US 20240370099A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2024/0370099 A1**

YAN et al. (43) **Pub. Date: Nov. 7, 2024**

(54) **MOTION-MODEL BASED TRACKING OF ARTIFICIAL REALITY INPUT DEVICES**

(71) Applicant: **Meta Platforms Technologies, LLC**, Menlo Park, CA (US)

(72) Inventors: **Chengyuan YAN**, San Bruno, CA (US); **Sheng SHEN**, Shoreline, WA (US)

(21) Appl. No.: **18/618,421**

(22) Filed: **Mar. 27, 2024**

Related U.S. Application Data

(60) Provisional application No. 63/499,263, filed on May 1, 2023.

Publication Classification

(51) **Int. Cl.**

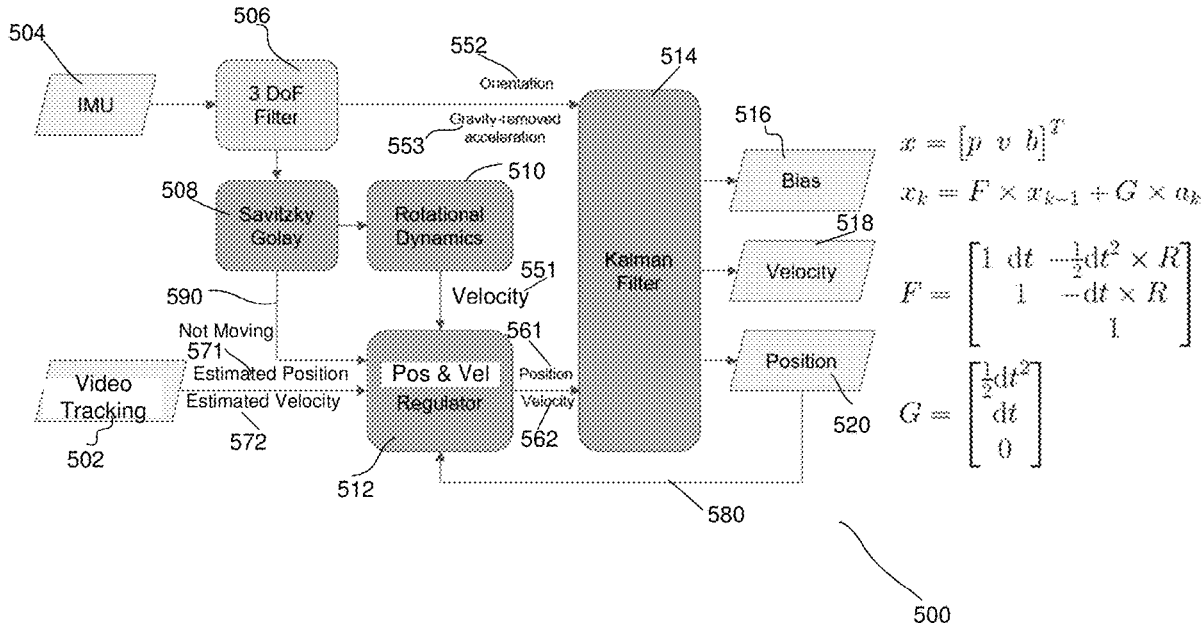
G06F 3/0346	(2013.01)
G06T 7/277	(2017.01)
G06T 7/70	(2017.01)

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

CPC ***G06F 3/0346*** (2013.01); ***G06T 7/277*** (2017.01); ***G06T 7/70*** (2017.01); *G06T 2207/10016* (2013.01)

(57) **ABSTRACT**

Example implementations are for tracking an artificial reality input device by receiving, for the input device, video tracking data and inertial motion unit ("IMU") data based on motion input. Example implementations generate, from the video tracking data, a video tracking position and a video tracking velocity; generate, from the IMU data, an IMU orientation and an IMU linear acceleration; and generate, from the IMU orientation and the IMU linear acceleration, an IMU linear velocity. Example implementations determine if the video tracking position and the video tracking velocity is reliable and determine, by a Kalman filter for the input device, a current bias, a current velocity and a current position.



