940 may correspond, in at least some respects, to the unprojections 530, 540 described hereinabove with reference to Figure 5, and performing the unprojections 1930, 1940 may provide 3D unprojected feature matches 1970. The ellipses 1950 and 1960 indicate that a system may perform any number of unprojections 1930, 1940 to generate any number of 3D unprojected feature matches 1970.

[00142] Figure 20 illustrates an example of identifying an updated matrix 2010 using the unprojected feature matches 1970. As with the base matrix 610 described hereinabove, in some instances, an updated matrix 2010 represents a 3D rotation matrix that facilitates rotation of the unprojected feature points 1920A–1920C into the coordinate system of the corresponding unprojected feature points 1910A–1910C (the corresponding unprojected
30 feature points 1910A–1910C having been unprojected using the updated match camera pose 1130). Furthermore, as with the base matrix 610, a system may identify an updated matrix 2010 using unprojected feature matches 1970 in a variety of ways, such as by minimizing a cost function (e.g., according to the Wahba method), utilizing machine learning techniques, and/or other approaches.

[00143] Although, in some implementations, a system may utilize the updated matrix 2010 to facilitate mapping of pixels of the updated reference frame 1110 to pixels of the updated match frame 1115 (e.g., by utilizing the updated matrix 2010 as $R_{ref \rightarrow match}$ in Equation 1), it should be noted that feature point identification is a noisy process. For example, the pixel coordinates of features points that describe 3D points of a captured environment may shift from expected positions across consecutively captured frames. Such occurrences may cause observable spatial flickers in overlap regions of composite frames. Furthermore, in some instances, at least some feature points that are present/identifiable in one image may not be present/identifiable in another image, even where the different images are captured using the same camera modality (e.g., because of occlusions).

[00144] Accordingly, in some implementations, a system aligns the updated matrix 2010 with a previously computed matrix (e.g., base matrix 610) to generate an alignment matrix by fusion. Utilizing such an alignment matrix in a motion model for mapping pixels from reference frames to match frames may have the effect of smoothing out noise that may otherwise affect composite frames generated using the motion model.

[00145] Figures 21 and 22 illustrate an example of generating an aligned updated matrix 2210 by modifying the updated matrix 2010 using inertial tracking data associated with the reference camera 260 and the match camera 215. As noted hereinabove, in some instances, the inertial tracking data 270 obtained by the reference camera IMU 265 tracks the pose changes of the reference camera 260 from the base reference camera timepoint 325 to the updated reference camera timepoint 1125 (e.g., by identifying the base reference camera pose 320 and the updated reference camera pose 1120). Similarly, the inertial tracking data 240 obtained by the match camera IMU 235 tracks the pose changes of the match camera 215 from the base match camera timepoint 335 to the updated match camera timepoint 1135 (e.g., by identifying the base match camera pose 330 and the updated match camera pose 1130).

[00146] In some instances, a system utilizes the inertial tracking data described above to modify the updated matrix 2010 to bring the updated matrix 2010 into the same reference/coordinate system that exists for the base matrix 610. Figure 21 illustrates a conceptual representation of a system modifying the 3D rotation described by the updated matrix 2010 by identifying a reference camera rotational matrix 2130 and a match camera rotational matrix 2150 based on the inertial tracking data that describes the motion of the reference camera 260 and the match camera 215 between capturing the base frames (e.g.,

the base reference frame 310 and the base match frame 315) and the updated frames (e.g., the updated reference frame 1110 and the updated match frame 1115). The system modifies the updated matrix 2010 with the reference camera rotational matrix 2130 and the match camera rotational matrix 2150 to generate the aligned updated matrix 2210, as shown in Figure 22. Generating the aligned updated matrix 2210 by using the reference camera rotational matrix 2130 and the match camera rotational matrix 2150 may be conceptualized as undoing the motion that occurred between when the base frames were captured (e.g., the base reference frame 310 and the base match frame 315) and when the updated frames were captured (e.g., the updated reference frame 1110 and the updated match frame 1115). Generating an aligned updated matrix 2210 may be represented by the following equation:

$$R'_{ref \rightarrow match} = P_{match_base} * P_{match_cur}^{-1} * R'_{ref_cur \rightarrow match_cur} * P_{ref_cur} * P_{ref_base}^{-1} \quad (3)$$

[00147] Where $R'_{ref \rightarrow match}$ may represent the aligned updated matrix 2210. As before, $R_{ref \rightarrow match}$ may represent the base matrix 610 described hereinabove. Furthermore, as before, P_{ref_base} and P_{match_base} refer to the base reference camera pose 320 and the base match camera pose 330, respectively, and P_{ref_cur} and P_{match_cur} refer to the updated reference camera pose 1120 and the updated match camera pose 1130, respectively. In some implementations, the combination of $P_{ref_cur} * P_{ref_base}^{-1}$ refers to the reference camera rotational matrix 2130 described hereinabove. Furthermore, in some instances, the combination of $P_{match_base} * P_{match_cur}^{-1}$ refers to the match camera rotational matrix 2150 described hereinabove.

[00148] Figure 22 also depicts the base matrix 610 and illustrates that, in some instances, although the aligned updated matrix 2210 and the base matrix 610 are computed to share the same reference/coordinate system, differences exist between the base matrix 610 and the aligned updated matrix 2210 (e.g., because of IMU drift, noise in performing feature matching, changes in relative positioning of the reference camera 260 and the match camera 215, etc.). Accordingly, in some implementations, a system fuses the base matrix 610 with the aligned updated matrix 2210 to generate an alignment matrix for a motion model to facilitate continuous image alignment of separate camera imagery.

[00149] Figure 23A illustrates an example of generating an alignment matrix 2315 using the base matrix 610 and the aligned updated matrix 2210. Figure 23A illustrates that, in some instances, a system uses the alignment updated matrix 2210 and the base matrix 610 as inputs for a fuser 2300. In the implementation depicted in Figure 23A, the fuser 2300

implements an interpolation function 2305, which, in essence, blends or combines the aligned updated matrix 2210 with the base matrix 610 to generate the alignment matrix 2315. The alignment matrix 2315 is, in some instances, also a 3D rotation matrix.

[00150] Figure 23A also illustrates that, in some instances, the interpolation function 2305 utilizes a smoothness term 2310 (or smoothness function). In some implementations, the smoothness term determines how to weight the aligned updated matrix 2210 and the base matrix 610 for generating the alignment matrix 2315. By way of example, in some implementations, when the smoothness term 2310 has a value that approaches a maximum value (e.g., a value of 1), the interpolation function 2305 increasingly ignores the aligned updated matrix 2210, giving more weight to the base matrix 610 for generating the alignment matrix 2315, which may be beneficial when the expected accuracy of the base matrix 610 is high. Furthermore, in some implementations, when the smoothness term 2310 has a value that approaches a minimum value (e.g., a value of 0), the interpolation function 2305 increasingly ignores the base matrix 610, giving more weight to the aligned updated matrix 2210 for generating the alignment matrix 2315, which may be beneficial when the expected accuracy of the base matrix 610 is low.

[00151] In some instances, the smoothness term 2310 is set to a constant value between (e.g., a value between 0 and 1, such as a value of 0.7), whereas in other instances, the smoothness term 2310 is intelligently determined/updated based on various factors (e.g., an expected accuracy of the base matrix). Additional details concerning intelligently determining a smoothness term 2310 will be provided hereinafter.

[00152] The ellipsis 2330 indicates that, in some instances, a fuser 2300 incorporates additional and/or alternative components.

[00153] Generating an alignment matrix 2315 using an aligned updated matrix 2210 and a base matrix 610 (or a previous alignment matrix) may be represented by the following equation:

$$R_{ref \rightarrow match} = interpolate(R'_{ref \rightarrow match}, R_{ref \rightarrow match}, smoothness()) \quad (4)$$

[00154] Where $R_{ref \rightarrow match}$ on the left side of Equation 4 may represent the alignment matrix 2315 described hereinabove. The function *interpolate()* may represent the interpolation function 2305 of the fuser 2300 described hereinabove, and the function *smoothness()* may represent the smoothness term 2310 described hereinabove. $R'_{ref \rightarrow match}$ may refer to the aligned updated matrix 2210, and $R_{ref \rightarrow match}$ on the right side of the equation may refer to the base matrix 610, or, in some instances, may refer to a

previously computed alignment matrix. For example, Figure 23B illustrates a fuser 2300 receiving the alignment matrix 2315 generated according to Figure 23A and a subsequent aligned updated matrix 2320 (e.g., an aligned updated matrix generated subsequent to generating the alignment matrix 2315 from Figure 23A) as inputs for generating a subsequent alignment matrix 2325. Thus, in some instances, a system utilizes an identified alignment matrix in conjunction with a subsequent updated matrix in order to identify a subsequent alignment matrix. Accordingly, generating alignment matrices may be thought of as a running average that incorporates past alignment matrices to generate new alignment matrices.

10 **[00155]** In some instances, a system utilizes the alignment matrix 2315 as part of the motion model for mapping a set of pixels from a reference frame to a corresponding set of pixels of a match frame (e.g., by utilizing $R_{ref \rightarrow match}$ on the left side of Equation 4 as $R_{ref \rightarrow match}$ in Equation 2). For example, Figure 24 illustrates an example of a motion model 2430 configured to map a reference frame 2410 onto a match frame 2420. The motion model 15 2430 of Figure 24 includes a reference camera intrinsic matrix 2435 and a match camera intrinsic matrix 2455 (e.g., to facilitate the unprojection and/or projection operations described hereinabove, such as those referred to with reference to Figures 5, 7, 9, 12, 16, and 19 and/or with reference to Equations 1 and 2). The motion model 2430 of Figure 24 also includes a match camera transformation matrix 2440 and a reference camera 20 transformation matrix 2445 (e.g., for application to 3D points and/or to modify/align updated matrices, as depicted in Figures 13, 15, and 21 and/or in Equations 2 and 3). The motion model 2430 also includes an alignment matrix 2450, which may correspond to a base matrix 610, an aligned updated matrix 2210, an alignment matrix 2315, and/or a subsequent alignment matrix 2325 described hereinabove. The ellipsis 2465 indicates that a 25 motion model 2430 may comprise any number of components, including or different than those depicted in Figure 24.

[00156] In some instances, a system utilizes the motion model 2430 to map pixels of a reference frame 2410 to pixels of a match frame 2420 in order to generate a composite image 2460. For example, in some implementations, a system generates 3D points by unprojecting 30 pixels of the reference frame 2410 using the reference camera intrinsic matrix 2435; generates modified 3D points by applying the reference camera transformation matrix 2445, the alignment matrix 2450, and the match camera transformation matrix 2440 to the 3D points; and projects the modified 3D points to generate a composite image 2460 that

overlays the pixels of the reference frame 2410 onto corresponding pixels of the match frame 2420.

Example Method(s) for Continuous Image Alignment of Separate Cameras

5 [00157] The following discussion now refers to a number of methods and method acts that may be performed. Although the method acts may be discussed in a certain order or illustrated in a flow chart as occurring in a particular order, no particular ordering is required unless specifically stated, or required because an act is dependent on another act being completed prior to the act being performed.

10 [00158] Figure 25 illustrates an example flow diagram 2500 depicting acts associated with generating a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by a reference camera to a corresponding set of pixels of a match frame captured by a match camera. The discussion of the various acts represented in flow diagram 2500 includes references to various hardware components described in more detail with reference to Figures 1, 2, and 34.

15 [00159] Act 2502 of flow diagram 2500 includes obtaining an updateable base matrix. Act 2502 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250) and using images captured by a reference camera 260 and a match camera 215. For example, in some instances, the updateable base matrix is a 3D rotation matrix obtained using based on visual
20 correspondences between a base reference frame captured by the reference camera at a base reference camera pose and a base match frame captured by the match camera at a base match camera pose.

[00160] In some implementations, a computer system obtains an updateable base matrix by identifying a set of base feature matches by performing feature matching between the
25 base reference frame and the base match frame. The computer system then generates a set of unprojected base feature matches by unprojecting the base feature matches into 3D space. The computer system then computes the updateable base matrix based on the set of unprojected base feature matches by minimizing a cost function (e.g., according to the Wahba method).

30 [00161] Act 2504 of flow diagram 2500 includes computing an updated matrix. Act 2504 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250) and using images captured by a reference camera 260 and a match camera 215. In some instances, a computer system computes an updated matrix using visual correspondences between an updated reference

frame captured by the reference camera at an updated reference camera pose and an updated match frame captured by the match camera at an updated match camera pose for generating an updated matrix.

5 [00162] In some implementations, a computer system computes an updated matrix by identifying a set of updated feature matches by performing feature matching between the updated reference frame and the updated match frame. The computer system also generates a set of unprojected updated feature matches by unprojecting the updated feature matches into 3D space, and the computer system computes the updated matrix based on the set of unprojected updated feature matches by minimizing a cost function. In some
10 implementations, the computer system computes the updated matrix using the Wahba method.

[00163] In response to successfully computing an updated matrix according to act 2504, act 2506 of flow diagram 2500 includes generating an aligned matrix, and act 2508 of flow diagram 2500 includes updating the updateable base matrix. Acts 2506 and 2508 are
15 performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system generates an aligned updated matrix (according to act 2506) using a base reference camera pose, a base match camera pose, an updated reference camera pose, and an updated match camera pose, which may be poses associated with the updateable base matrix
20 obtained according to act 2502. Furthermore, in some instances, a computer system updates the updateable base matrix (according to act 2508) by using the aligned updated matrix and the updateable base matrix (as previously obtained according to act 2502) as inputs for updating the updateable base matrix. The computer system may then proceed to perform act 2510, which includes generating a motion model using the updateable base matrix.

25 [00164] It should be noted that, in some instances, a computer system fails to compute an updated matrix. For example, in some instances, the match camera and the reference camera are not directed toward a common portion of a captured environment which may cause a system to fail to identify feature correspondences between a reference frame and a match frame. A computer system may fail to identify feature points for other reasons as
30 well, such as occlusions, differences in camera modalities, etc. Failure to identify feature points within the match frame and/or the reference frame may cause the system to fail to compute an updated matrix.

[00165] In response to failing to compute an updated matrix according to act 2504, in some implementations, a computer system refrains from updating the updateable base

matrix obtained according to act 2502. Instead, the computer system proceeds to act 2510 with the updateable base matrix obtained according to act 2502 (without updating the updateable base matrix, in contrast with act 2508 performed in response to successfully computing an updated matrix according to act 2504).

5 **[00166]** Act 2510 of flow diagram 2500 includes generating a motion model using the updateable base matrix (whether the updateable base matrix was updated according to act 2508 or not). Act 2510 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system the motion model includes the updateable base matrix, a
10 reference camera transformation matrix generated based on inertial tracking data associated with the reference camera, and a match camera transformation matrix generated based on inertial tracking data associated with the match camera.

[00167] Arrow 2512 of flow diagram 2500 indicates that, in some instances, an updateable base matrix obtained according to act 2502 is updateable base matrix that was
15 used to generate the motion model according to act 2510, whether the updateable base matrix was updated according to act 2508 or not. In this sense, an updateable base matrix may be regarded as a running average.

[00168] Figure 26 illustrates an example flow diagram 2600 depicting acts associated with facilitating continuous image alignment of two cameras. The discussion of the various
20 acts represented in flow diagram 2600 includes references to various hardware components described in more detail with reference to Figures 1, 2, and 34.

[00169] Act 2602 of flow diagram 2600 includes identifying a reference camera transformation matrix. Act 2602 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a
25 user instrument 250). In some implementations, the reference camera transformation matrix is a 3D rotational matrix (e.g., identified based on inertial tracking data 270 obtained by a reference camera IMU 265) between a base reference camera pose and an updated reference camera pose, the base reference camera pose being associated with a base reference camera timepoint that occurs prior to an updated reference camera timepoint.

30 **[00170]** Act 2604 of flow diagram 2600 includes identifying a match camera transformation matrix. Act 2604 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some implementations, the match camera transformation matrix is a 3D rotational matrix (e.g., identified based on inertial tracking data 240 obtained by a

match camera IMU 235) between a base match camera pose and an updated match camera pose, the base match camera pose being associated with a base match camera timepoint that occurs prior to an updated match camera timepoint.

[00171] Act 2606 of flow diagram 2600 includes identifying an alignment matrix. Act 2606 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, the alignment matrix is based on visual correspondences between one or more reference frames captured by the reference camera and one or more match frames captured by the match camera

[00172] Identifying an alignment matrix according to act 2606 may include various acts. For example, flow diagram 2600 illustrates that act 2606A includes identifying a base matrix (it should be noted that a base matrix may refer to a previously computed alignment matrix). In some instances, a computer system identifies a base matrix by identifying a set of base feature matches by performing feature matching between a base reference frame captured by the reference camera at the base reference camera pose and a base match frame captured by the match camera at the base match camera pose. The computer system also generates a set of unprojected base feature matches by unprojecting the base feature matches into 3D space. The computer system may then compute the base matrix based on the set of unprojected base feature matches by minimizing a cost function (e.g., according to the Wahba method). In some instances, the base matrix is a 3D rotation matrix.

[00173] Act 2606B includes identifying an updated matrix. In some instances, a computer system identifies an updated matrix by identifying a set of updated feature matches by performing feature matching between an updated reference frame captured by the reference camera at the updated reference camera pose and an updated match frame captured by the match camera at the updated match camera pose. The computer system also generates a set of unprojected updated feature matches by unprojecting the updated feature matches into 3D space. The computer system may then compute the updated matrix based on the set of unprojected updated feature matches by minimizing a cost function.

[00174] Act 2606C includes generating an aligned updated matrix. In some instances, a computer system generates the aligned updated matrix by aligning the updated matrix with the base matrix using the base reference camera pose, the base match camera pose, the updated reference camera pose, and the updated match camera pose.

[00175] Act 2606D includes fusing the aligned updated matrix with the base matrix. In some instances, a computer system fuses the aligned updated matrix with the base matrix

by applying the aligned updated matrix and the base matrix as inputs to a function for generating the alignment matrix. In some implementations, the function for generating the alignment matrix is an interpolation function, and the interpolation function may comprise a smoothness term that determines weights of the aligned updated matrix and the base matrix for generating the alignment matrix.

[00176] Act 2608 of flow diagram 2600 includes identifying a set of 3D points by unprojecting a set of pixels of the reference frame. Act 2608 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system identifies the 3D points by unprojecting the set of pixels of the reference frame using an intrinsic matrix of the reference camera and using a uniform depth.

[00177] Act 2610 of flow diagram 2600 includes generating a modified set of 3D points. Act 2610 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system generates a modified set of 3D points by applying the motion model to the set of 3D points identified according to act 2608.

[00178] Act 2612 of flow diagram 2600 includes projecting the modified set of 3D points. Act 2612 of flow diagram 2600 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system projects the modified set of 3D points (from act 2610) using an intrinsic matrix of the match camera. The projected modified set of 3D points may enable a computer system to generate a composite image for display to a user (e.g., on displays 225A and 225B of an HMD 200).

Updating Continuous Image Alignment of Separate Cameras

[00179] As described hereinabove, a system may generate a motion model that is configured to facilitate continuous mapping of sets of pixels from a reference frame to a corresponding sets of pixels of a match frame, even as the pose of the reference camera and the match camera change over time (see, e.g., Figures 11–17 and Equations 2 and 4). A motion model includes an alignment matrix, which may be thought of as a running average that uses a previous alignment matrix and an aligned current matrix to generate an updated alignment matrix for use in the motion model (as used herein, a “current matrix” may be thought of as analogous to an “updated matrix” as used herein). In some instances, as described hereinabove, a computer system fuses a previous alignment matrix (or base matrix) with an aligned current matrix to generate the updated alignment matrix.

[00180] To illustrate, Figure 27 depicts an example of a reference camera 260 of a user instrument 250 and a match camera 215 of an HMD 200 capturing an environment that includes a physical object 305 at different timepoints. Specifically, Figure 27 illustrates the reference camera 260 capturing an image of the physical object 305 at a previous reference camera pose 2705 and at a previous reference camera timepoint 2710. Similarly, Figure 27 illustrates the match camera 215 capturing an image of the physical object 305 at a previous match camera pose 2715 and at a previous match camera timepoint 2720.