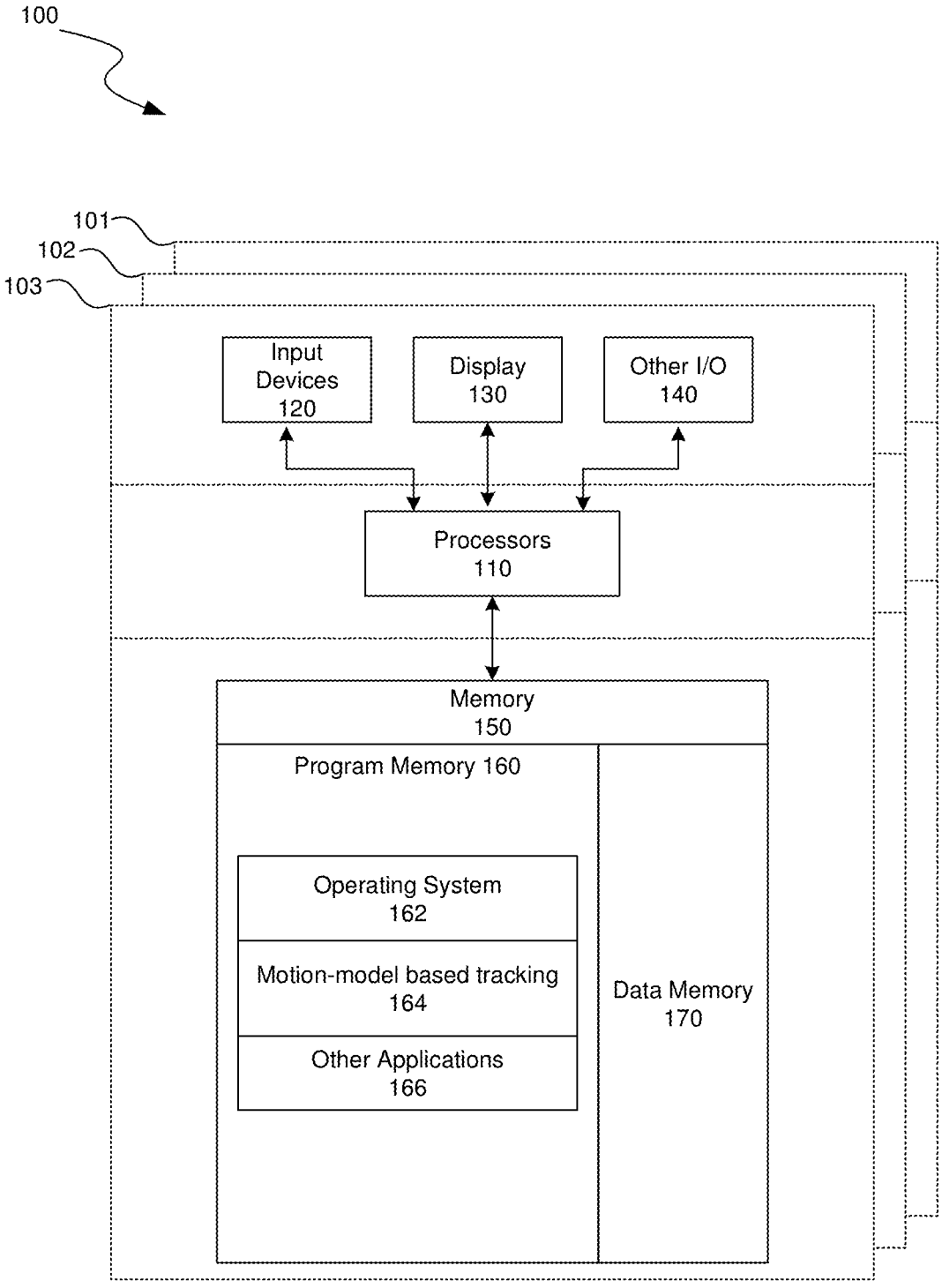

100
101
102
103
Input Devices 120
Display 130
Other I/O 140
Processors 110
Memory 150
Program Memory 160
Operating System 162
Motion-model based tracking 164
Other Applications 166
Data Memory 170

FIG. 1

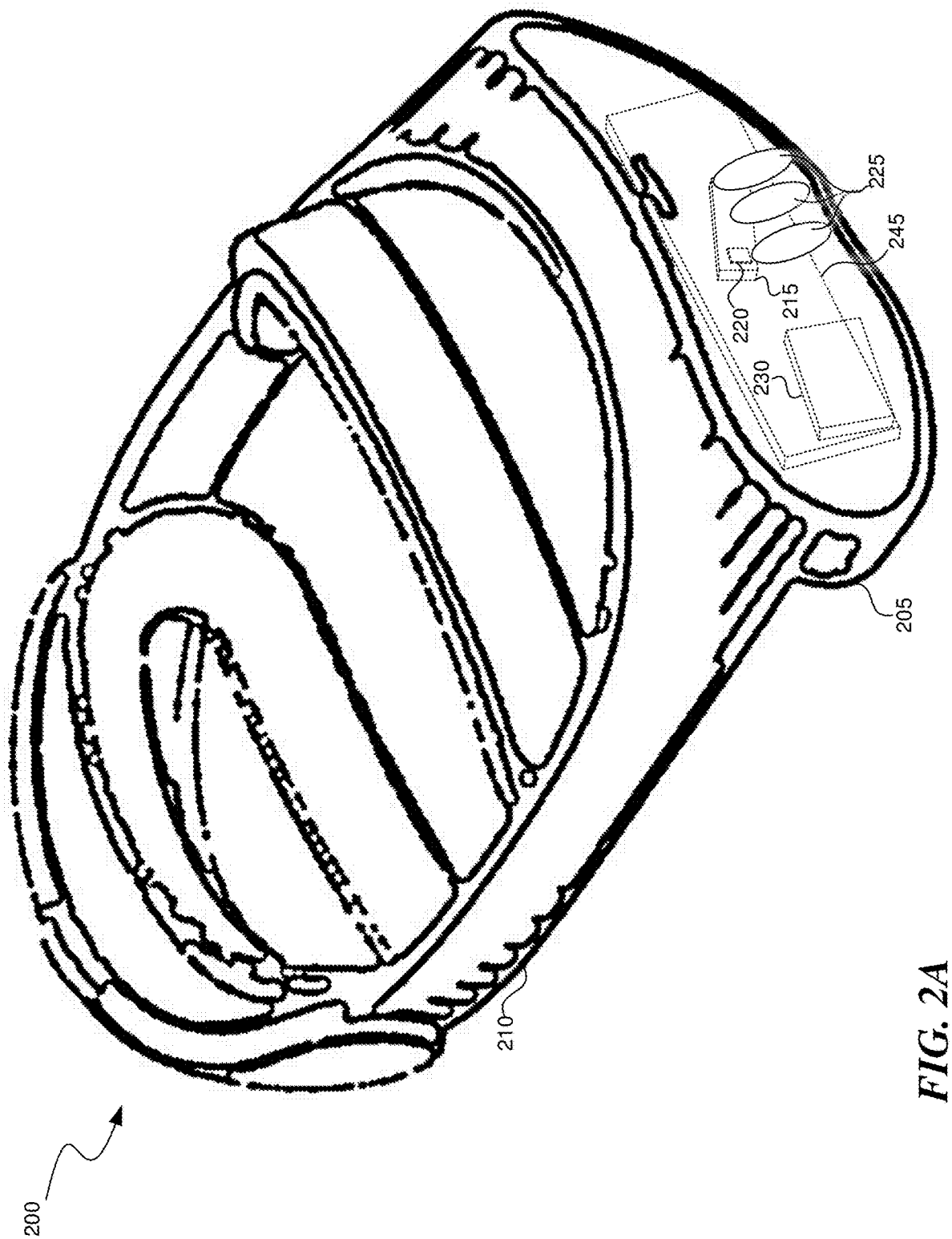


FIG. 2A

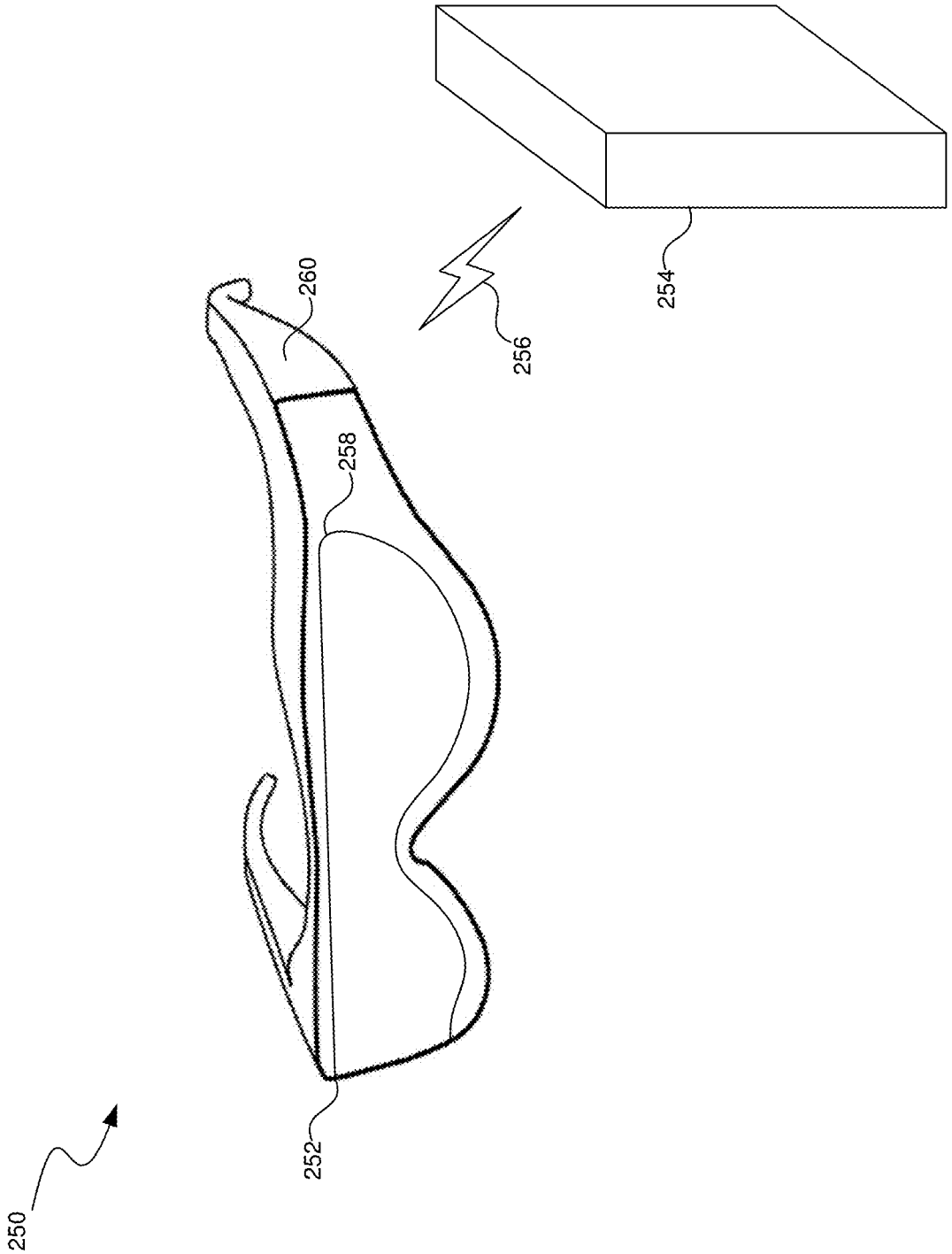


FIG. 2B

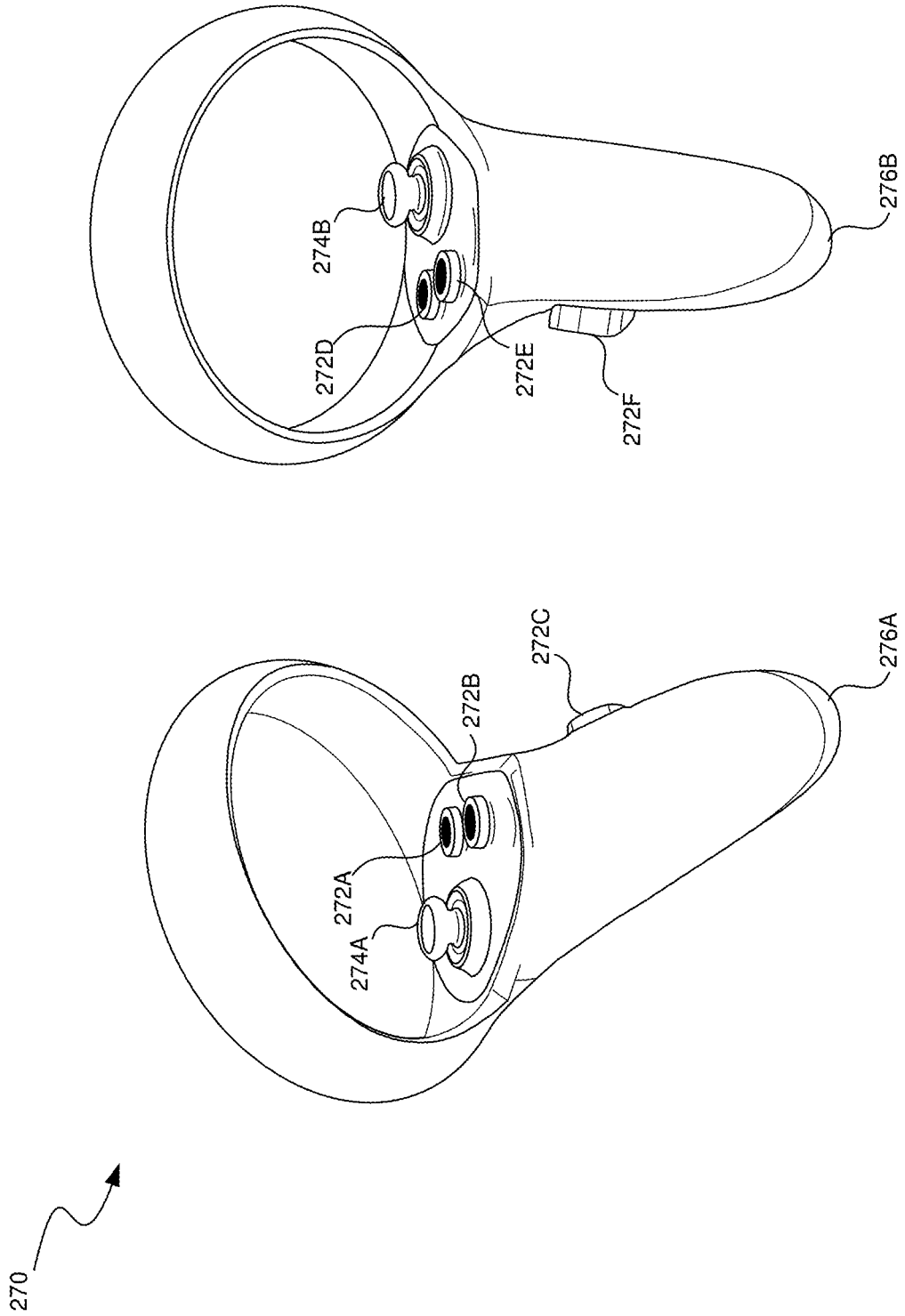


FIG. 2C

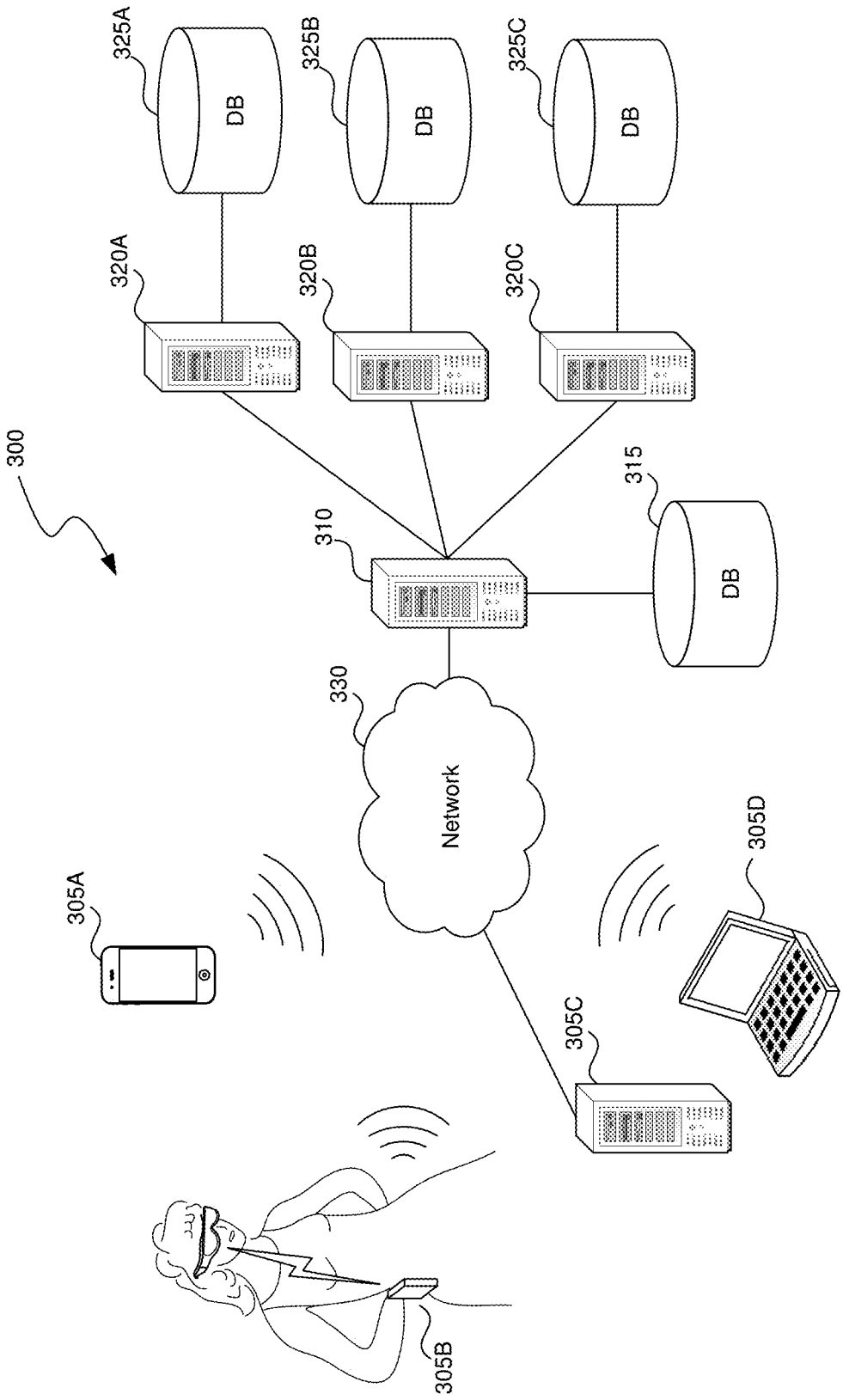