[00181] Referring briefly to Figure 28, the images captured by the reference camera 260 and the match camera 215 at their respective previous timepoints form a previous frame pair 2820. In some instances, a system identifies visual correspondences 2815 between the images of the previous frame pair 2820 (e.g., by performing feature matching on the images of the previous frame pair 2820). A system may then determine a previous alignment matrix 2810 using the visual correspondences 2815 (and/or other components, such as one or more alignment matrices for timepoints prior to timepoints associated with the previous frame pair 2820, indicated by ellipsis 2825). In some implementations, the previous alignment matrix 2810 operates as part of a motion model to map pixels of a previous reference frame (e.g., captured at the previous reference camera timepoint 2710) to corresponding pixels of a previous match frame (e.g., captured at the previous match camera timepoint 2720) (e.g., see Equation 1 and/or 2).

[00182] Returning to example(s) depicted in Figure 27, after capturing images of the physical object 305 with the reference camera 260 at the previous reference camera timepoint 2710 and with the match camera 215 at the previous match camera timepoint 2720, the pose of the reference camera 260 changes (indicated by arrow 2725) to correspond to the intermediate reference camera pose 2735 at the intermediate reference camera timepoint 2740. Similarly, the pose of the match camera 215 changes (indicated by arrow 2730) to correspond to the intermediate match camera pose 2745 at the intermediate match camera timepoint 2750. Both the reference camera 260 and the match camera 215 capture images while oriented according to their respective intermediate pose (e.g., intermediate reference camera pose 2735 and intermediate match camera pose 2745, respectively) to form an intermediate frame pair.

[00183] As is evident from Figure 27, the reference camera 260, when oriented according to intermediate reference camera pose 2735, and the match camera 215, when oriented according to intermediate match camera pose 2745, are directed away from the physical object 305 such that neither camera captures the physical object 305. Accordingly, a

computer system may fail to identify visual correspondences between the images of the intermediate frame pair. Thus, Figure 28 illustrates that an intermediate alignment matrix 2830 for mapping pixels of an intermediate reference frame (e.g., captured at the intermediate reference camera timepoint 2740) to corresponding pixels of an intermediate match frame (e.g., captured at the intermediate match camera timepoint 2750) may incorporate the previous alignment matrix 2810, without generating an updated alignment matrix based on the intermediate frame pair (e.g., see Equation 2).

[00184] Returning again to example(s) depicted in Figure 27, after capturing the intermediate frame pair with the reference camera 260 at the intermediate reference camera timepoint 2740 and with the match camera 215 at the intermediate match camera timepoint 2750, the pose of the reference camera 260 changes (indicated by arrow 2755) to correspond to the current reference camera pose 2765 at the current reference camera timepoint 2770. Similarly, the pose of the match camera 215 changes (indicated by arrow 2760) to correspond to the current match camera pose 2775 at the current match camera timepoint 2780. Both the reference camera 260 and the match camera 215 capture images while oriented according to their respective current pose (e.g., current reference camera pose 2765 and current match camera pose 2775, respectively) to form an current frame pair 2850 (see, briefly, Figure 28). As is evident from Figure 27, the reference camera 260, when oriented according to current reference camera pose 2765, and the match camera 215, when oriented according to current match camera pose 2775, are directed toward the physical object 305 such that both cameras capture the physical object 305.

[00185] Referring again to Figure 28, in some instances, a system identifies visual correspondences 2845 between the images of the current frame pair 2850 (e.g., by performing feature matching on the images of the current frame pair 2850). A system may then determine a current matrix 2840 using the visual correspondences 2845. Figure 28 also illustrates that, in some instances, a system generates an aligned current matrix 2865 (e.g., according to Equation 3) and fuses the aligned current matrix 2865 with the previous alignment matrix 2810 (or the intermediate alignment matrix 2830, which incorporates the previous alignment matrix) to generate an updated alignment matrix 2860 (e.g., according to Equation 4). For example, with parenthetical reference to Figures 23A and 23B, a computer system may utilize a fuser (e.g., fuser 2300) that implements an interpolation function (e.g., interpolation function 2305), which, in essence, blends or combines the aligned current matrix 2865 (e.g., aligned updated matrix 2210) with the previous alignment matrix 2810 (e.g., base matrix 610 or alignment matrix 2315) to generate the updated

alignment matrix 2860 (e.g., alignment matrix 2315 or subsequent alignment matrix 2325).

[00186] In some implementations, the updated alignment matrix 2860 operates as part of a motion model to map pixels of a current reference frame (e.g., captured at the current reference camera timepoint 2770) to corresponding pixels of a current match frame (e.g., captured at the current match camera timepoint 2780) (e.g., see Equation 2).

[00187] As indicated hereinabove, in some implementations, an interpolation function utilizes a smoothness term (or smoothness function, see Equation 4). In some implementations, the smoothness term determines how to weight the previous alignment matrix 2810 and the aligned current matrix 2865 for generating the updated alignment matrix 2860. By way of example, in some implementations, when a smoothness term has a value that approaches a maximum value (e.g., a value of 1), the interpolation function increasingly ignores the aligned current matrix 2865, giving more weight to the previous alignment matrix 2810 for generating the updated alignment matrix 2860, which may be beneficial when expected accuracy of the previous alignment matrix 2810 is high. Furthermore, in some implementations, when the smoothness term has a value that approaches a minimum value (e.g., a value of 0), the interpolation function increasingly ignores the previous alignment matrix 2810, giving more weight to the aligned current matrix 2865 for generating the updated alignment matrix 2860, which may be beneficial when the expected accuracy of the previous alignment matrix 2810 is low.

[00188] Different smoothness values may be appropriate for different circumstances. For example, selecting a smoothness value that gives more weight to the previous alignment matrix 2810 may ameliorate the effects of noise that may be present in the aligned current matrix 2865 (e.g., noise introduced when performing feature matching). However, selecting a smoothness value that gives more weight to the aligned current matrix 2865 may ameliorate potential inaccuracy of the previous alignment matrix 2810 with respect to current frames (e.g., inaccuracy brought about by IMU drift).

[00189] Selecting a smoothness value that gives more weight to the previous alignment matrix 2810 when the previous alignment matrix 2810 is inaccurate may cause inaccuracies in composite frames. Although these inaccuracies may correct over time, the delay in providing accurate composite frames may render a motion model unsuitable for certain applications (e.g., precise operations and/or operations with rapid changes in camera orientation). Thus, in some implementations, it may be beneficial to intelligently determine/update the smoothness value based on the expected accuracy of the previous alignment matrix 2810.

[00190] The expected accuracy of a previous alignment matrix 2810 may depend on the circumstances. For example, in some instances, a reference camera 260 and/or a match camera 215 undergo(es) a significant amount of motion from the time that a previous frame pair 2820 was captured for generating a previous alignment matrix 2810, which may
5 degrade the accuracy/applicability of a previous alignment matrix 2810 with respect to frame pairs captured at current timepoints (e.g., because of IMU drift, changes in parallax, etc.). This may occur, for example, where a system fails to identify visual correspondences between images of one or more intermediate frame pairs that intervene between current timepoints and previous timepoints at which visual correspondences were successfully
10 identified (e.g., as illustrated in Figures 27 and 28).

[00191] Accordingly, in some instances, a computer system intelligently determines/updates the smoothness term based the expected accuracy of one or more previous alignment matrices.

[00192] Figure 29 illustrates the same frame capture and pose change sequence described
15 hereinabove with reference to Figures 27 and 28. Figure 29 demonstrates that in some instances, a system identifies a difference value 2910 associated with the reference camera 260 between the current reference camera timepoint 2770 and the previous reference camera timepoint 2710 (which may be the most recent previous reference camera timepoint for which visual correspondences were successfully identified to generate a previous alignment
20 matrix). Additionally, or alternatively, a system may identify a difference value 2920 associated with the match camera 215 between the current match camera timepoint 2780 and the previous match camera timepoint 2720 (which may be the most recent previous match camera timepoint for which visual correspondences were successfully identified to generate a previous alignment matrix). High difference value(s) 2910 and/or 2920 may
25 indicate that an expected accuracy of a previous alignment matrix 2810 should be regarded as low (e.g., because of IMU drift errors accumulating), whereas low difference value(s) 2910 and/or 2920 may indicate that an expected accuracy of a previous alignment matrix 2810 should be regarded as high. Additional details concerning difference values will be provided hereinafter.

[00193] Figure 30 illustrates an example of generating an updated alignment matrix 3015
30 using an aligned current matrix 2865, a previous alignment matrix 2810, and difference value(s) 2910 and/or 2920. For example, in some implementations, a system uses the aligned current matrix 2865, the previous alignment matrix 2810, and the difference value(s) 2910 and/or 2920 as inputs to a fuser 3000 for generating the updated alignment matrix 3015.

Similar to the fuser 2300 described hereinabove with reference to Figures 23A and 23B, the fuser 3000 may include an interpolation function 3005 that incorporates a smoothness term 3010 (and/or other alternative or additional terms, indicated by ellipsis 3030).

[00194] The smoothness term 3010, in some instances, is intelligently determined based on the difference value(s) 2910 and/or 2920. As described hereinabove, the smoothness term 3010 may determine a weight for blending the previous alignment matrix 2810 with the aligned current matrix 2865 to generate the updated alignment matrix 3015. For example, higher difference value(s) 2910 and/or 2920 may cause the smoothness term to approach a minimum value (e.g., a value of 0), which may cause the interpolation function 3005 to ignore the previous alignment matrix 2810 to a higher degree when blending the aligned current matrix 2865 with the previous alignment matrix 2810 to generate the updated alignment matrix 3015. Conversely, for example, lower difference value(s) 2910 and/or 2920 may cause the smoothness term to approach a maximum value (e.g., a value of 1), which may cause the interpolation function 3005 to ignore the aligned current matrix 2865 to a higher degree when blending the aligned current matrix 2865 with the previous alignment matrix 2810 to generate the updated alignment matrix 3015.

[00195] Figure 31 illustrates example implementations of the difference value(s) 2910 and/or 2920 that a system may identify in association with a reference camera 260 and/or a match camera 215 at different timepoints. Figure 31 illustrates that, in some instances, the difference value(s) 2910 and/or 2920 is/are identified as a motion value 3105. For instance, a motion value 3105 may indicate an amount of motion 3110 that the reference camera 260 and/or the match camera 215 has/have undergone between a previous timepoint (e.g., previous reference camera timepoint 2710, previous match camera timepoint 2720) and a current timepoint (current reference camera timepoint 2770, current match camera timepoint 2780). A large amount of motion may indicate that IMU drift errors have been able to accumulate over time, indicating that the expected accuracy of a previous alignment matrix 2810 is low, which may cause a system to select a low smoothness term 3010.

[00196] Figure 31 also demonstrates that a motion value 3105 may indicate one or more acceleration values 3115 experienced by the reference camera 260 and/or the match camera 215 between a previous timepoint (e.g., previous reference camera timepoint 2710, previous match camera timepoint 2720) and a current timepoint (current reference camera timepoint 2770, current match camera timepoint 2780). High acceleration values may cause IMU drift errors to be higher than would exist for low acceleration values (e.g., particularly where components of the IMU become saturated, such as a gyroscope), which may indicate that the

expected accuracy of a previous alignment matrix 2810 is low and may cause a system to select a low smoothness term 3010. The ellipsis 3125 indicates that a motion value 3105 may indicate other metrics, such as, by way of non-limiting example, translational velocity (which may indicate a change in parallax that may render a previous alignment matrix 2810 less accurate).

[00197] As depicted in Figure 31, difference value(s) 2910 and/or 2920 are identified as a temporal value 3120, such as a number of frames or amount of time that have/has elapsed since the most recent previous timepoint at which visual correspondences were successfully identified. A high number of frames or amount of time between successful visual correspondences may indicate that a previous alignment matrix 2810 may have become inaccurate, which may cause a system to select a lower smoothness term 3010. The ellipsis 3130 indicates that other implementations of difference value(s) 2910 and/or 2920 are within the scope of this disclosure.

[00198] Those skilled in the art will recognize, in view of the present disclosure, that the difference value(s) 2910 and/or 2920 may take on various forms. Accordingly, it will be appreciated, in view of the present disclosure, that the descriptions herein of difference values as “high” or “low” may refer to the magnitude of the difference value in absolute terms (e.g., regardless of directionality and/or whether a particular measured value has a positive or negative value).

[00199] The following discussion focuses on a particular example implementation of a smoothness term/smoothness function as described hereinabove. One will appreciate, in view of the present disclosure, that the following particular example implementation is provided to assist in understanding, and not by way of limitation.

[00200] In one example implementation, the function *smoothness()* is defined as follows:

$$smoothness() = strength * e^{\frac{-0.5 * angle_sum()^2}{falloff^2}} \quad (5)$$

[00201] Where the parameters *strength* and the parameter *falloff* may be set to predetermined constants (e.g., **0.7** for *strength*, **30** for *falloff*). The function *angle_sum* may be thought of as computing difference values for the reference camera and the match camera as described hereinabove and may be defined as follows:

$$angle_sum() = angle(P_{ref_cur}, P_{ref_last}) + angle(P_{match_cur}, P_{match_last}) \quad (6)$$

[00202] Where P_{ref_cur} represents the current pose of the reference camera and P_{ref_last} represents the reference camera pose at which visual correspondences were most recently

successfully identified/computed (e.g., the reference camera pose for the most recent timepoint at which an alignment matrix was successfully updated). Similarly, P_{match_cur} represents the current pose of the match camera and P_{match_last} represents the match camera pose at which visual correspondences were most recently successfully identified/computed.

5 [00203] The function *angle*() may be generalized as follows:

$$angle(P, P') = \text{acos}((P * (0,0,1)^T)^T * (P' * (0,0,1)^T)) \quad (7)$$

[00204] Which computes the cross product of the third columns of the two matrices P and P' , which may correspond to the z axes of the two matrices. The *acos*() operation may provide the angle between the computed axes.

10 [00205] Figure 32 illustrates an example plot of an example implementation of a smoothness function described hereinabove with reference to Equations 5–7. As is evident from Figure 32, the value of the smoothness term is greater for smaller difference values (e.g., for smaller anglesum values, in absolute terms), and the value of the smoothness term is smaller for greater difference values (e.g., for larger anglesum values, in absolute terms).

Example Method(s) for Updating Continuous Image Alignment of Separate Cameras

15 [00206] The following discussion now refers to a number of methods and method acts that may be performed. Although the method acts may be discussed in a certain order or illustrated in a flow chart as occurring in a particular order, no particular ordering is required unless specifically stated, or required because an act is dependent on another act being completed prior to the act being performed.

20 [00207] Figure 33 illustrates an example flow diagram 3300 depicting acts associated with updating continuous image alignment of a reference camera and a match camera. The discussion of the various acts represented in flow diagram 3300 includes references to various hardware components described in more detail with reference to Figures 1, 2, and 34.

25 [00208] Act 3302 of flow diagram 3300 includes identifying a previous alignment matrix. Act 3302 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, the previous alignment matrix is associated with a previous frame pair captured at one or more previous timepoints by a reference camera and a match camera. In some instances, the
30 previous alignment matrix is based on visual correspondences between images of the previous frame pair.

[00209] Act 3304 of flow diagram 3300 includes identifying a current matrix. Act 3304

is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, the current matrix is associated with a current frame pair captured at one or more current timepoints by the reference camera and the match camera. In some instances, the current matrix is based on
5 visual correspondences between images of the current frame pair. Furthermore, in some implementations, the current matrix is an aligned current matrix, in that the current matrix is aligned with the previous alignment matrix using inertial tracking data associated with the reference camera and the match camera.

[00210] Act 3306 of flow diagram 3300 includes identifying a difference value. Act 3306
10 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). The difference value may be associated with the reference camera or the match camera relative to the one or more previous timepoints and the one or more current timepoints. In some implementations, the difference value comprises a motion value associated with the reference camera or the match
15 camera relative to the one or more previous timepoints and the one or more current timepoints. The motion value may indicate an amount of motion associated with the reference camera or the match camera relative to the one or more previous timepoints and the one or more current timepoints. Additionally, or alternatively, the motion value may indicate an acceleration associated with the reference camera or the match camera relative
20 to the one or more previous timepoints and the one or more current timepoints. In some implementations, the difference value comprises a temporal value associated with the reference camera or the match camera relative to the one or more previous timepoints and the one or more current timepoints.

[00211] Act 3308 of flow diagram 3300 includes generating an updated alignment matrix
25 using the previous alignment matrix, the current matrix, and the difference value. Act 3308 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system utilizes the previous alignment matrix, the current matrix, and the difference value as inputs to a fuser that includes an interpolation function for generating the updated
30 alignment matrix. In some instances, the interpolation function blends the previous alignment matrix with the current matrix to generate the updated alignment matrix. Furthermore, in some implementations, the interpolation function comprises a smoothness term that is determined based on the difference value. The smoothness term may determine a weight for blending the previous alignment matrix with the current matrix to generate the

updated alignment matrix. For example, when the smoothness term has a minimum smoothness value, the weight may cause the interpolation function to ignore the previous alignment matrix when generating the updated alignment matrix.

5 [00212] Act 3310 of flow diagram 3300 includes using the updated alignment matrix as a previous alignment matrix for generating a subsequent updated alignment matrix. Act 3310 is performed, in some instances, using one or more processors 3405 of a computer system 3400 (e.g., an HMD 200 and/or a user instrument 250). In some instances, a computer system utilizes the updated alignment matrix, a subsequently obtained current matrix, and a subsequently identified difference value as inputs to a fuser that includes an
10 interpolation function for generating the subsequent updated alignment matrix (e.g., similar to the manner described hereinabove with reference to act 3308).

Example Computer System(s)

[00213] Having just described the various features and functionalities of some of the disclosed embodiments, the focus will now be directed to Figure 34 which illustrates an
15 example computer system 3400 that may include and/or be used to facilitate the embodiments described herein, including the acts described in reference to the foregoing Figures. In particular, this computer system 3400 may be implemented as part of a mixed-reality HMD, such as any HMD referenced herein.

[00214] Computer system 3400 may take various different forms. For example, computer
20 system 3400 may be embodied as a tablet, a desktop, a laptop, a mobile device, a cloud device, an HMD, or a standalone device, such as those described throughout this disclosure. Computer system 3400 may also be a distributed system that includes one or more connected computing components/devices that are in communication with computer system 3400. Figure 34 specifically calls out how computer system 3400 may be embodied as a tablet
25 3400A, a laptop 3400B, or an HMD 3400C, but the ellipsis 3400D indicates that computer system 3400 may be embodied in other forms as well.

[00215] The computer system 3400 includes various different components. Figure 34 shows that computer system 3400 includes one or more processors 3405 (aka a “hardware processing unit”), a machine learning (ML) engine 3410, graphics rendering engine(s) 3425,
30 a display system 3430, input/output (I/O) interfaces 3435, one or more sensors 3440, and storage 3445.

[00216] Regarding the processor(s) 3405, it will be appreciated that the functionality described herein can be performed, at least in part, by one or more hardware logic components (e.g., the processor(s) 3405). For example, and without limitation, illustrative

types of hardware logic components/processors that can be used include Field-Programmable Gate Arrays (“FPGA”), Program-Specific or Application-Specific Integrated Circuits (“ASIC”), Application-Specific Standard Products (“ASSP”), System-On-A-Chip Systems (“SOC”), Complex Programmable Logic Devices (“CPLD”), Central Processing Units (“CPU”), Graphical Processing Units (“GPU”), or any other type of programmable hardware.

[00217] As used herein, the terms “executable module,” “executable component,” “component,” “module,” or “engine” can refer to hardware processing units or to software objects, routines, or methods that may be executed on computer system 3400. The different components, modules, engines, and services described herein may be implemented as objects or processors that execute on computer system 3400 (e.g. as separate threads).