300
325A
DB
325B
DB
325C
DB
320A
320B
320C
315
DB
310
330
Network
305A
305D
305C
305B

FIG. 3

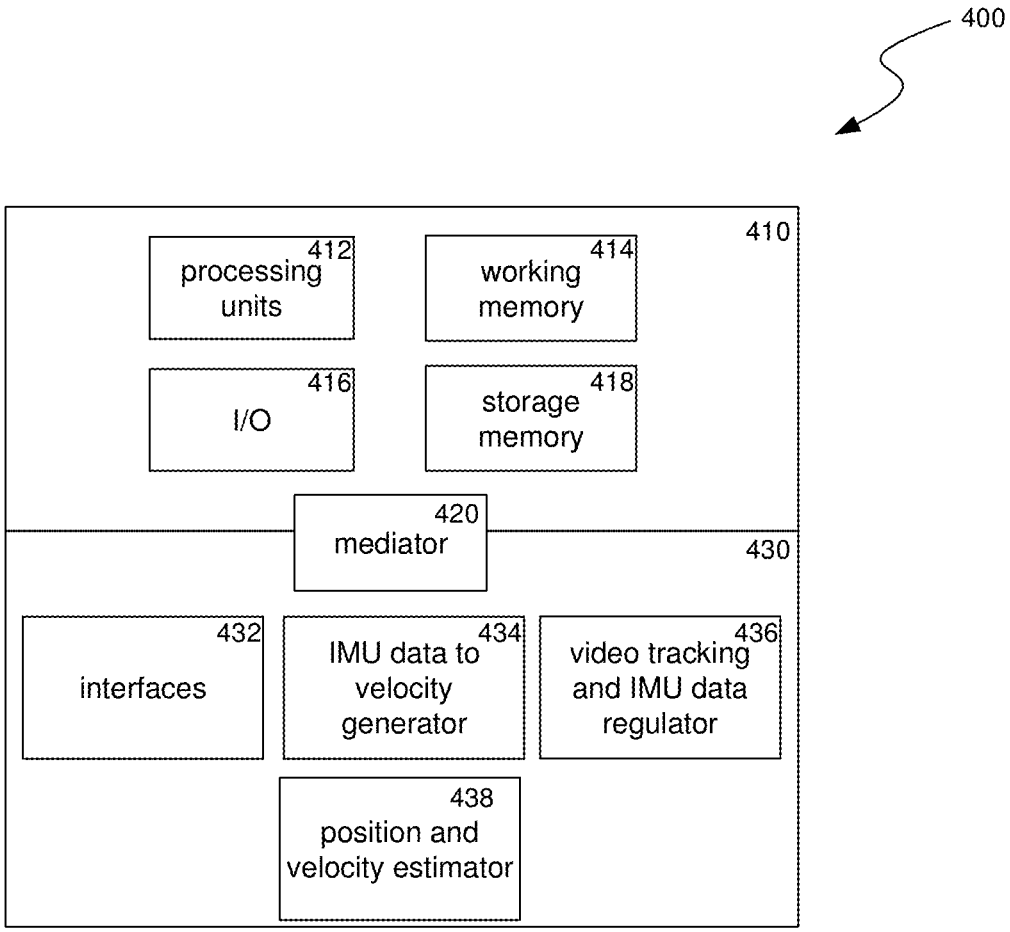

400
410
412
processing units
414
working memory
416
I/O
418
storage memory
420
mediator
430
432
interfaces
434
IMU data to velocity generator
436
video tracking and IMU data regulator
438
position and velocity estimator

FIG. 4

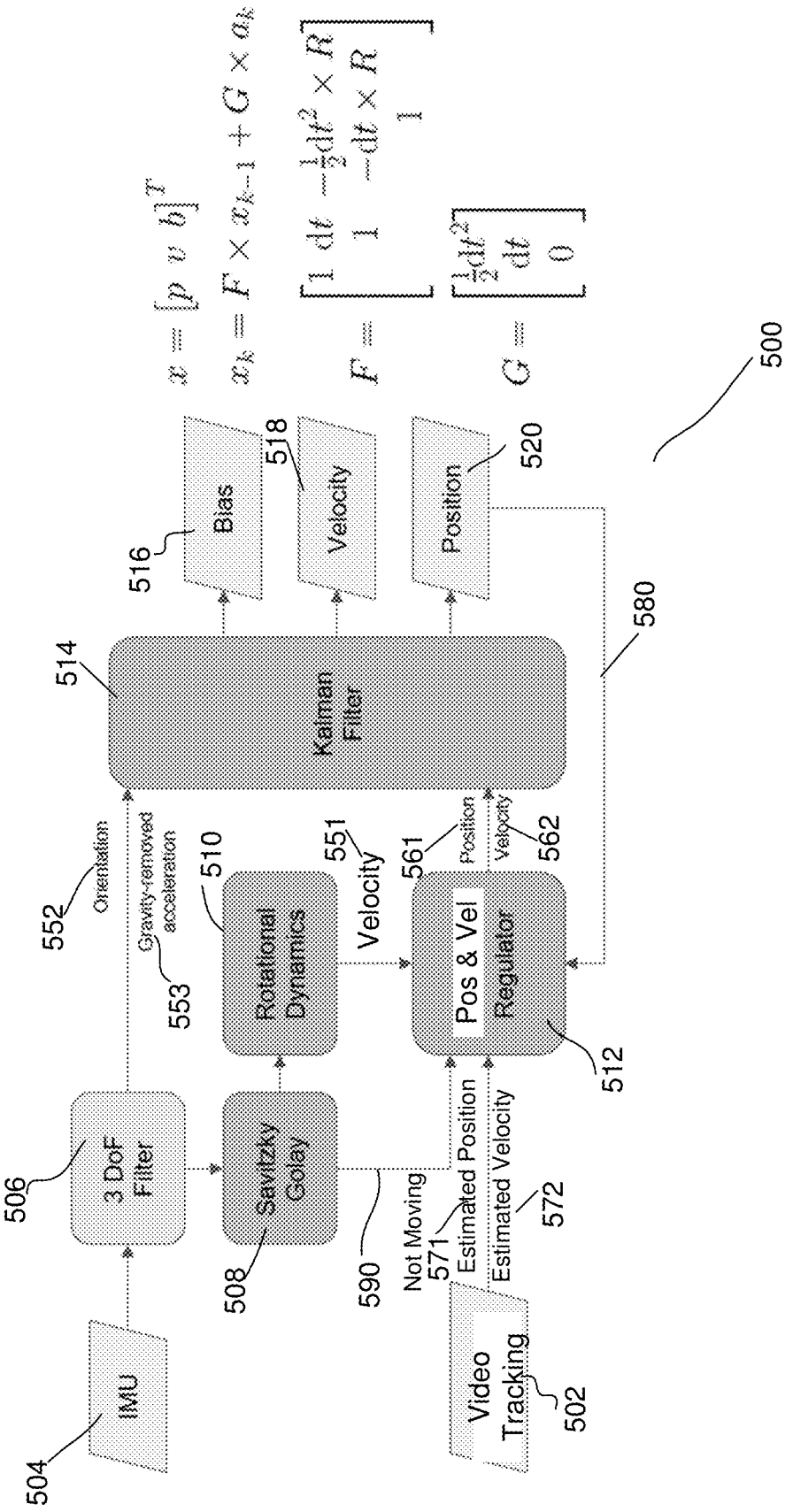

$x = [p \ v \ b]^T$
$x_k = F \times x_{k-1} + G \times a_k$
$F = \begin{bmatrix} 1 & dt & -\frac{1}{2}dt^2 \times R \\ 1 & -dt \times R \\ & & 1 \end{bmatrix}$
$G = \begin{bmatrix} \frac{1}{2}dt^2 \\ dt \\ 0 \end{bmatrix}$
500
IMU
504
Video Tracking
502
3 DoF Filter
506
Savitzky Golay
508
Rotational Dynamics
510
Pos & Vel Regulator
512
Kalman Filter
514
Bias
516
Velocity
518
Position
520
Orientation
552
Gravity-removed acceleration
553
Velocity
551
Position
561
Velocity
562
Not Moving
590
Estimated Position
571
Estimated Velocity
572
580