[00218] The ML engine 3410 may be implemented as a specific processing unit (e.g., a dedicated processing unit as described earlier) configured to perform one or more specialized operations for the computer system 3400. The ML engine 3410 (or perhaps even just the processor(s) 3405) can be configured to perform any of the disclosed method acts or other functionalities.

[00219] In some instances, the graphics rendering engine 3425 is configured, with the hardware processing unit 3405, to render one or more virtual objects within the scene. As a result, the virtual objects accurately move in response to a movement of the user and/or in response to user input as the user interacts within the virtual scene. The computer system 3400 may include a display system 3430 (e.g., laser diodes, light emitting diodes (LEDs), microelectromechanical systems (MEMS), mirrors, lens systems, diffractive optical elements (DOEs), display screens, and/or combinations thereof) for presenting virtual objects within the scene.

[00220] I/O interface(s) 3435 includes any type of input or output device. Such devices include, but are not limited to, touch screens, displays, a mouse, a keyboard, a controller, and so forth. Any type of input or output device should be included among I/O interface(s) 3435, without limitation.

[00221] During use, in some instances, a user of the computer system 3400 is able to perceive information (e.g., a mixed-reality environment) through a display screen that is included among the I/O interface(s) 3435 and that is visible to the user. The I/O interface(s) 3435 and sensors 3440/3465 may also include gesture detection devices, eye tracking systems, and/or other movement detecting components (e.g., head tracking cameras, depth detection systems, gyroscopes, accelerometers, magnetometers, acoustic sensors, global

positioning systems (“GPS”), etc.) that are able to detect positioning and movement of one or more real-world objects, such as a user’s hand, a stylus, and/or any other object(s) that the user may interact with while being immersed in the scene.

5 [00222] The computer system 3400 may also be connected (via a wired or wireless connection) to external sensors 3465 (e.g., one or more remote cameras, accelerometers, gyroscopes, acoustic sensors, magnetometers, etc.). It will be appreciated that the external sensors include sensor systems (e.g., a sensor system including a light emitter and camera), rather than solely individual sensor apparatuses.

10 [00223] Storage 3445 may be physical system memory, which may be volatile, non-volatile, or some combination of the two. The term “memory” may also be used herein to refer to non-volatile mass storage such as physical storage media. If computer system 3400 is distributed, the processing, memory, and/or storage capability may be distributed as well.

15 [00224] Storage 3445 is shown as including executable instructions (i.e. code 3450). The executable instructions (i.e. code 3450) represent instructions that are executable by the processor(s) 3405 of computer system 3400 to perform the disclosed operations, such as those described in the various methods. Storage 3445 is also shown as including data 3455. Data 3455 may include any type of data, including image data, depth/disparity maps and/or other depth data, pose data, tracking data, and so forth, without limitation.

20 [00225] The disclosed embodiments may comprise or utilize a special-purpose or general-purpose computer including computer hardware, such as, for example, one or more processors (such as processor(s) 3905) and system memory (such as storage 3945), as discussed in greater detail below. Embodiments also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed
25 by a general-purpose or special-purpose computer system. Computer-readable media that store computer-executable instructions in the form of data are one or more “physical/hardware computer storage media” or “physical/hardware storage device(s)” that are distinguished from and that exclude mere transmission or transitory media. In contrast, computer-readable media that merely carry computer-executable instructions are
30 “transmission media.” Thus, by way of example and not limitation, the current embodiments can comprise at least two distinctly different kinds of computer-readable media: computer storage media and transmission media.

[00226] Computer storage media (aka “hardware storage device”) are computer-readable hardware storage devices, such as RAM, ROM, EEPROM, CD-ROM, solid state drives

(“SSD”) that are based on RAM, Flash memory, phase-change memory (“PCM”), or other types of memory, or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store desired program code means in the form of computer-executable instructions, data, or data structures and that can be
5 accessed by a general-purpose or special-purpose computer.

[00227] Computer system 3400 may also be connected (via a wired or wireless connection) to external sensors (e.g., one or more remote cameras) or devices via a network 3460. For example, computer system 3400 can communicate with any number devices or cloud services to obtain or process data. In some cases, network 3460 may itself be a cloud
10 network. Furthermore, computer system 3400 may also be connected through one or more wired or wireless networks 3460 to remote/separate computer systems(s) 3470 that are configured to perform any of the processing described with regard to computer system 3400.

[00228] A “network,” like network 3460, is defined as one or more data links and/or data switches that enable the transport of electronic data between computer systems, modules,
15 and/or other electronic devices. When information is transferred, or provided, over a network (either hardwired, wireless, or a combination of hardwired and wireless) to a computer, the computer properly views the connection as a transmission medium. Computer system 3400 will include one or more communication channels that are used to communicate with the network 3460. Transmission media include a network that can be
20 used to carry data or desired program code means in the form of computer-executable instructions or in the form of data structures. Further, these computer-executable instructions can be accessed by a general-purpose or special-purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

[00229] Upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to computer storage media (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a network
30 interface card or “NIC”) and then eventually transferred to computer system RAM and/or to less volatile computer storage media at a computer system. Thus, it should be understood that computer storage media can be included in computer system components that also (or even primarily) utilize transmission media.

[00230] Computer-executable (or computer-interpretable) instructions comprise, for

example, instructions that cause a general-purpose computer, special-purpose computer, or special-purpose processing device to perform a certain function or group of functions. The computer-executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter
5 has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[00231] Those skilled in the art will appreciate that the embodiments may be practiced
10 in network computing environments with many types of computer system configurations, including personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, routers, switches, and the like. The embodiments may also be
15 practiced in distributed system environments where local and remote computer systems that are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network each perform tasks (e.g. cloud computing, cloud services and the like). In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[00232] One will also appreciate how any feature or operation disclosed herein may be
20 combined with any one or combination of the other features and operations disclosed herein. Additionally, the content or feature in any one of the figures may be combined or used in connection with any content or feature used in any of the other figures. In this regard, the content disclosed in any one figure is not mutually exclusive and instead may be combinable
25 with the content from any of the other figures.

[00233] The present invention may be embodied in other specific forms without departing from its spirit or characteristics. The described embodiments are to be considered
in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes
30 which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

1. A system for facilitating continuous image alignment of two cameras, the system comprising:

one or more processors; and

one or more hardware storage devices storing computer-executable instructions that are executable by the one or more processors to configure the system to facilitate continuous image alignment of two cameras and by at least configuring the system to perform the following:

identify a reference camera transformation matrix between a base reference camera pose and an updated reference camera pose, the base reference camera pose being associated with a base reference camera timepoint that occurs prior to an updated reference camera timepoint;

identify a match camera transformation matrix between a base match camera pose and an updated match camera pose, the base match camera pose being associated with a base match camera timepoint that occurs prior to an updated match camera timepoint;

identify an alignment matrix based on visual correspondences between one or more reference frames captured by the reference camera and one or more match frames captured by the match camera; and

generate a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by the reference camera to a corresponding set of pixels of a match frame captured by the match camera, the motion model being generated based on at least the reference camera transformation matrix, the match camera transformation matrix, and the alignment matrix.

2. The system of claim 1, wherein at least one of the reference camera transformation matrix or the match camera transformation matrix or the alignment matrix is a 3D rotational matrix.

3. The system of claim 2, wherein at least one of the reference camera transformation matrix or the match camera transformation matrix is identified based on inertial tracking data.

4. The system of claim 1, wherein the motion model is further based on both an intrinsic matrix of the reference camera and an intrinsic matrix of the match camera.

5. The system of claim 1, wherein the computer-executable instructions are

further executable to configure the system to identify the alignment matrix using a base matrix, the base matrix being generated by:

identifying a set of base feature matches by performing feature matching between a base reference frame captured by the reference camera at the base reference camera pose and a base match frame captured by the match camera at the base match camera pose;

generating a set of unprojected base feature matches by unprojecting the base feature matches into 3D space; and

computing the base matrix based on the set of unprojected base feature matches by minimizing a cost function.

6. The system of claim 5, wherein the base matrix is a 3D rotation matrix.

7. The system of claim 5, wherein the base matrix is computed using the Wahba method.

8. The system of claim 5, wherein the computer-executable instructions are operable to configure the system to identify the alignment matrix using the base matrix and an updated matrix, the updated matrix being generated by:

identifying a set of updated feature matches by performing feature matching between an updated reference frame captured by the reference camera at the updated reference camera pose and an updated match frame captured by the match camera at the updated match camera pose;

generating a set of unprojected updated feature matches by unprojecting the updated feature matches into 3D space; and

computing the updated matrix based on the set of unprojected updated feature matches by minimizing a cost function.

9. The system of claim 8, wherein the computer-executable instructions are further executable to configure the system to identify the alignment matrix by:

generating an aligned updated matrix by aligning the updated matrix with the base matrix using the base reference camera pose, the base match camera pose, the updated reference camera pose, and the updated match camera pose; and

applying the aligned updated matrix and the base matrix as inputs to a function for generating the alignment matrix.

10. The system of claim 9, wherein the function for generating the alignment matrix is an interpolation function.

11. The system of claim 10, wherein the interpolation function comprises a

smoothness term that determines weights of the aligned updated matrix and the base matrix for generating the alignment matrix.

12. The system of claim 8, wherein the computer-executable instructions are further executable to configure the system to use the identified alignment matrix with a subsequent updated matrix for identifying a subsequent alignment matrix.

13. The system of claim 1, wherein the computer-executable instructions are further executable to configure the system to:

map the set of pixels of the reference frame to the corresponding set of pixels of the match frame by:

identifying a set of 3D points by unprojecting the set of pixels of the reference frame using an intrinsic matrix of the reference camera;

generating a modified set of 3D points by applying the motion model to the set of 3D points; and

projecting the modified set of 3D points using an intrinsic matrix of the match camera.

14. A system for generating a motion model configured to facilitate mapping of a set of pixels of a reference frame captured by a reference camera to a corresponding set of pixels of a match frame captured by a match camera, the system comprising:

one or more processors; and

one or more hardware storage devices storing computer-executable instructions that are executable by the one or more processors to configure the system to facilitate mapping of a set of pixels of a reference frame captured by a reference camera to a corresponding set of pixels of a match frame captured by a match camera, and by at least configuring the system to perform the following:

obtain an updateable base matrix based on visual correspondences between a base reference frame captured by the reference camera at a base reference camera pose and a base match frame captured by the match camera at a base match camera pose;

identify and apply visual correspondences between an updated reference frame captured by the reference camera at an updated reference camera pose and an updated match frame captured by the match camera at an updated match camera pose for generating an updated matrix;

in response to generating the updated matrix, modify the updateable base matrix with the updated matrix by at least:

generating an aligned updated matrix by aligning the updated matrix with the updateable base matrix using the base reference camera pose, the base match camera pose, the updated reference camera pose, and the updated match camera pose; and

applying the aligned updated matrix and the updateable base matrix as inputs for updating the updateable base matrix; and
generate a motion model using:

the updateable base matrix,

a reference camera transformation matrix generated based on inertial tracking data associated with the reference camera, and

a match camera transformation matrix generated based on inertial tracking data associated with the match camera.

15. The system of claim 14, wherein the computer-executable instructions are further executable to configure the system to refrain from updating the updateable base matrix in response to failing to successfully generate the updated matrix.

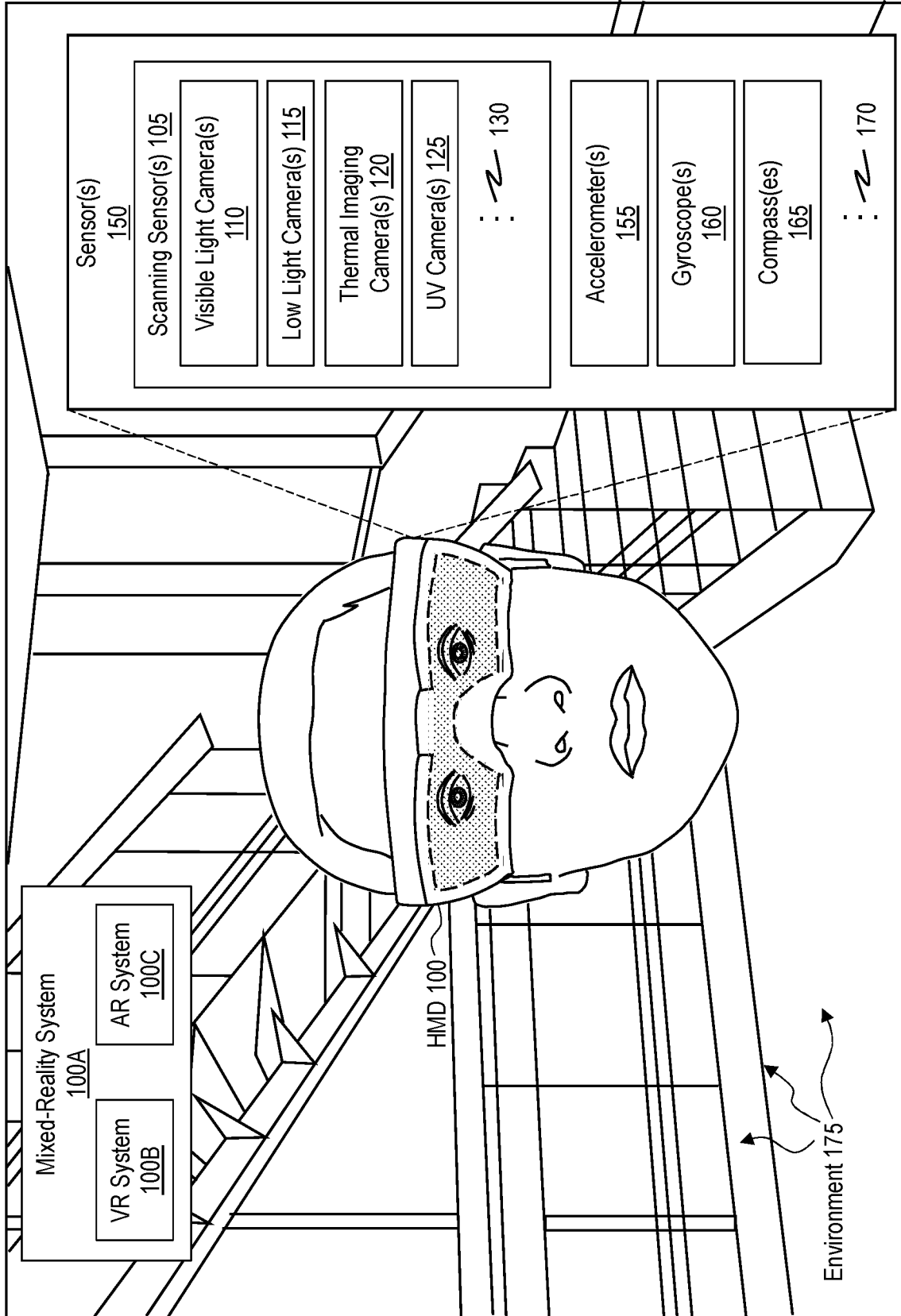


FIG. 1

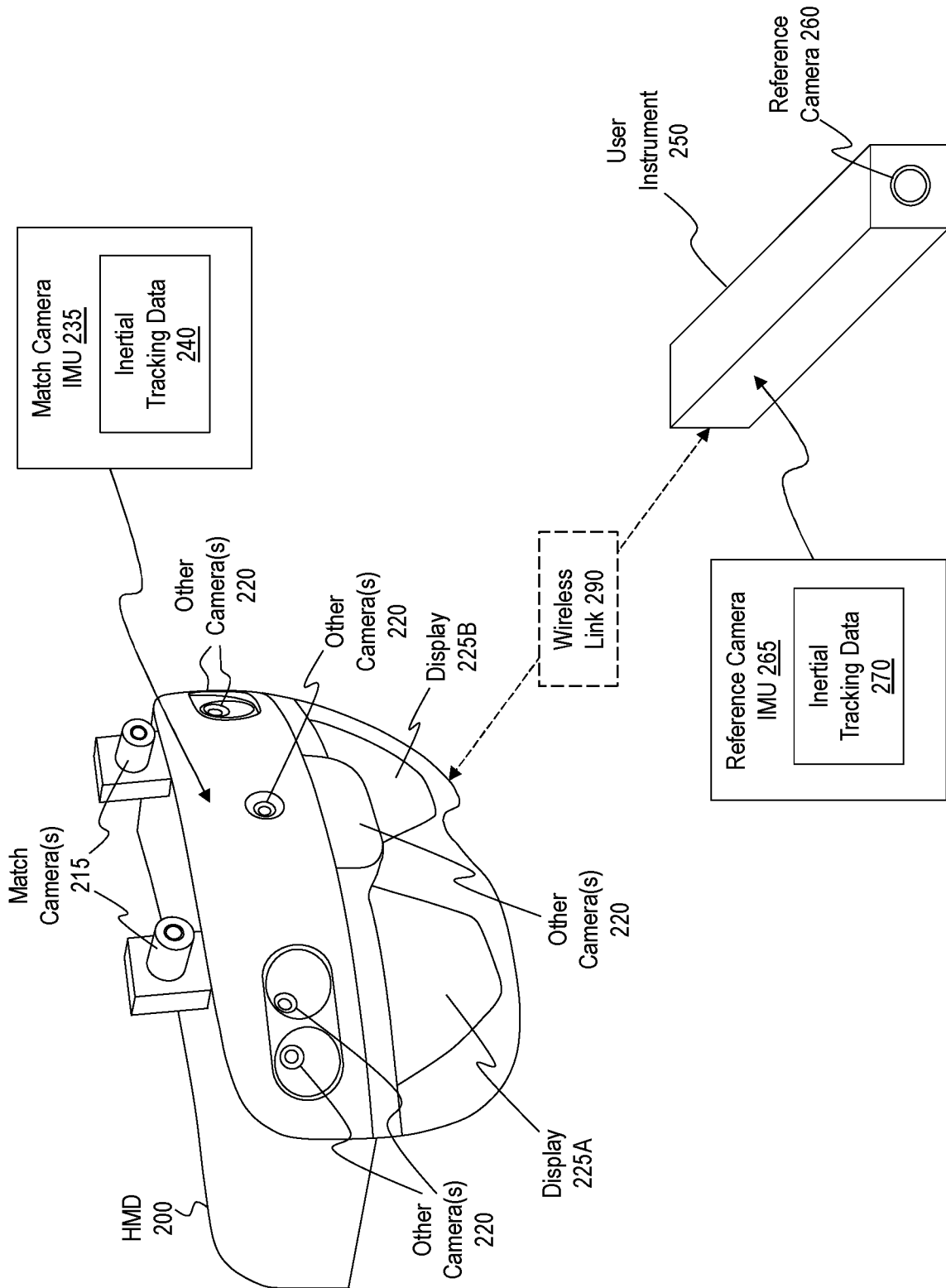


FIG. 2

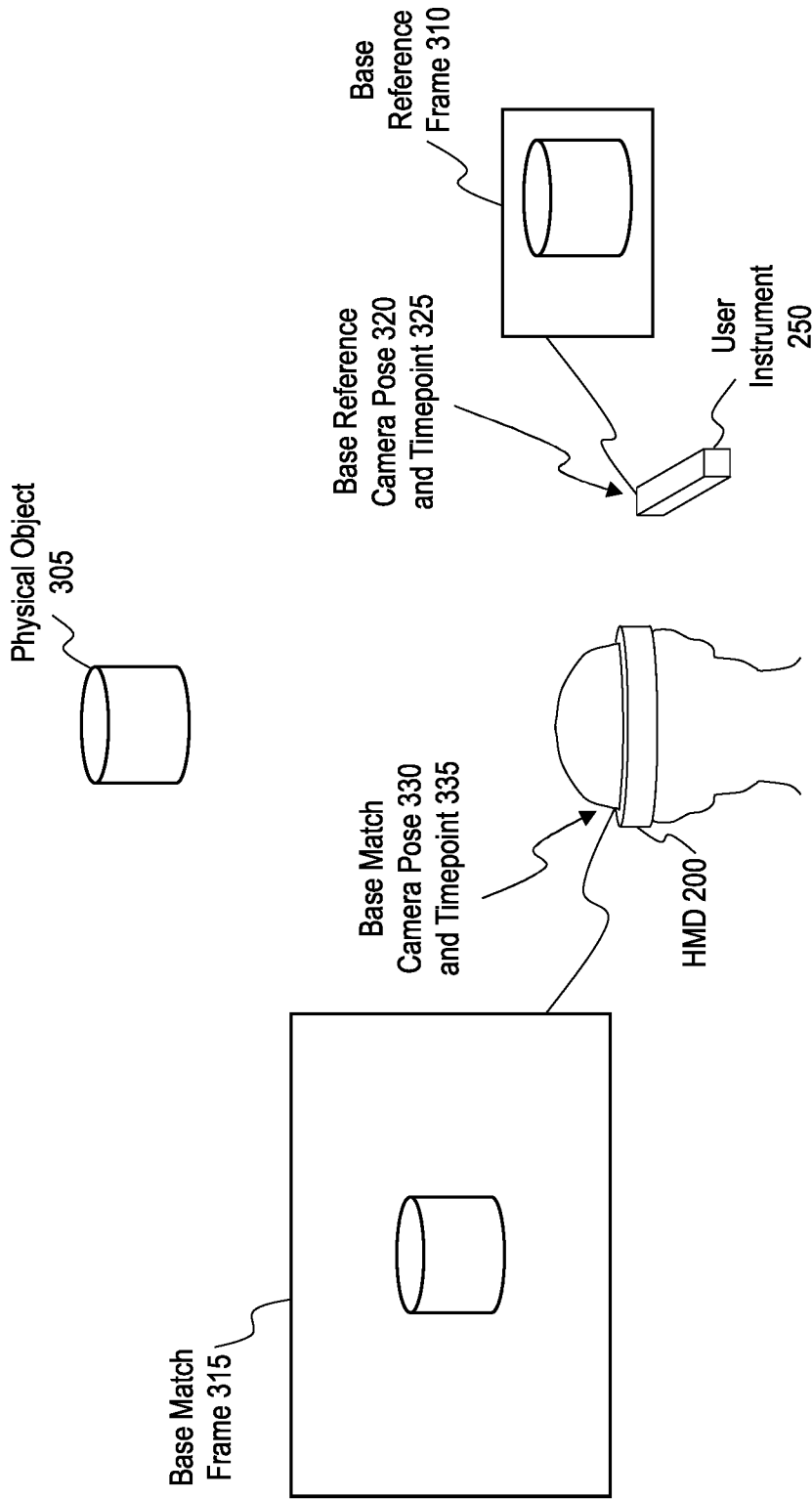


FIG. 3

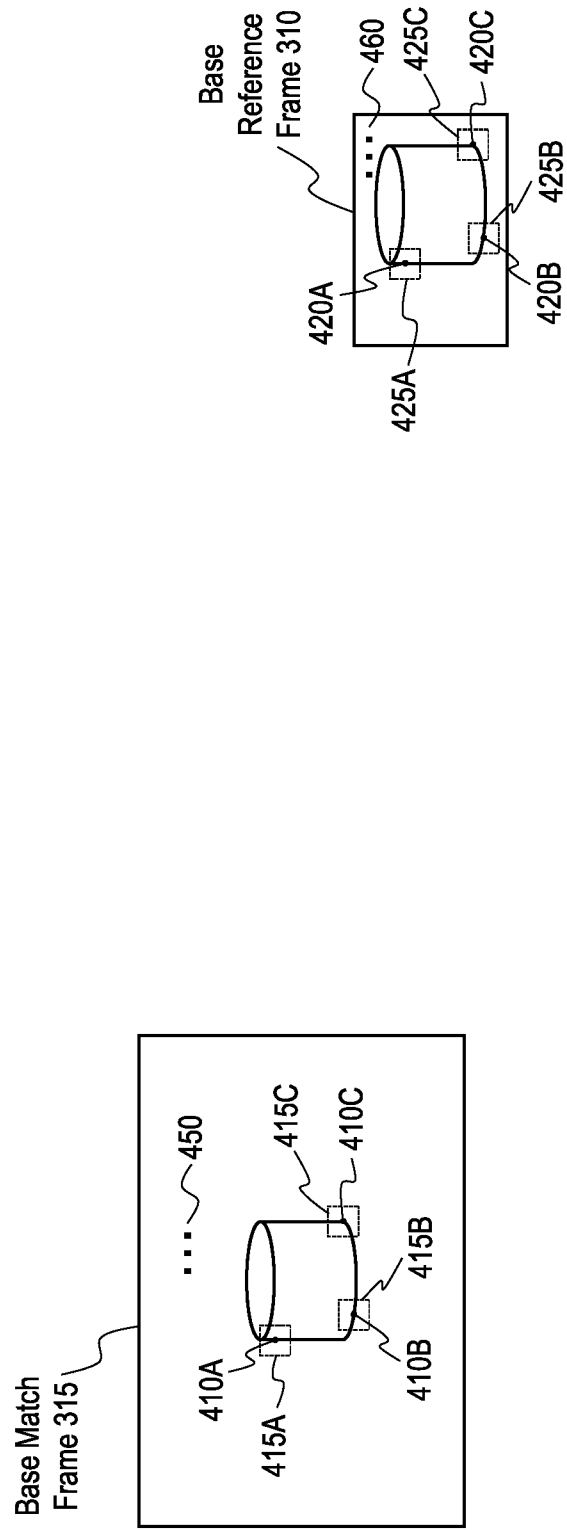


FIG. 4

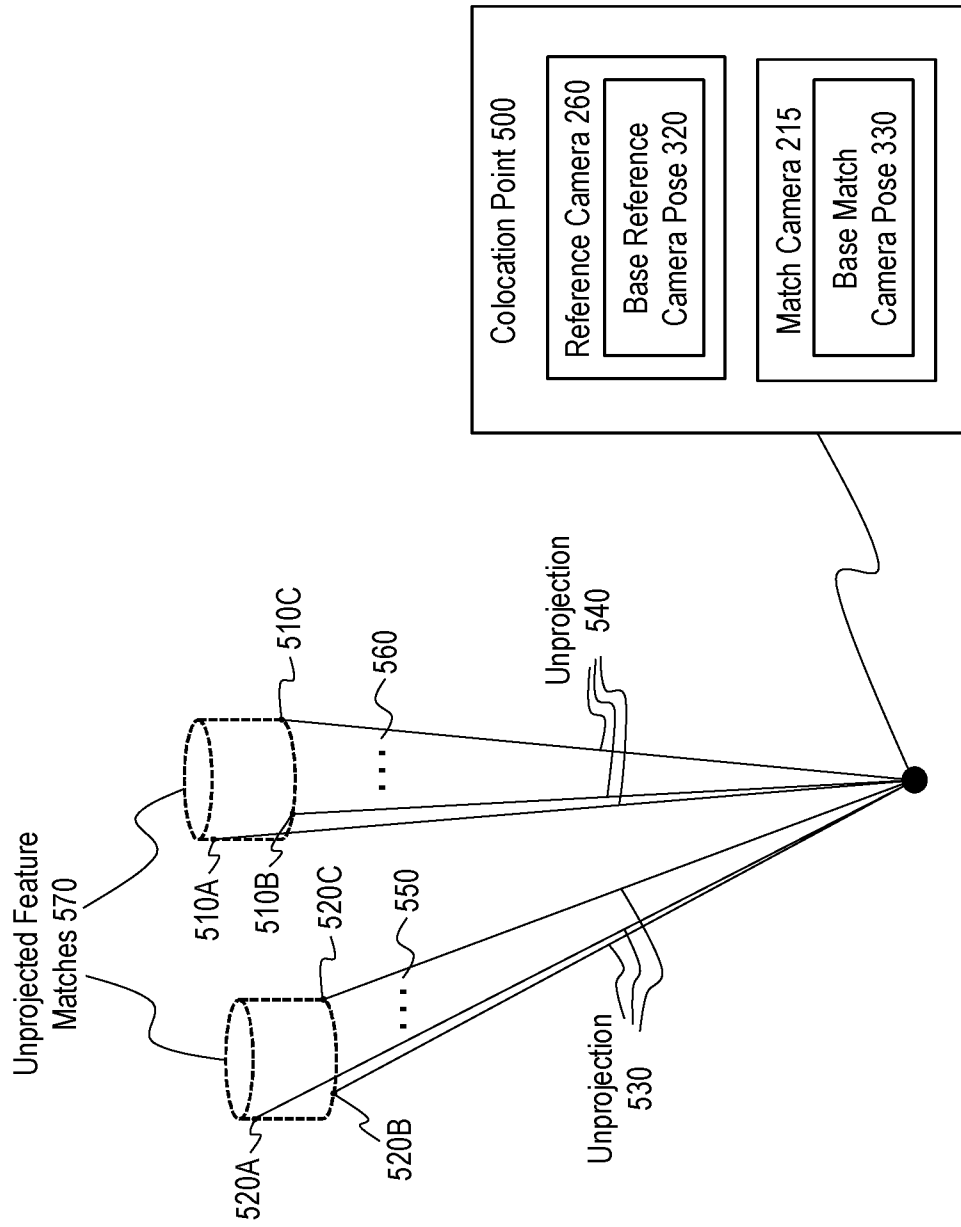


FIG. 5

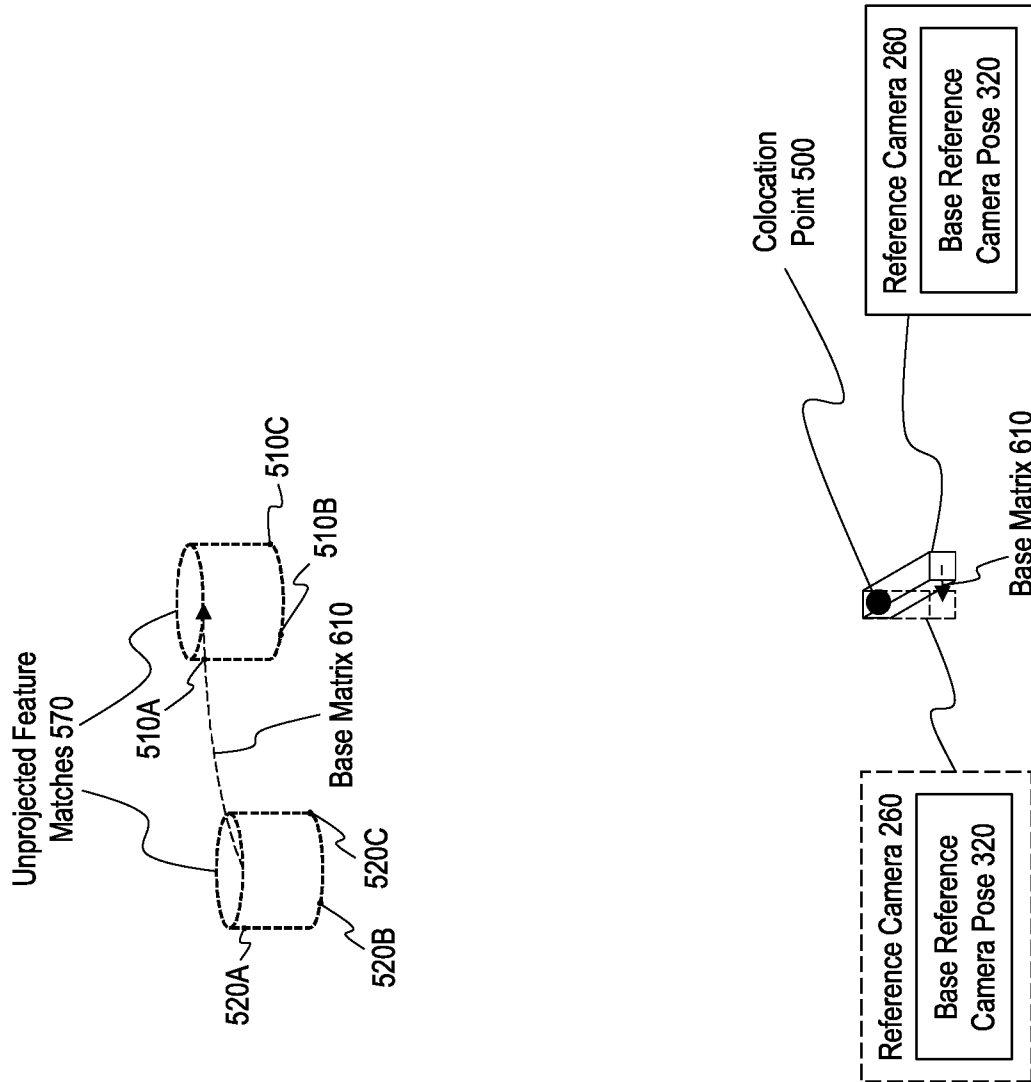


FIG. 6

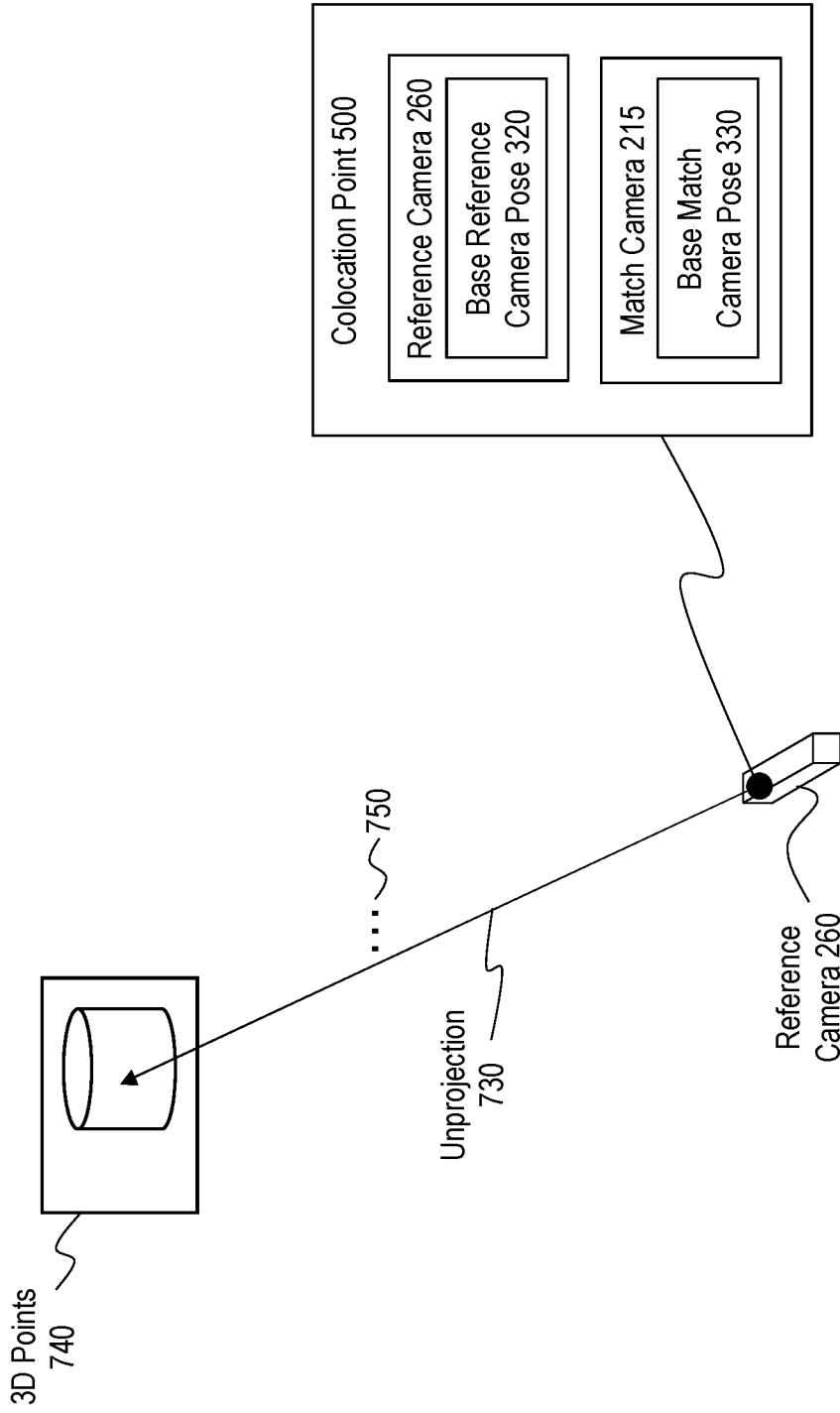


FIG. 7

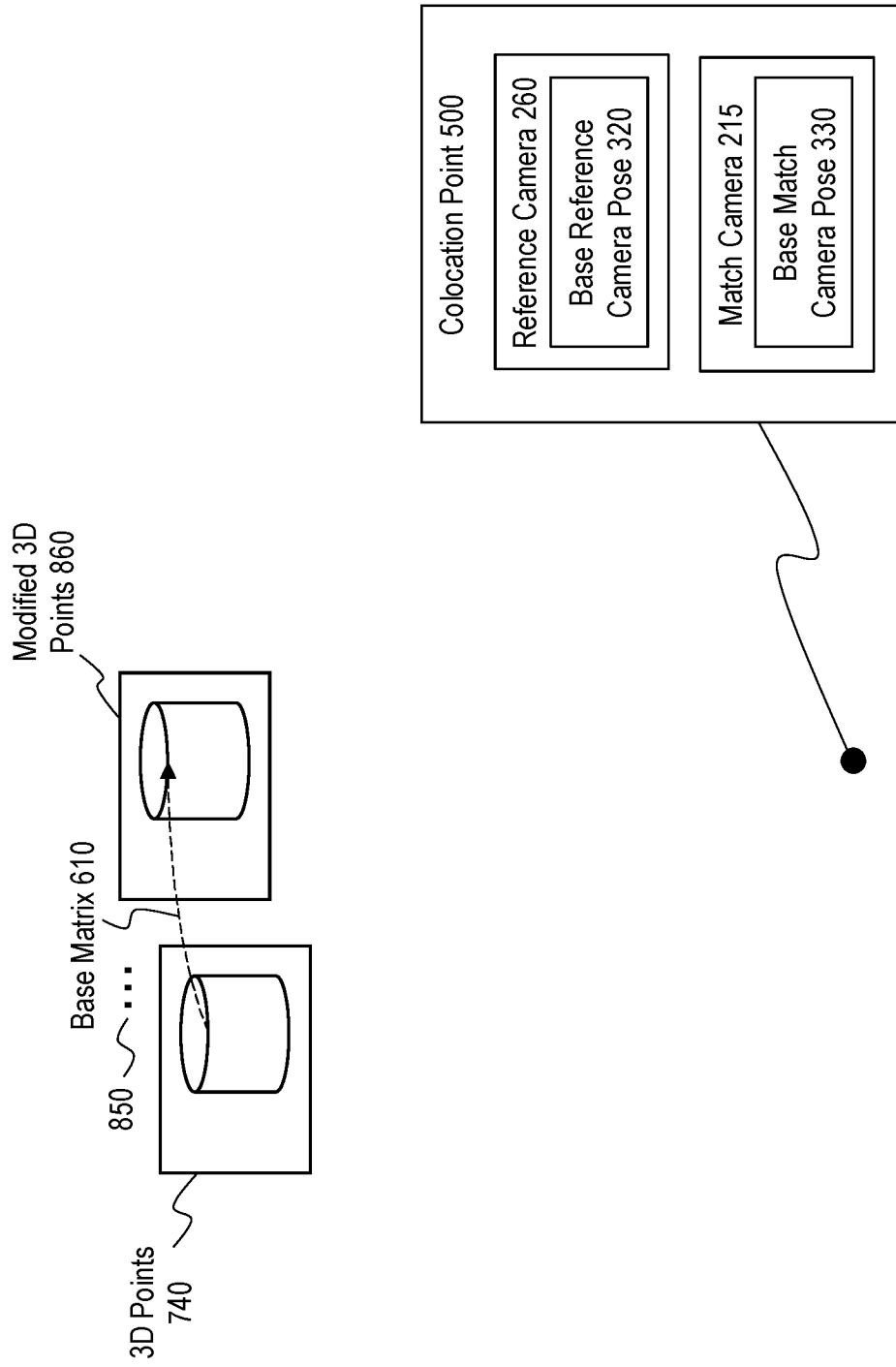


FIG. 8

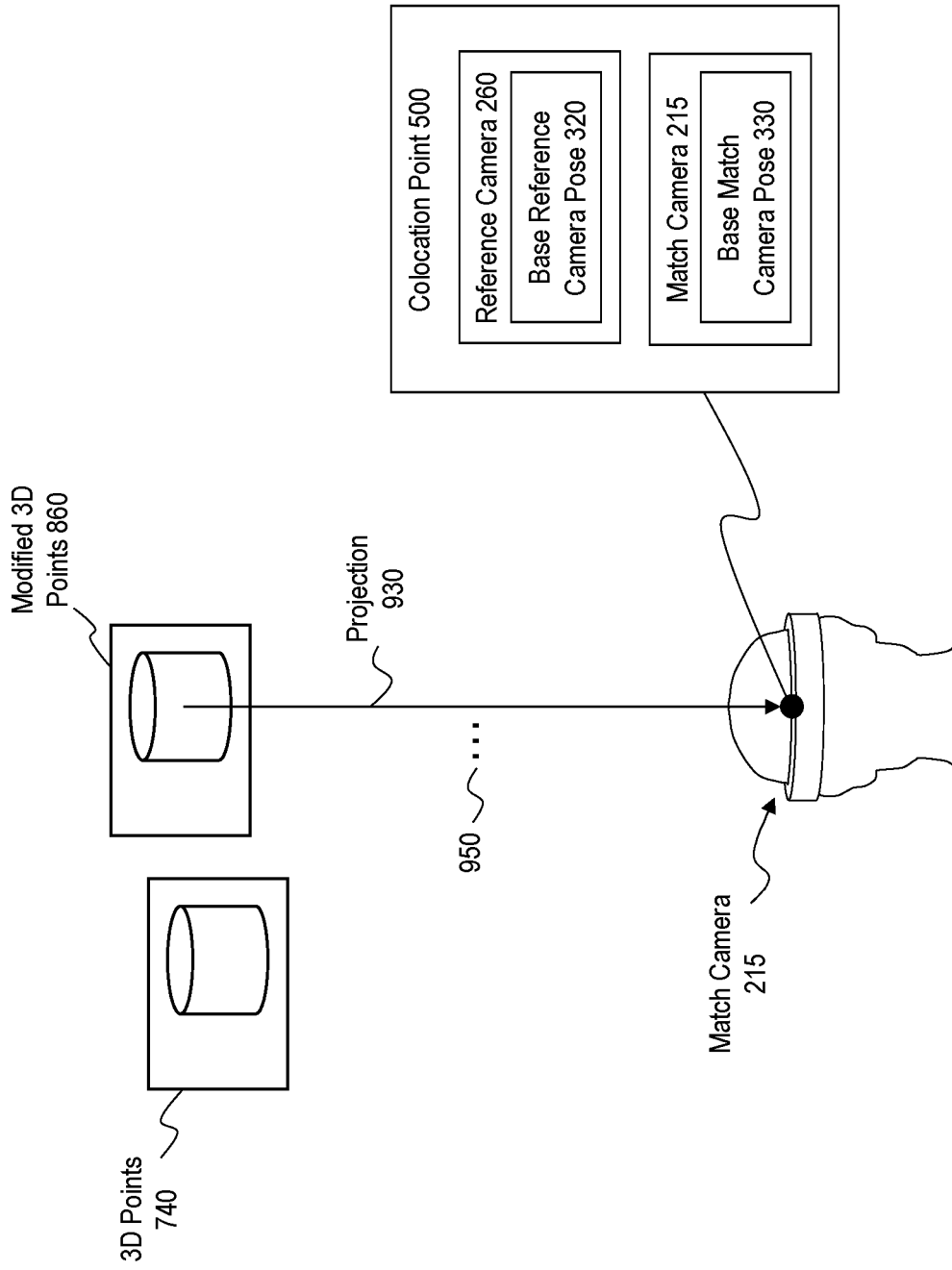


FIG. 9

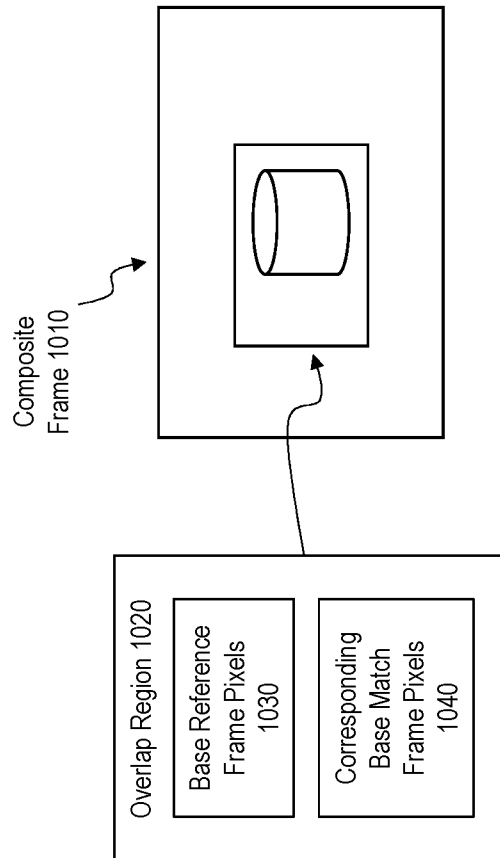


FIG. 10

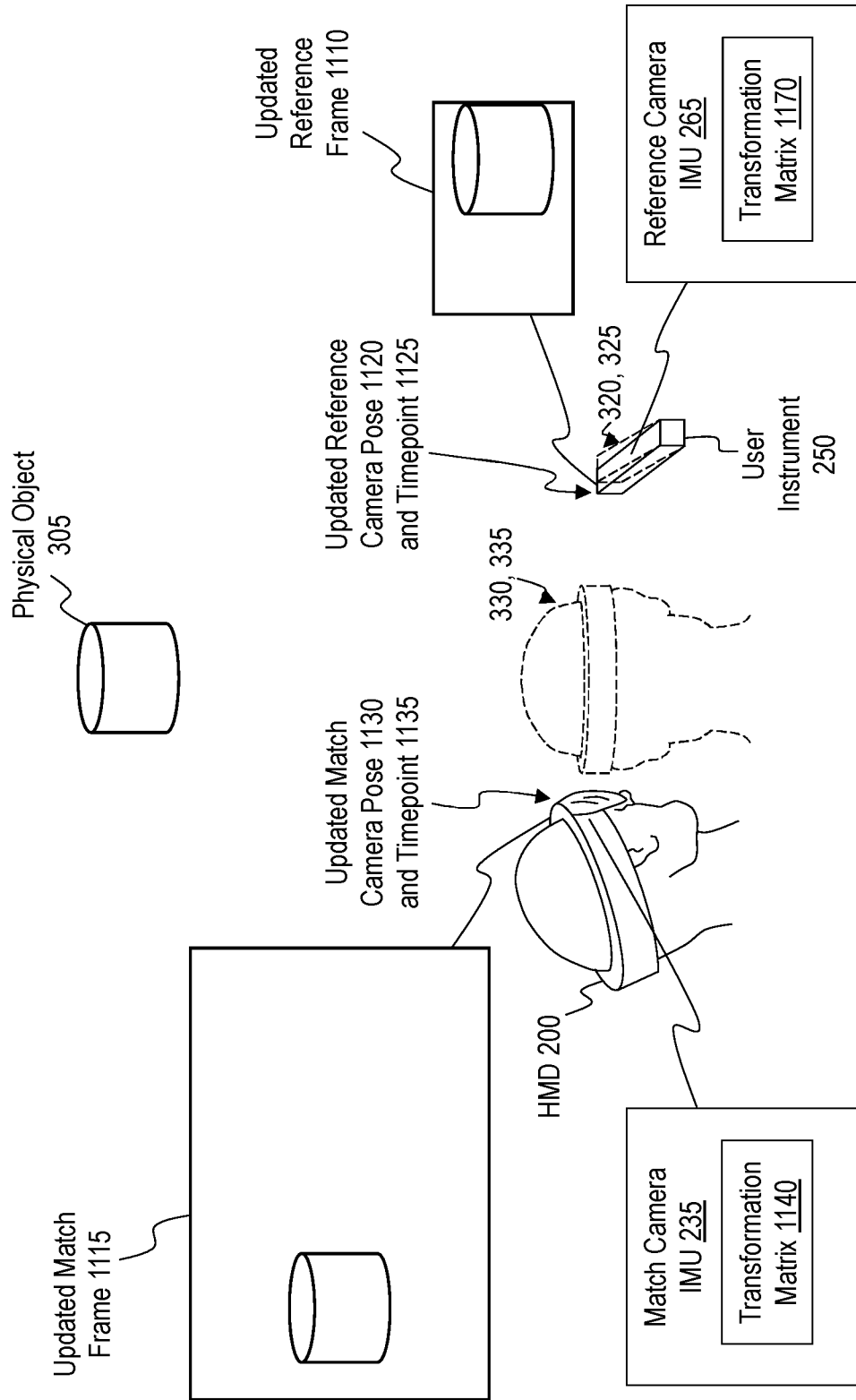


FIG. 11

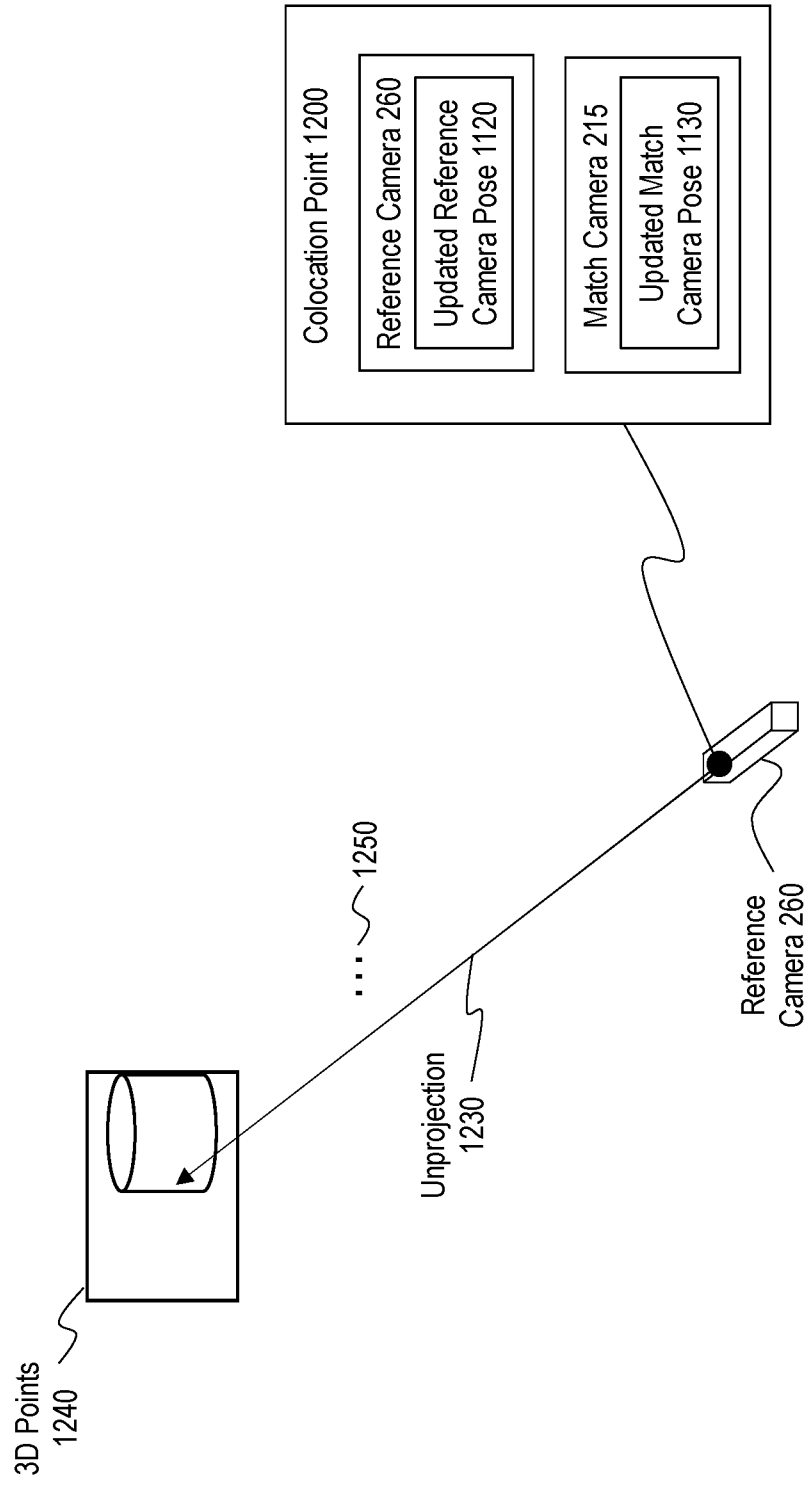


FIG. 12

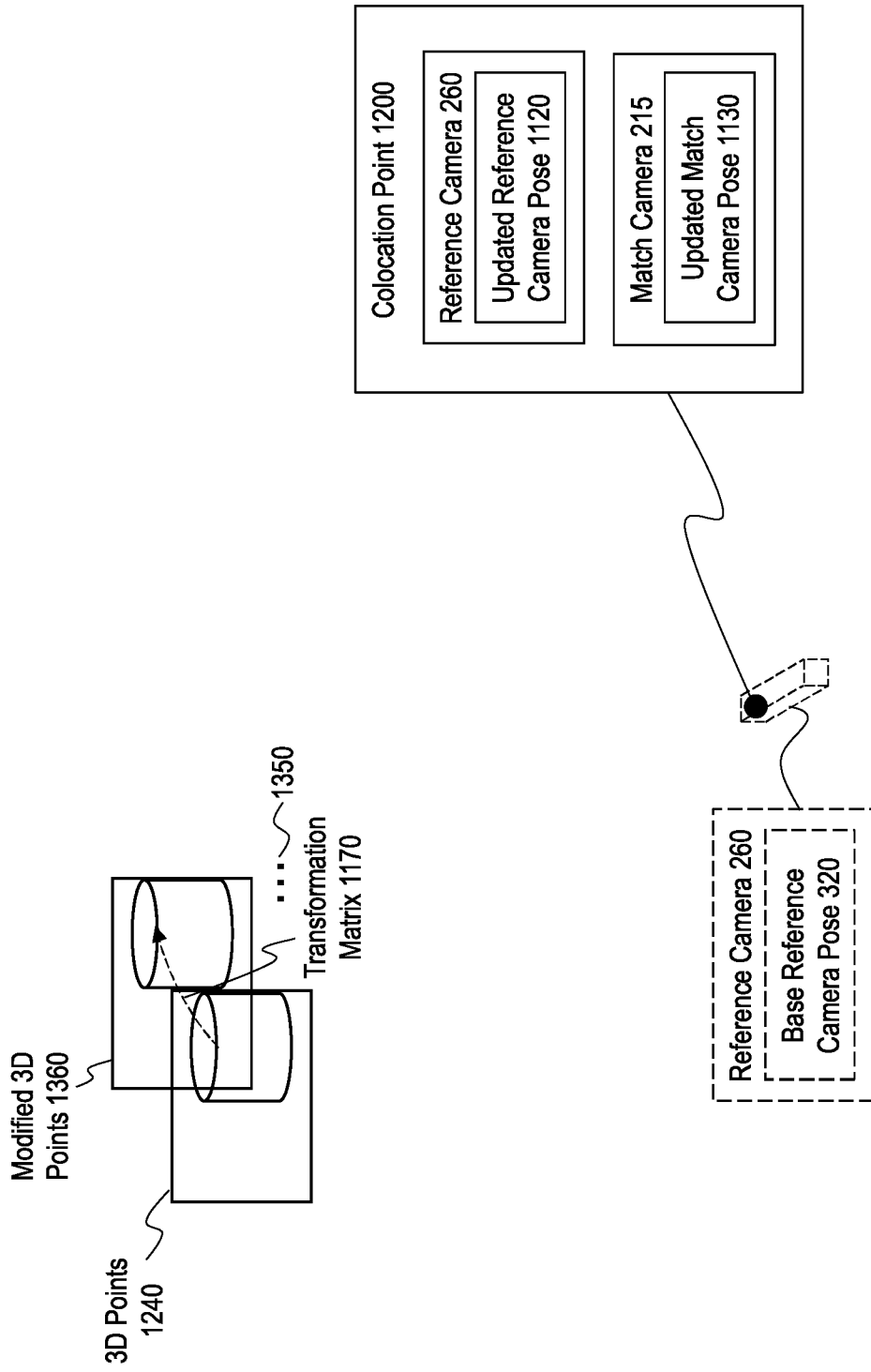