FIG. 5

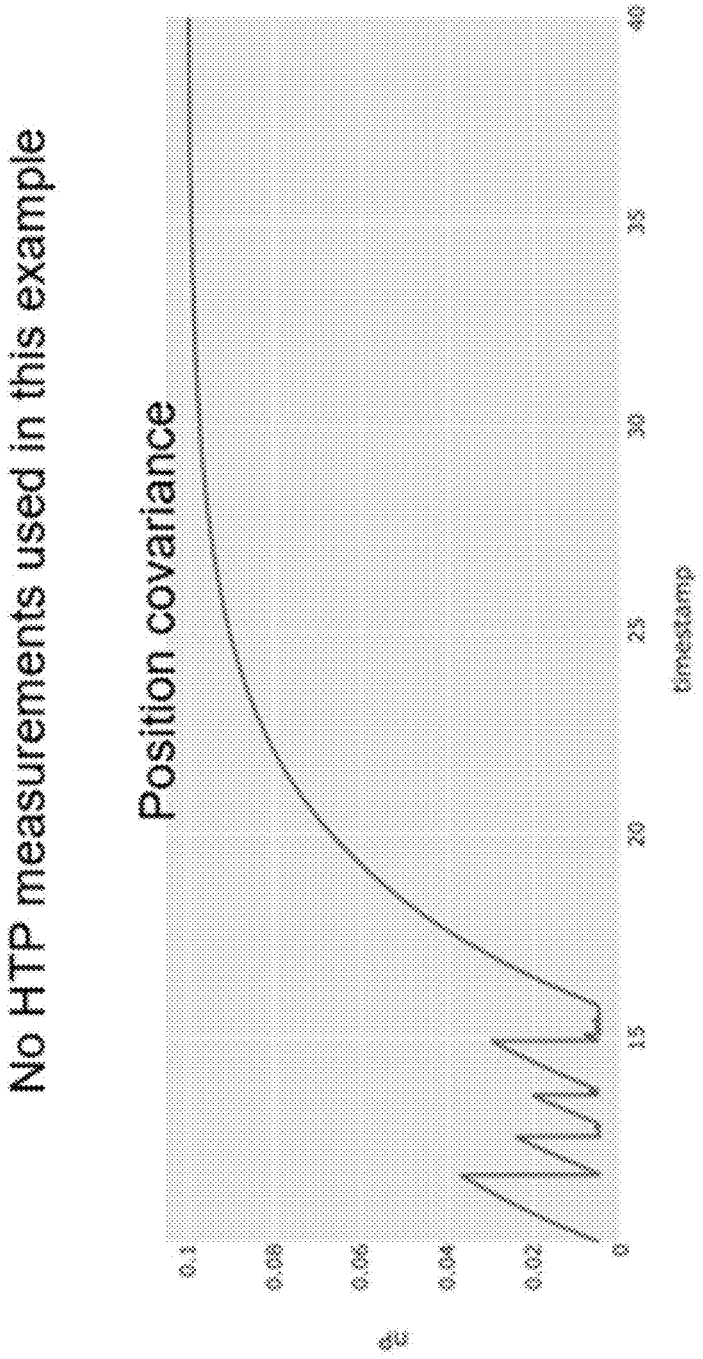
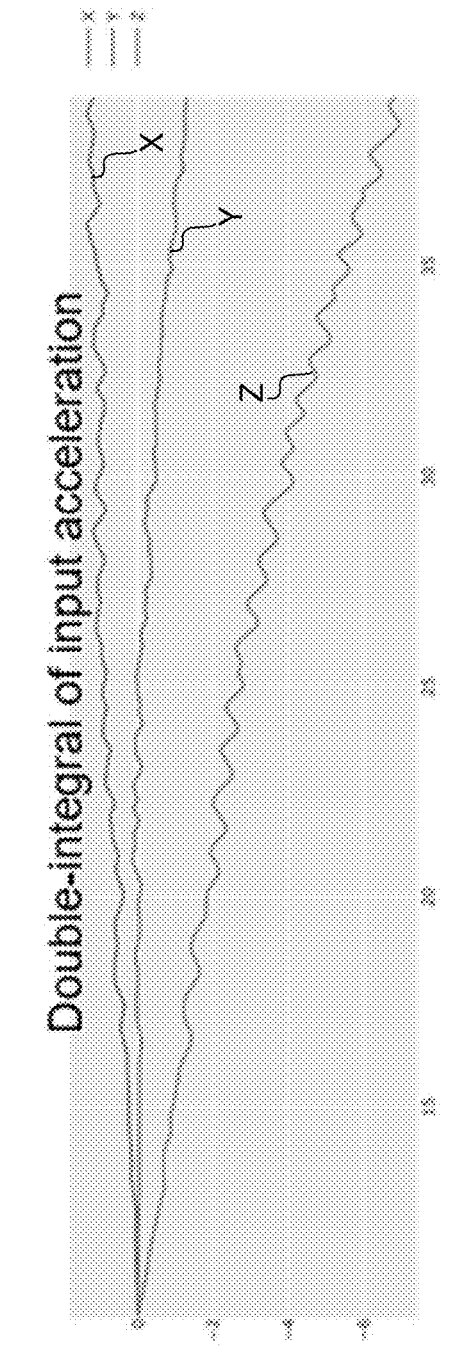

No HTP measurements used in this example
Position covariance
0.1
0.08
0.06
0.04
0.02
0
15
20
25
30
35
40
timestamp