FIG. 13

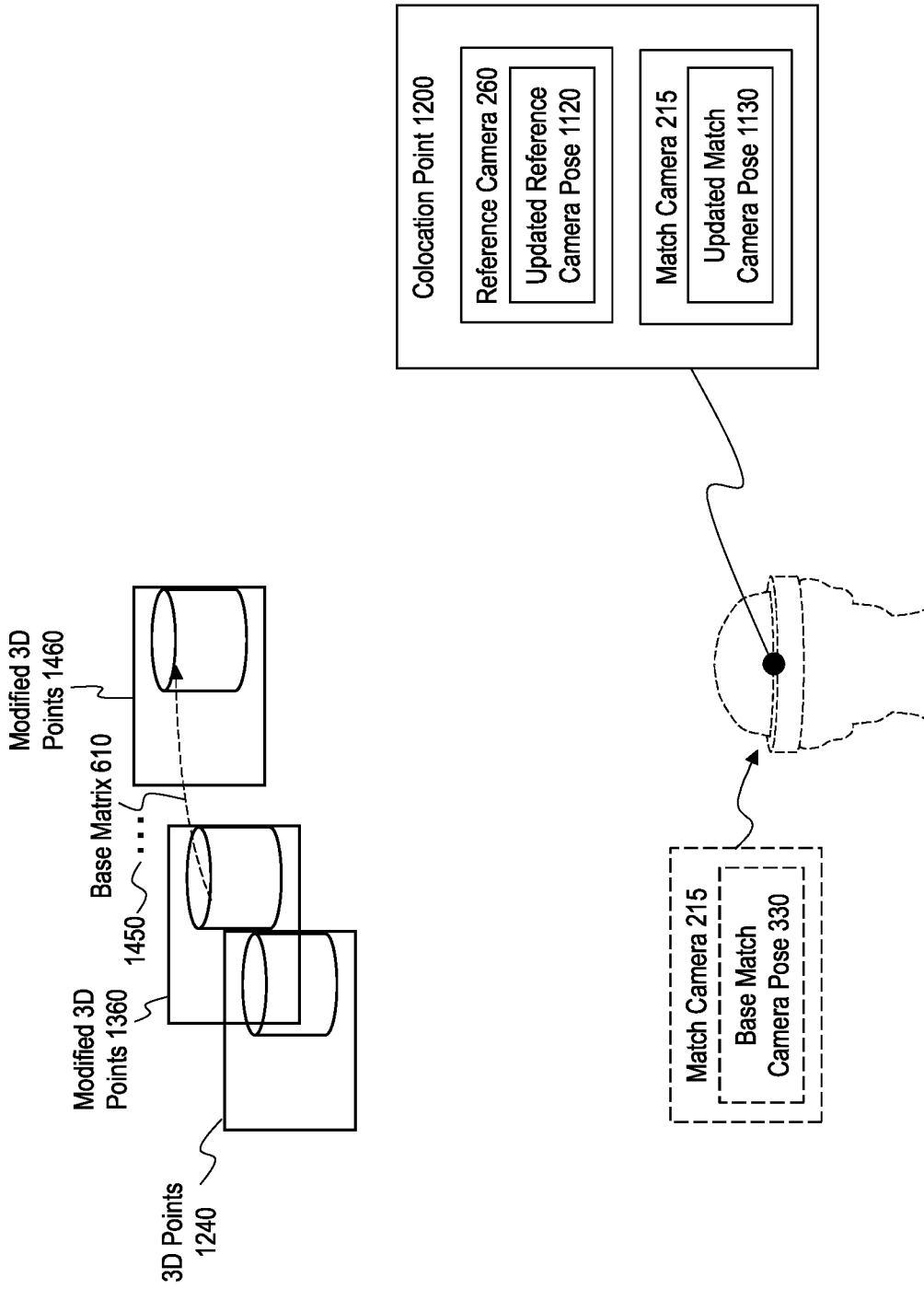


FIG. 14

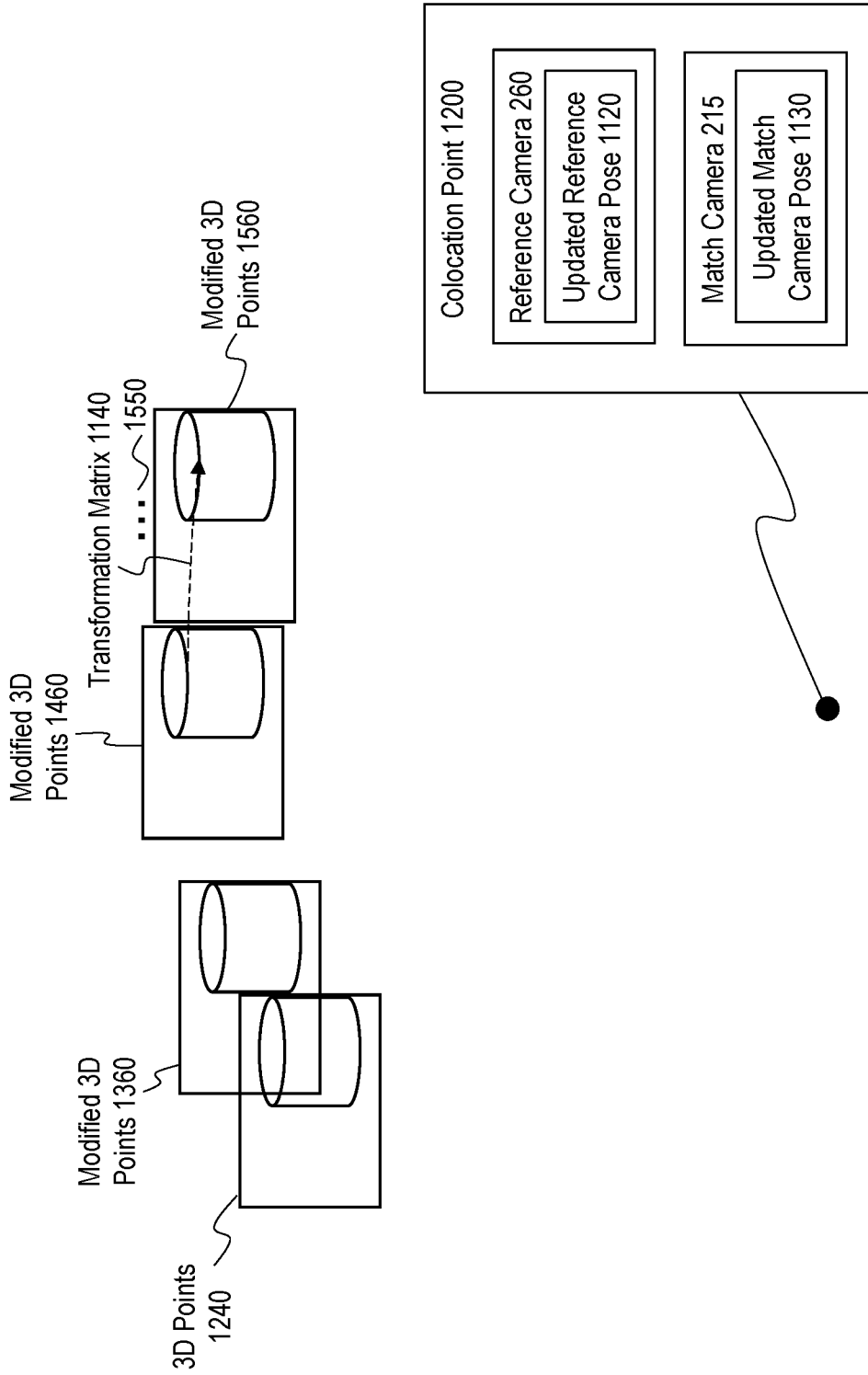


FIG. 15

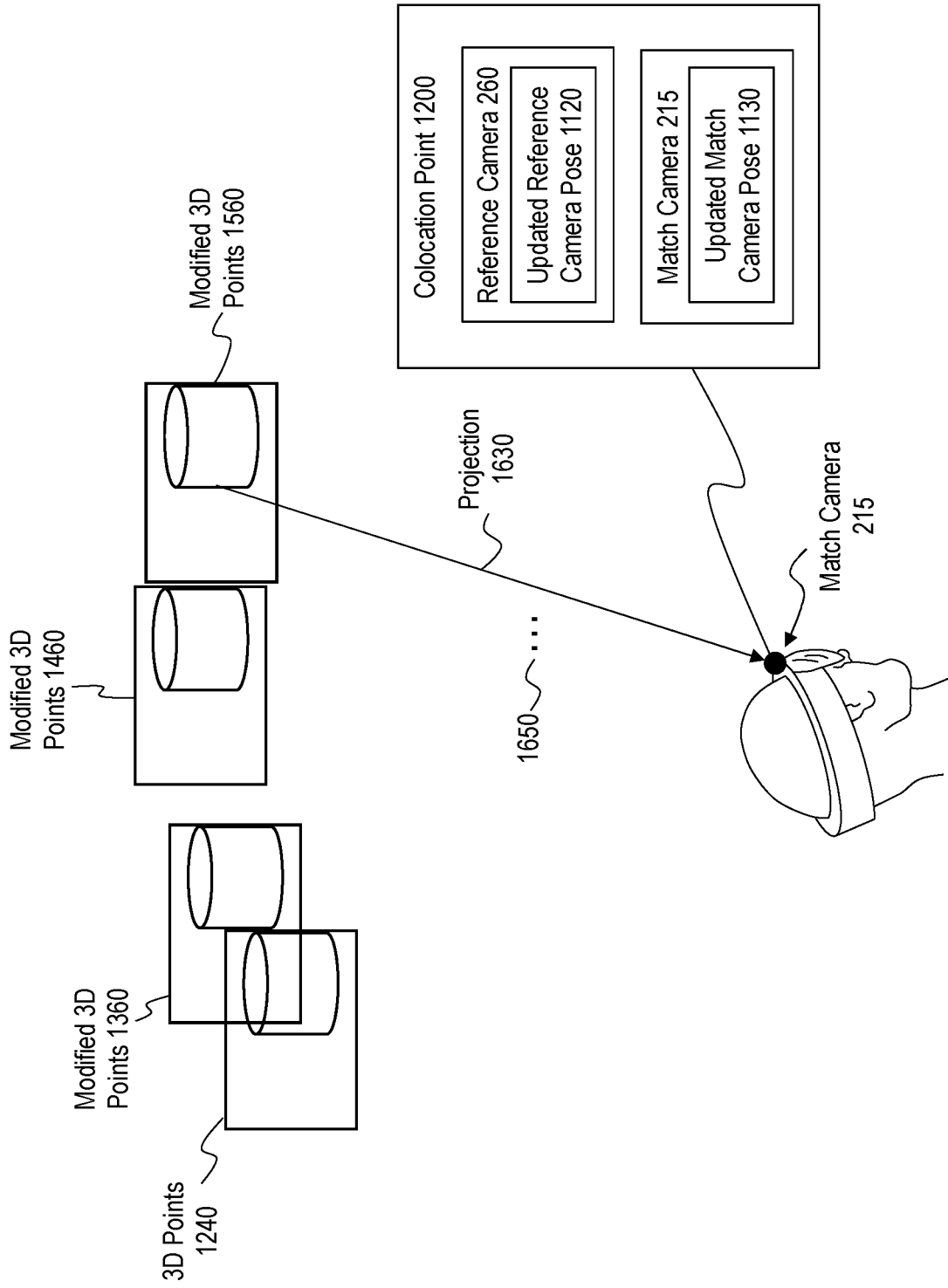


FIG. 16

17/34

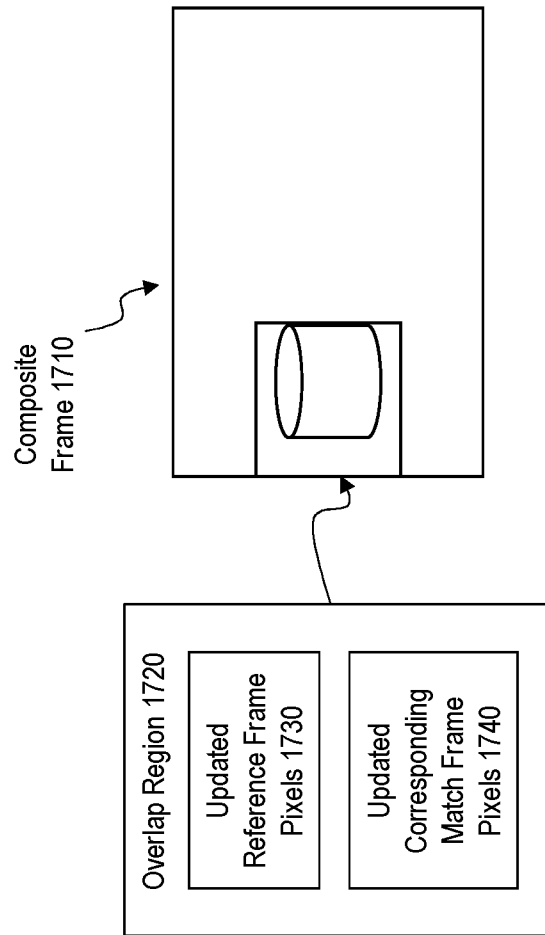


FIG. 17

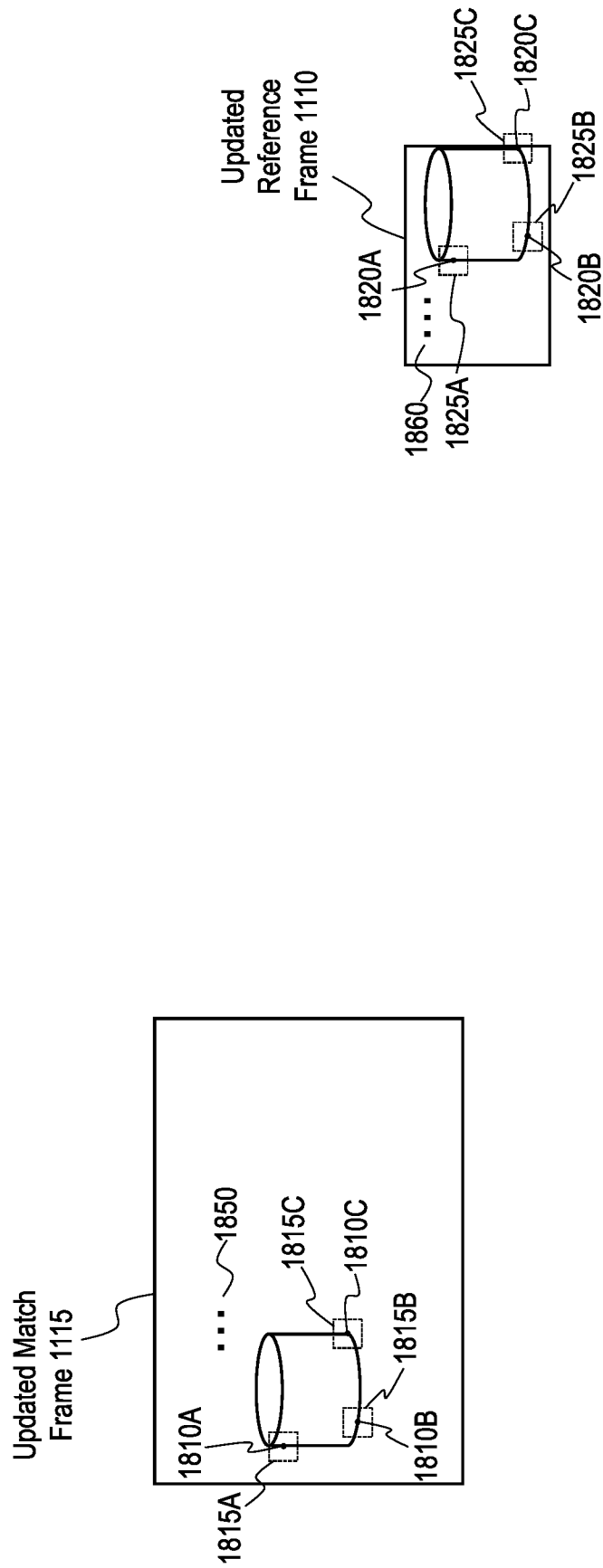


FIG. 18

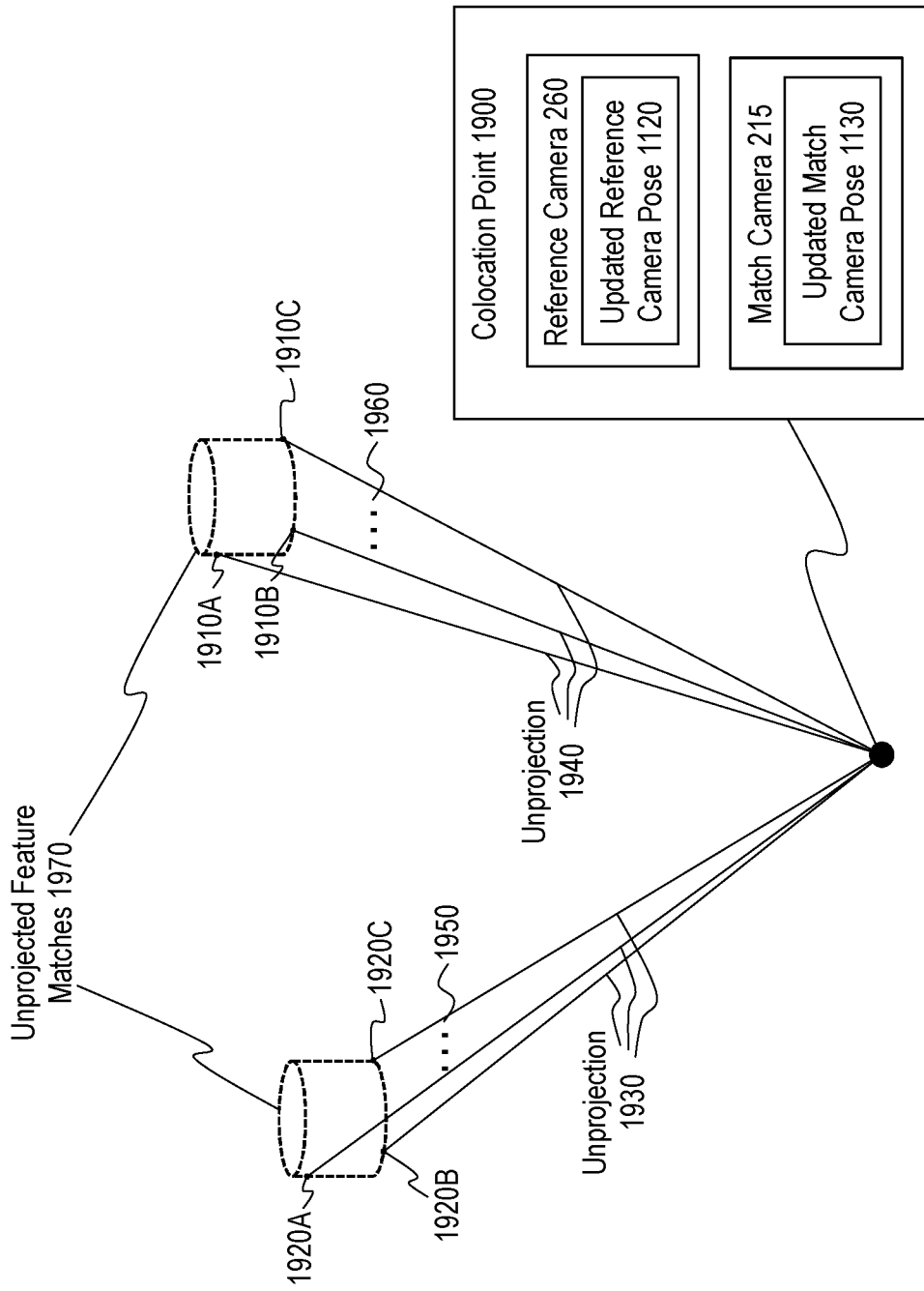


FIG. 19

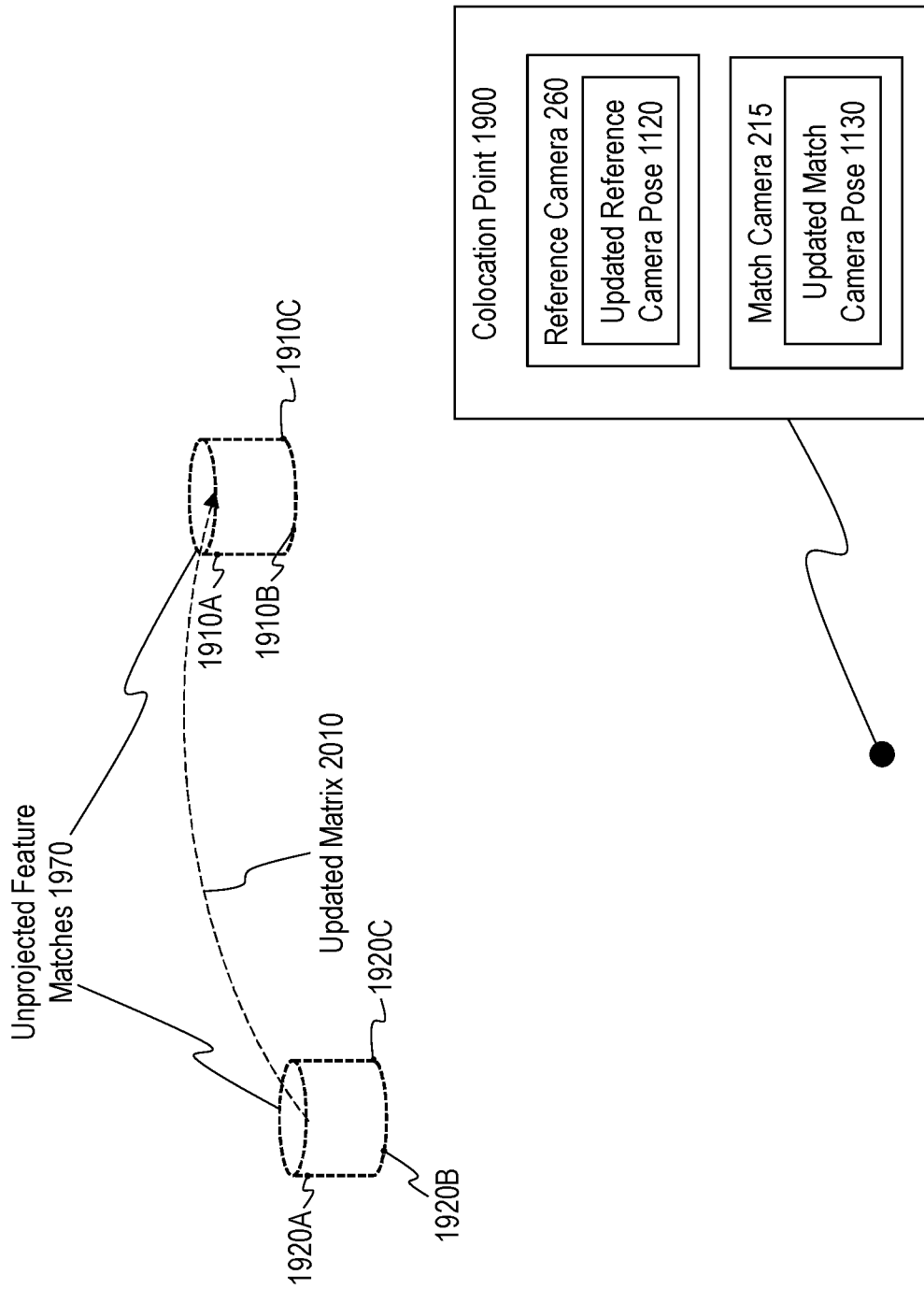


FIG. 20

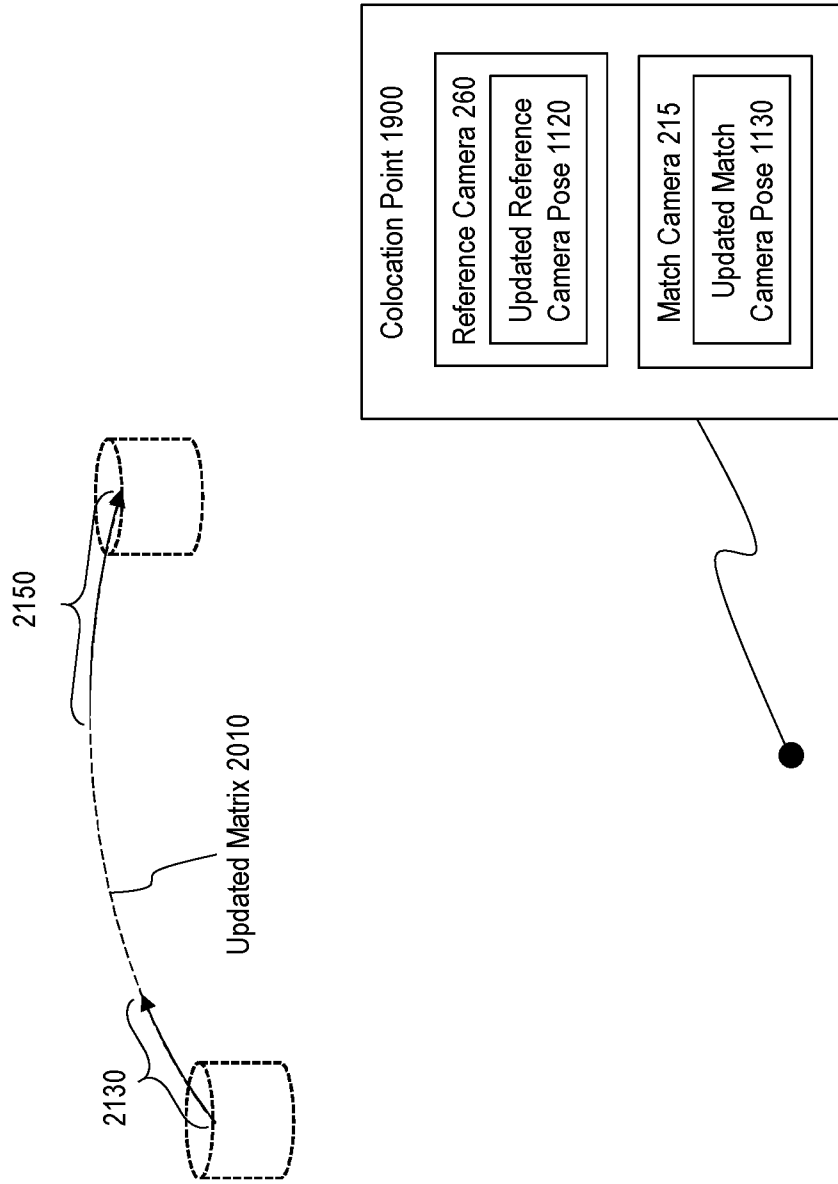


FIG. 21

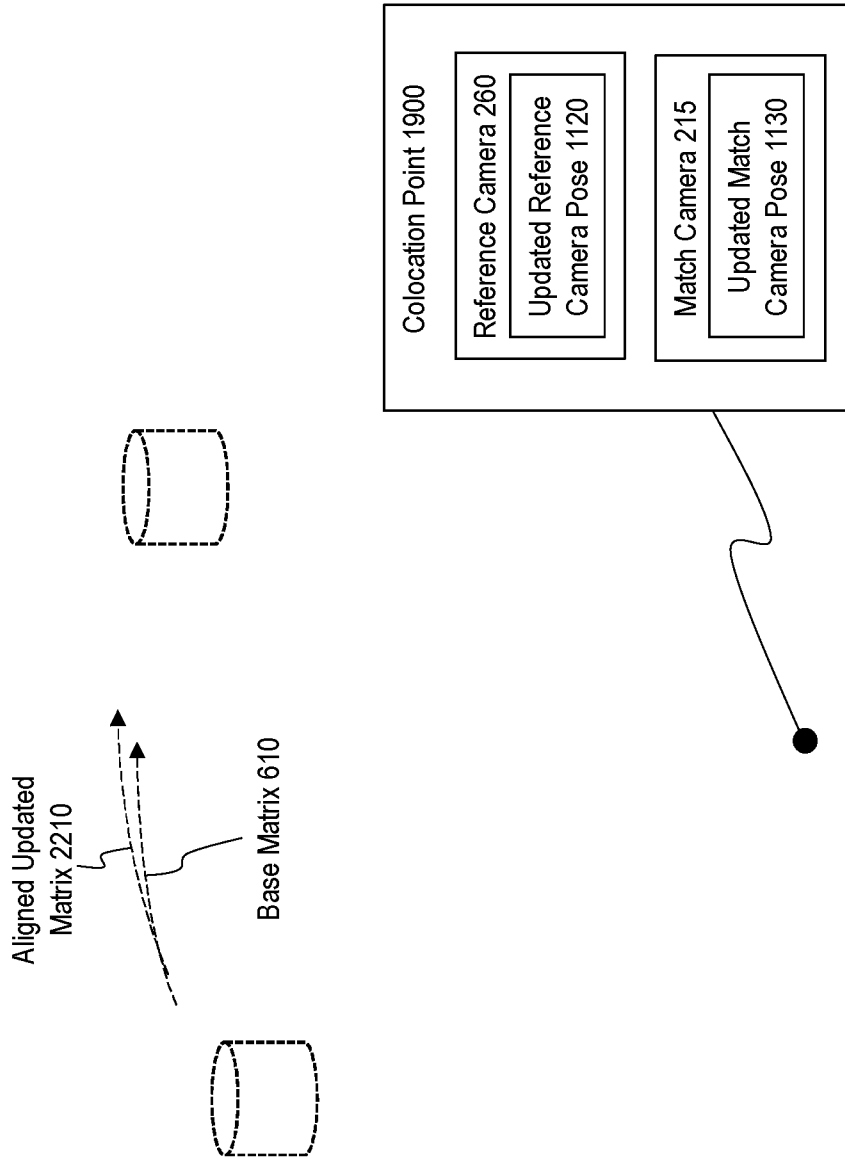


FIG. 22

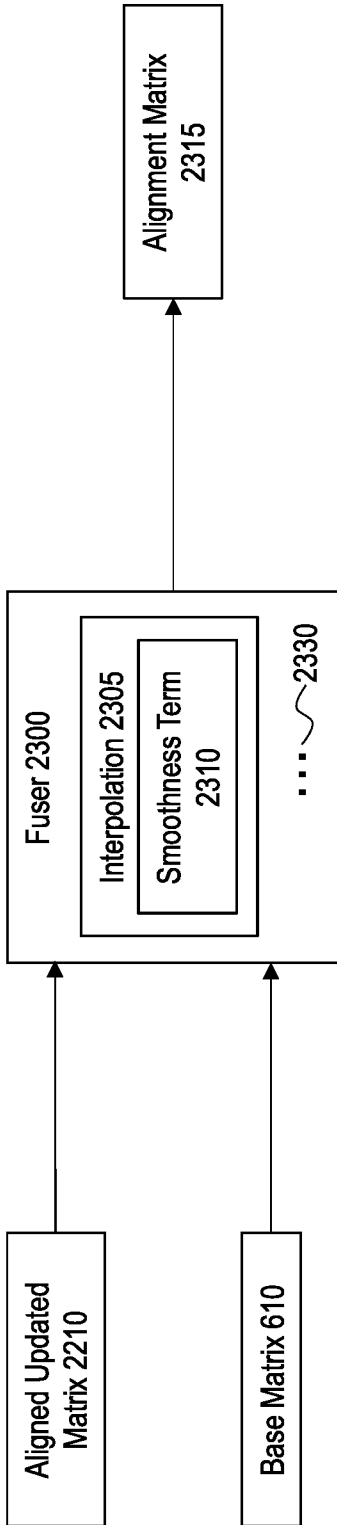


FIG. 23A

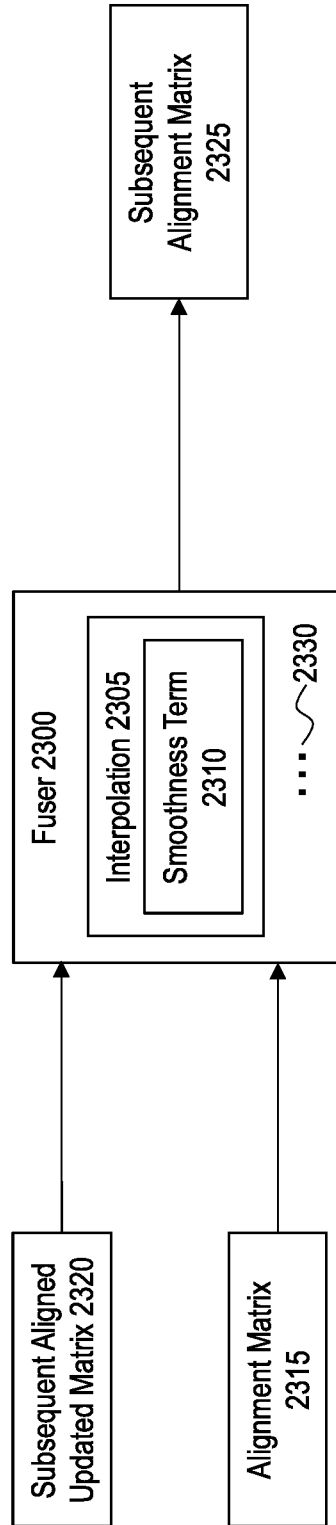


FIG. 23B

24/34

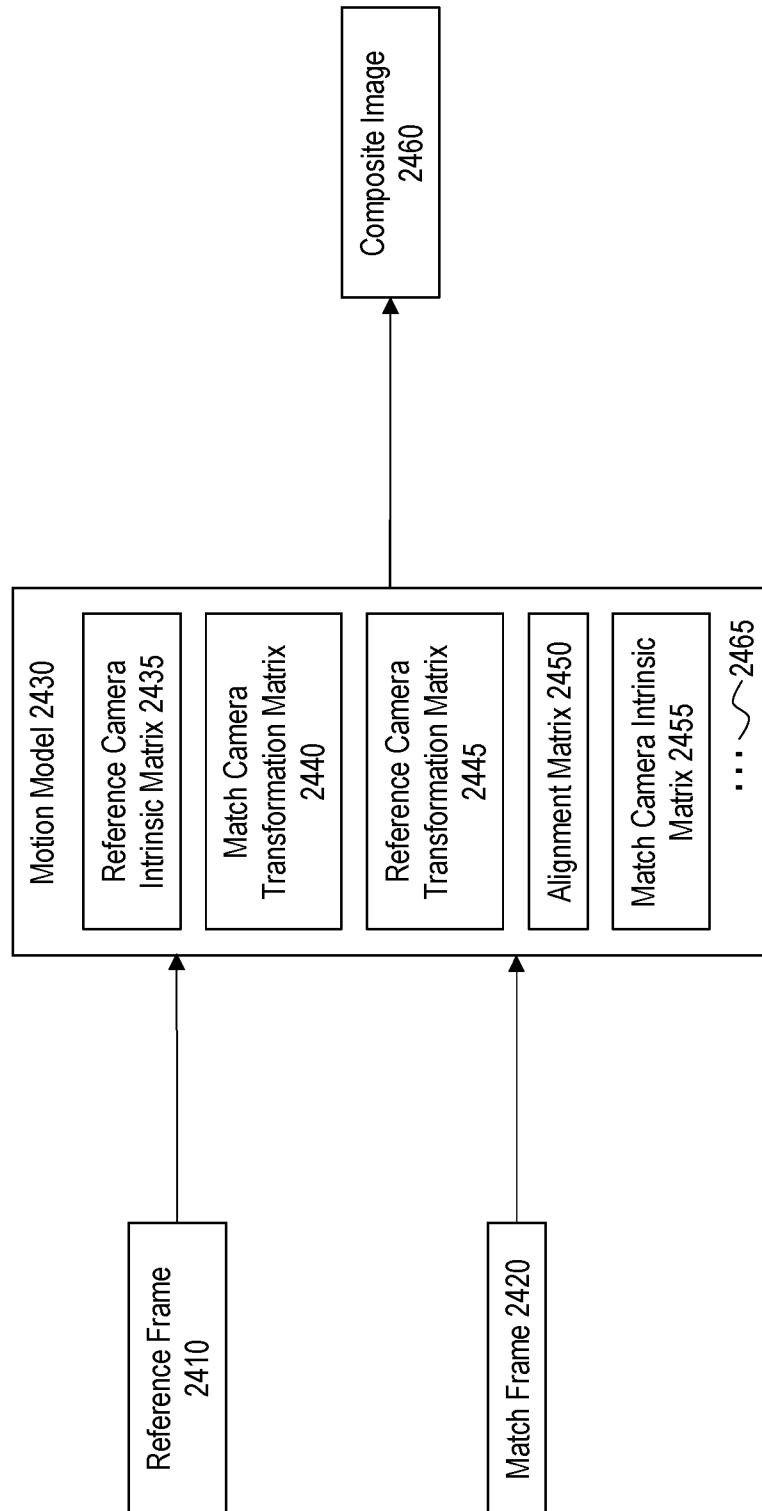


FIG. 24

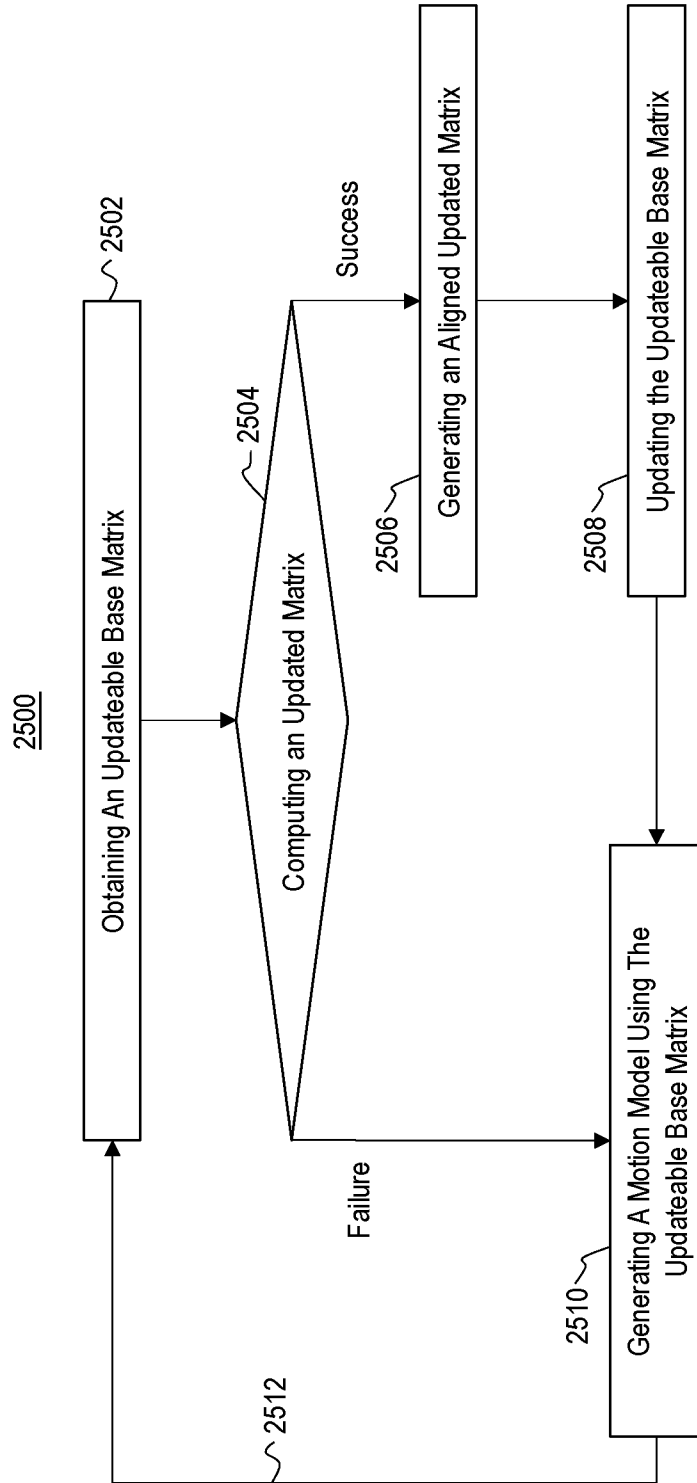


FIG. 25



FIG. 26

27/34

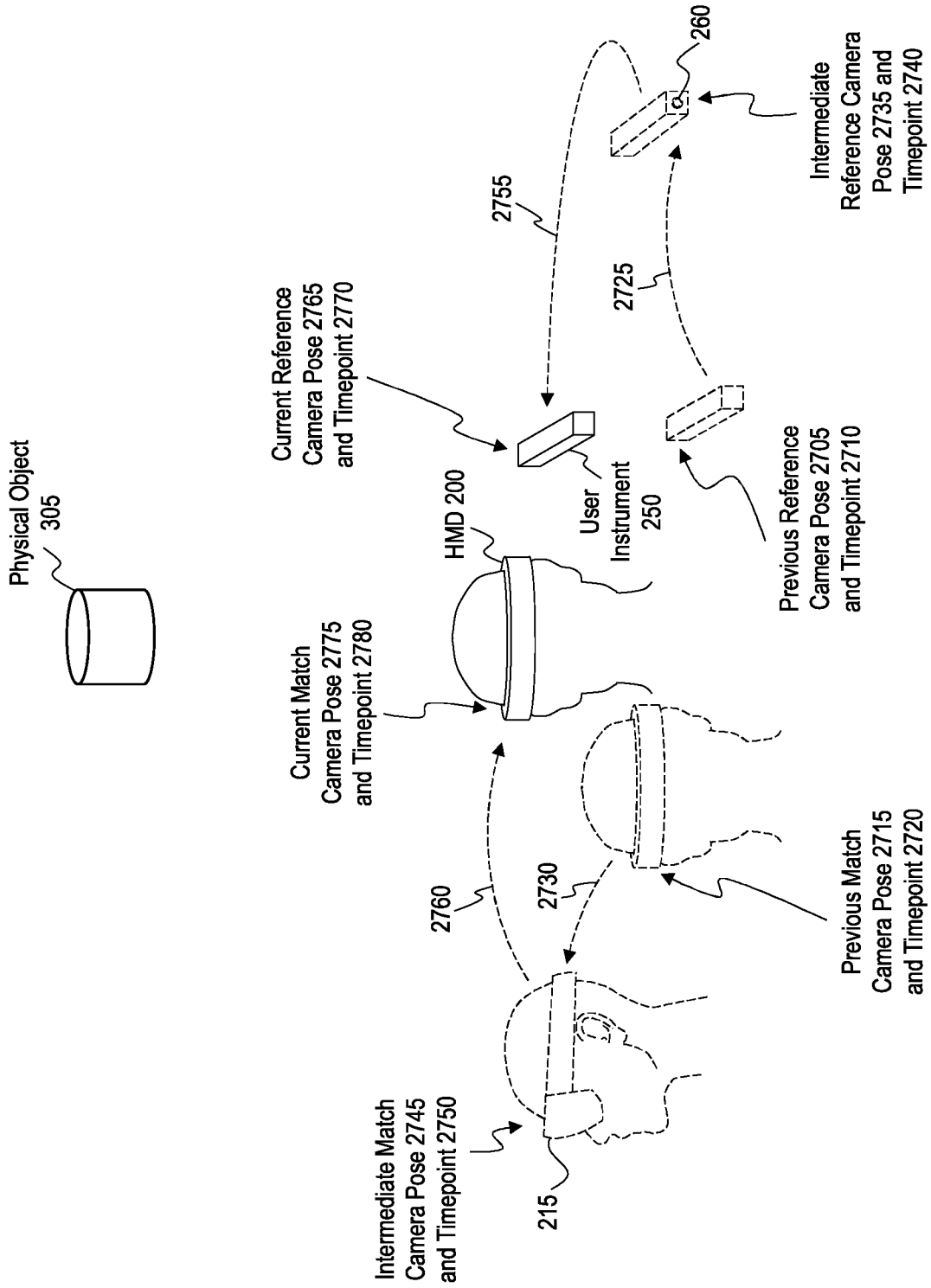


FIG. 27

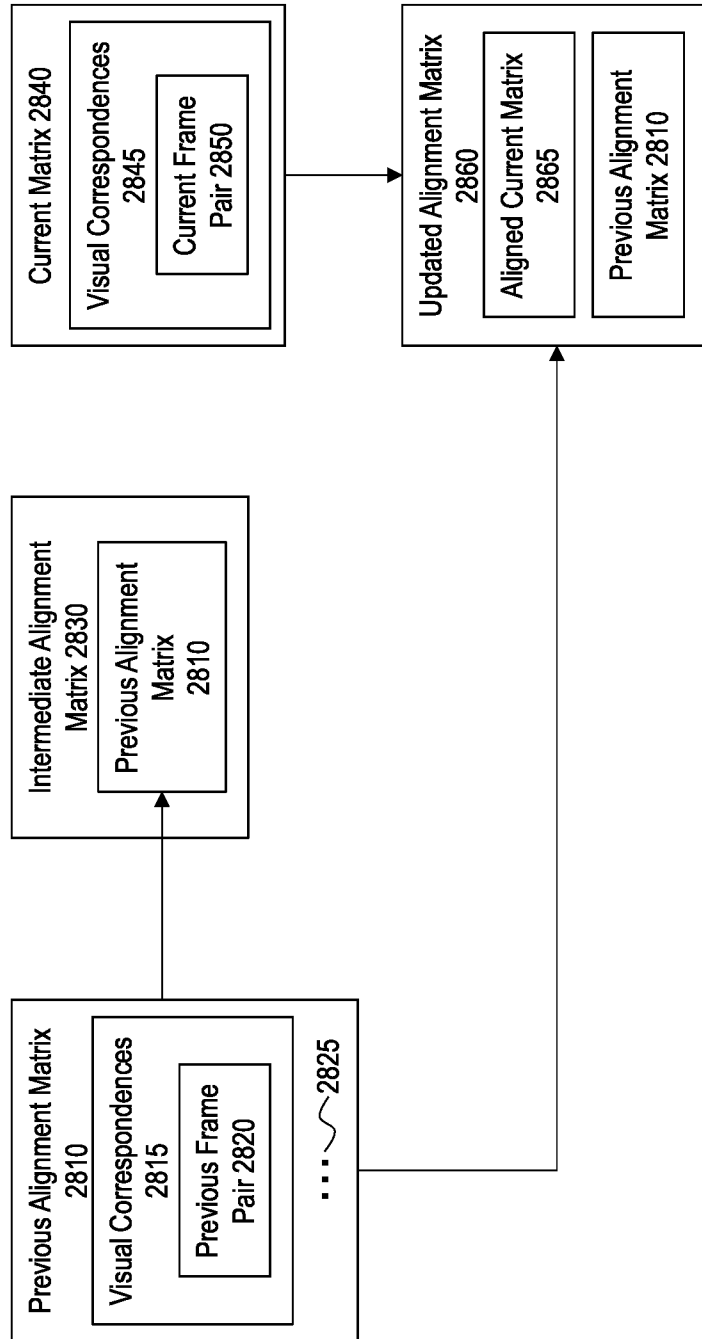


FIG. 28

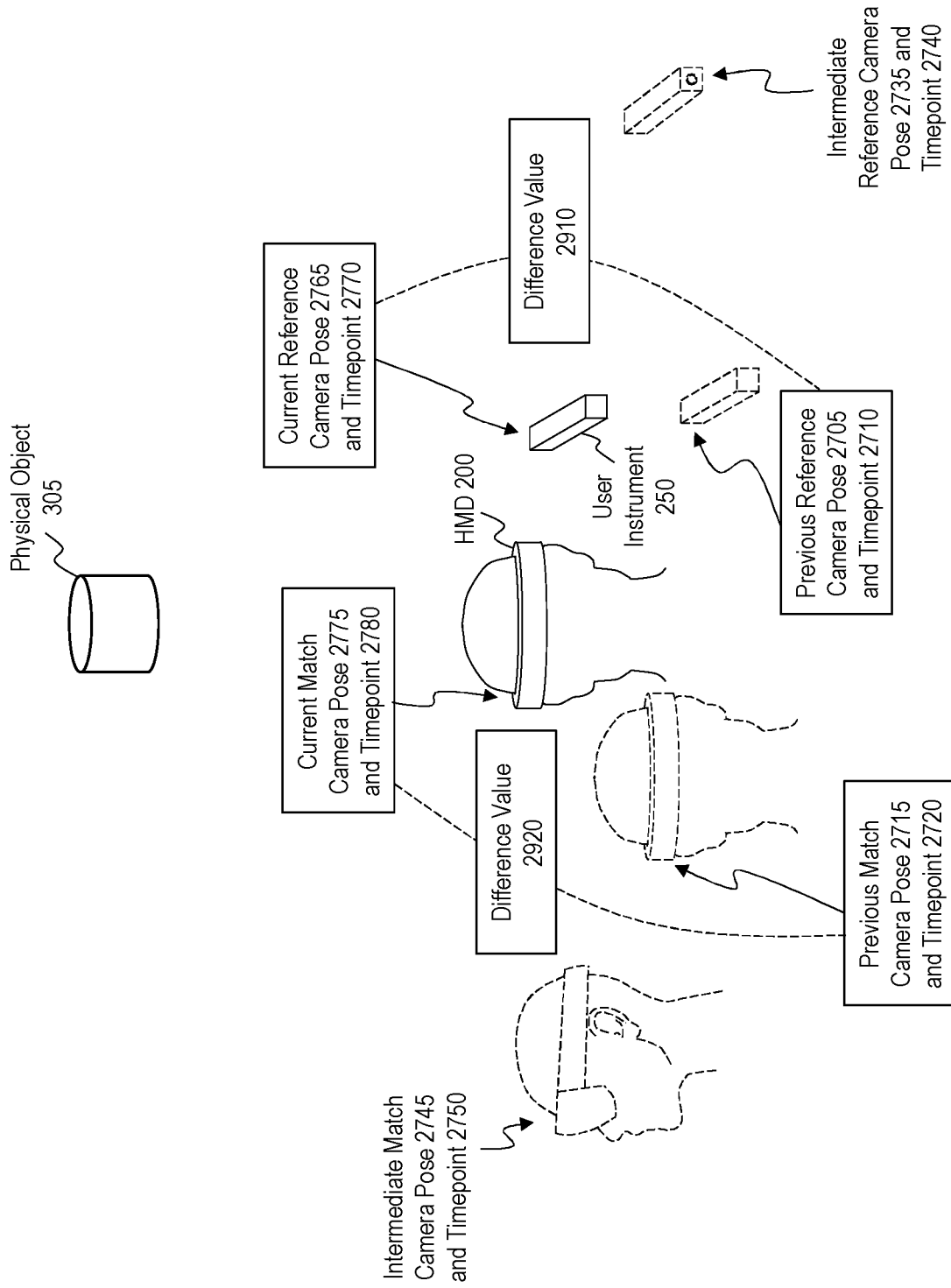


FIG. 29

30/34

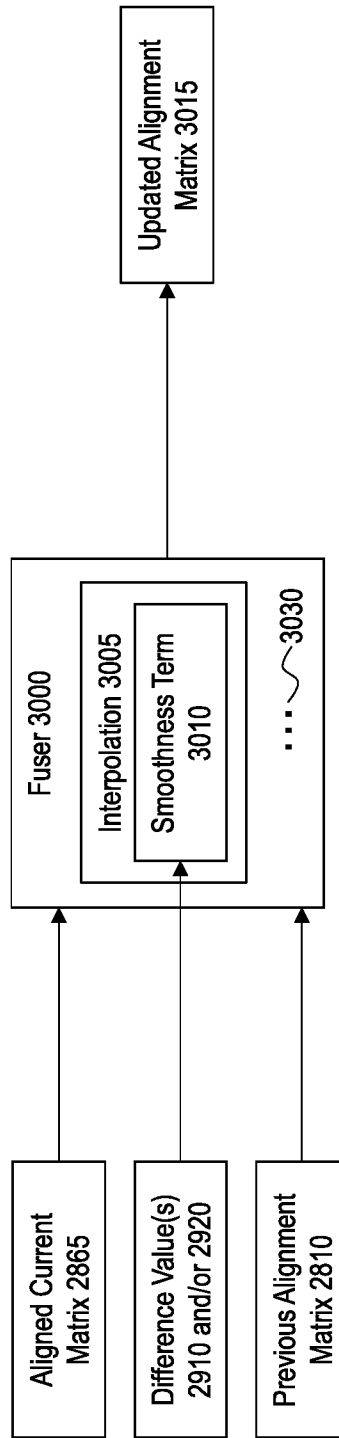


FIG. 30

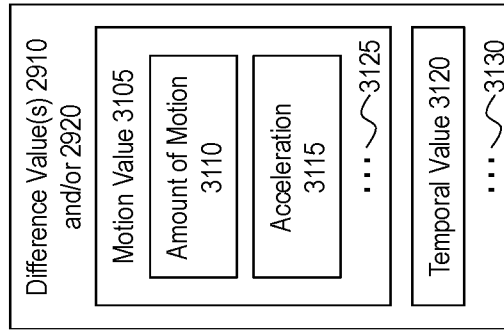


FIG. 31

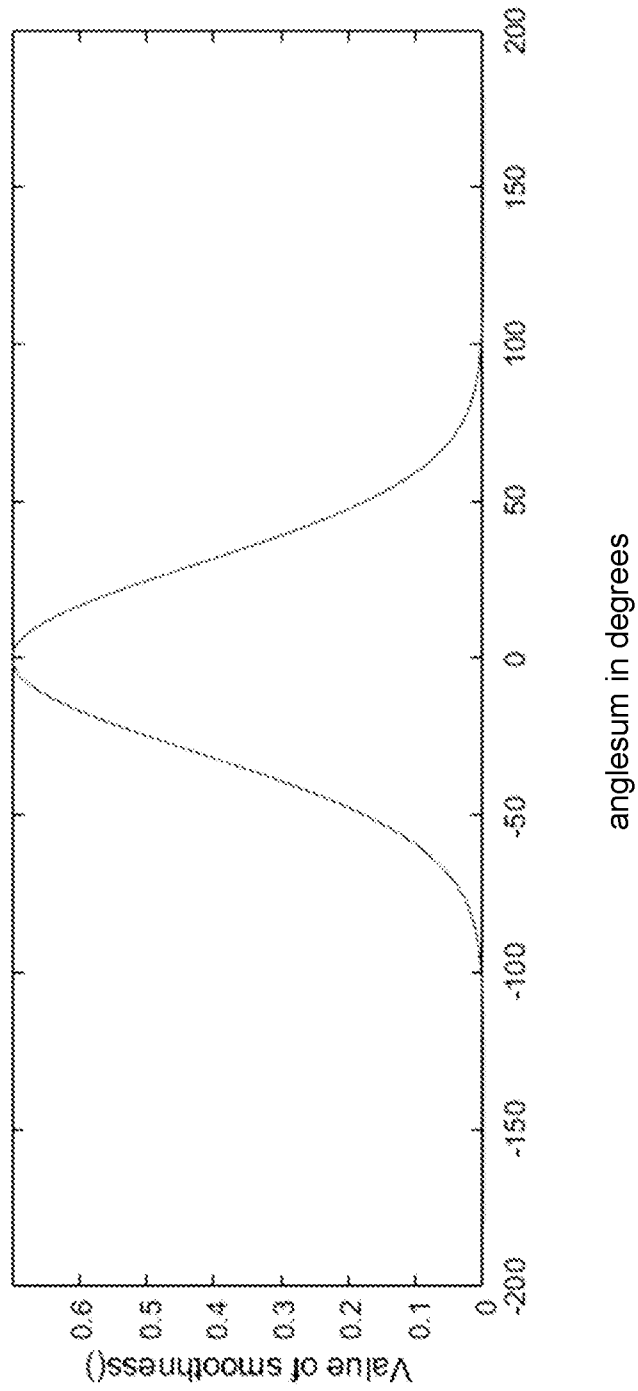


FIG. 32

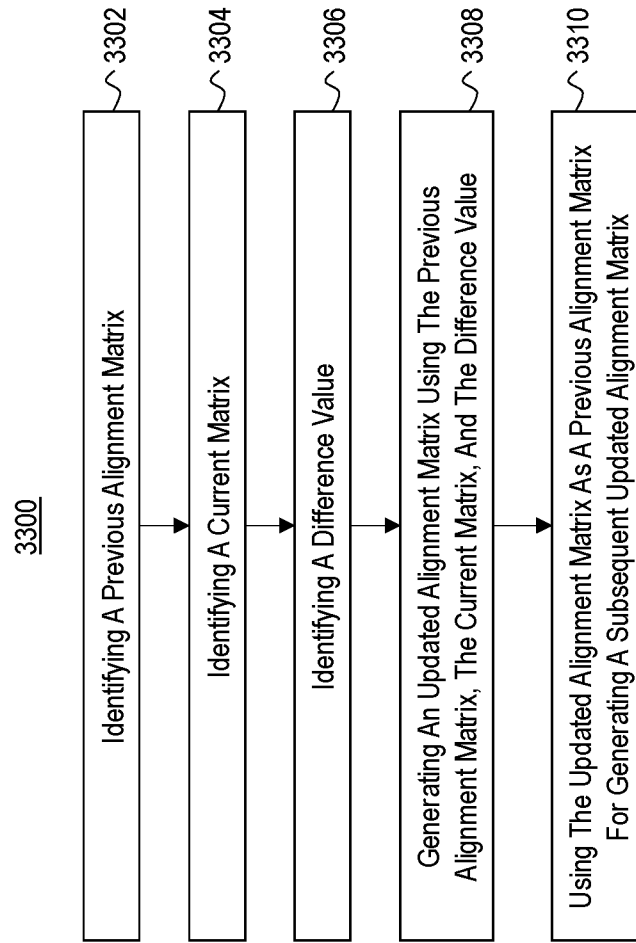


FIG. 33

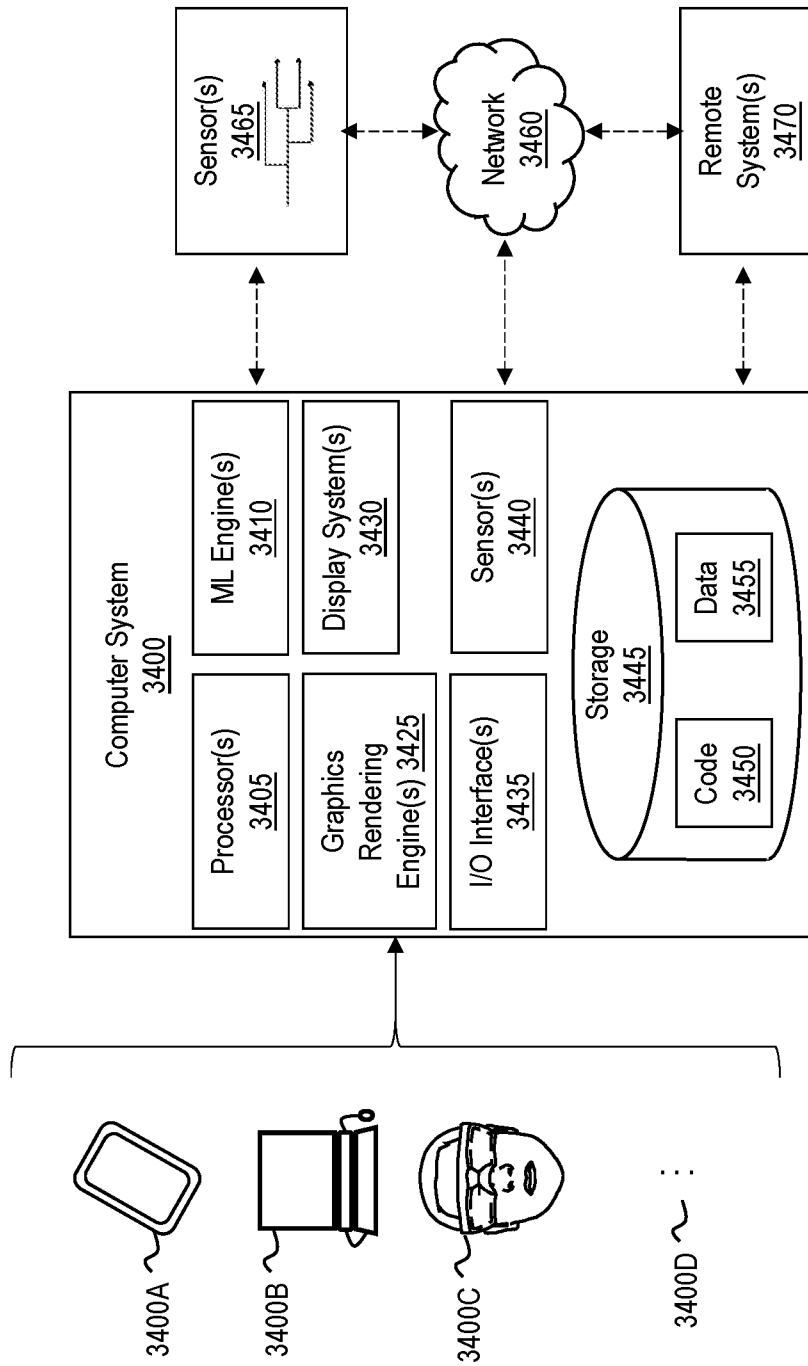


FIG. 34

INTERNATIONAL SEARCH REPORT

International application No PCT/US2021/029515

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06T7/292 G06T7/246 G06T7/33
 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.
X	XIANGYUN MENG ET AL: "SkyStitch", MULTIMEDIA, ACM, 2 PENN PLAZA, SUITE 701 NEW YORK NY 10121-0701 USA, 13 October 2015 (2015-10-13), pages 261-270, XP058076420, DOI: 10.1145/2733373.2806225 ISBN: 978-1-4503-3459-4	1-7,13
Y	figure 1 section 3, paragraphs 2-4; figure 2 section 3.2.1; figure 3 section 3.2.2; figure 4 section 3.4.1 section 3.2, paragraph 3 section 3.3, paragraph 1 section 3.4, paragraphs 1, 3 section 4, paragraph 3 -/--	8-12,14, 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	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 20 August 2021	Date of mailing of the international search report 31/08/2021
---	--

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 Kollreider, Klaus
--	---

INTERNATIONAL SEARCH REPORT

International application No PCT/US2021/029515

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p align="center">-----</p> <p>HE QINGLIN ET AL: "Stitching video streams captured by multi-UAVs with stabilization", SPIE PROCEEDINGS; [PROCEEDINGS OF SPIE ISSN 0277-786X], SPIE, US, vol. 11069, 6 May 2019 (2019-05-06), pages 110690E-110690E, XP060117962, DOI: 10.1117/12.2524245 ISBN: 978-1-5106-3673-6 section 2, paragraph 1 sections 3.1-3.2</p> <p align="center">-----</p>	8-12,14, 15