FIG. 6



Prior Art

FIG. 7A

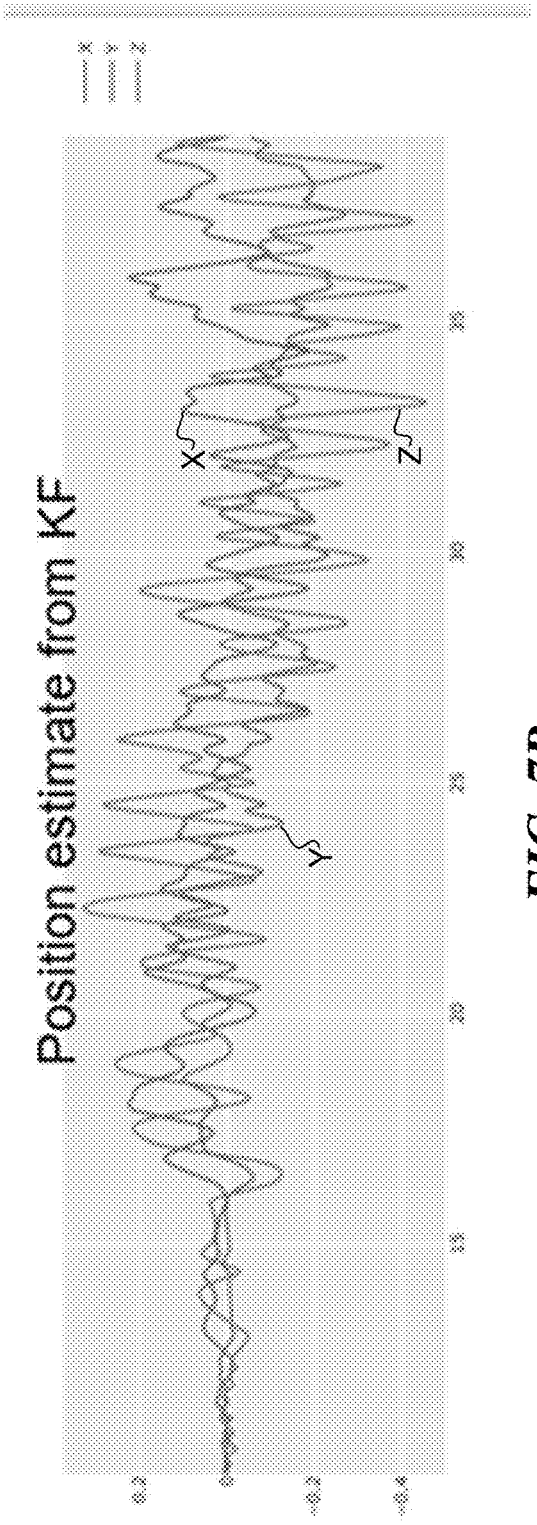


FIG. 7B

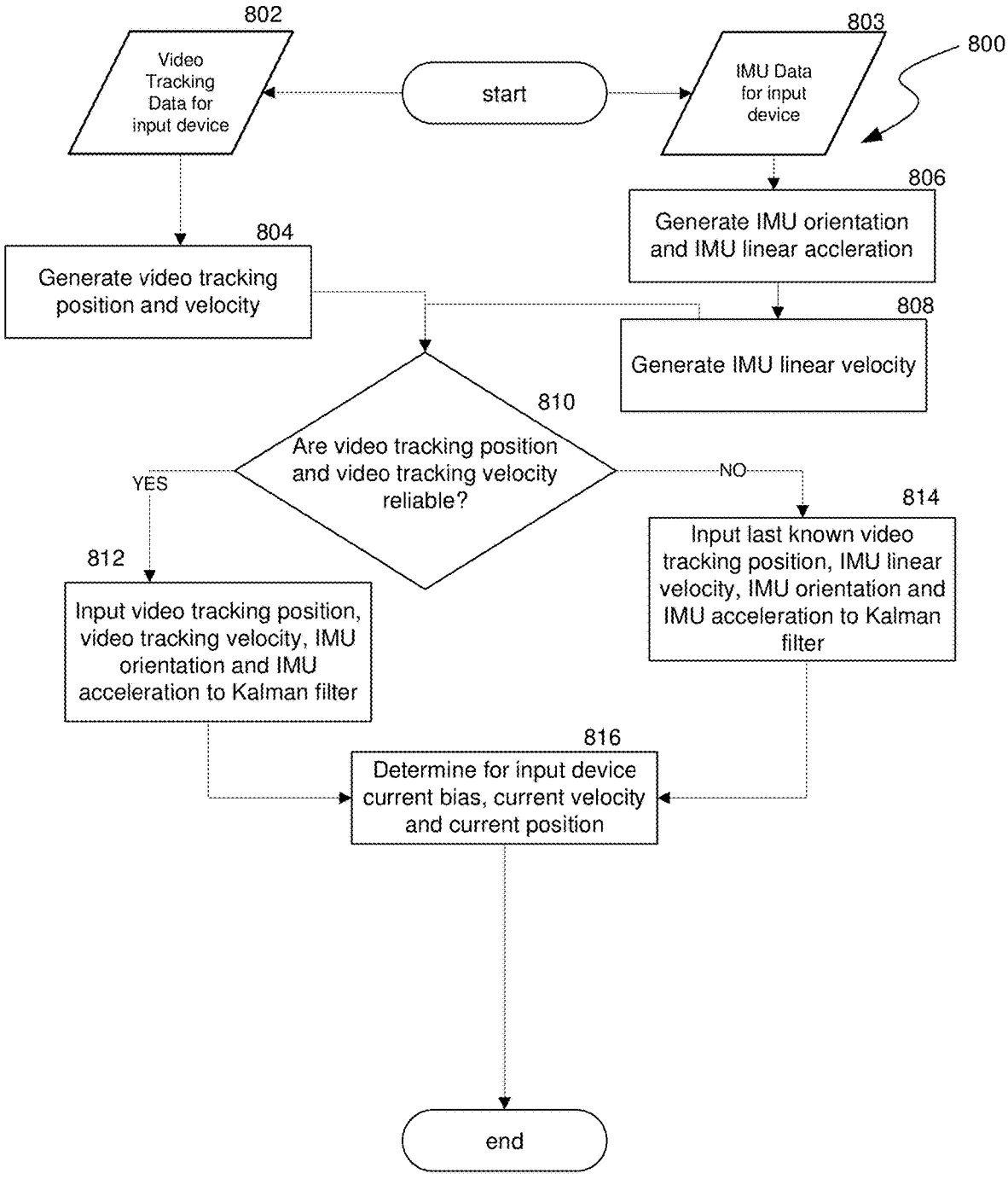

802
Video Tracking Data for input device
start
803
IMU Data for input device
800
806
Generate IMU orientation and IMU linear accleration
804
Generate video tracking position and velocity
808
Generate IMU linear velocity
810
Are video tracking position and video tracking velocity reliable?
YES
NO
812
Input video tracking position, video tracking velocity, IMU orientation and IMU acceleration to Kalman filter
814
Input last known video tracking position, IMU linear velocity, IMU orientation and IMU acceleration to Kalman filter
816
Determine for input device current bias, current velocity and current position
end

FIG. 8

MOTION-MODEL BASED TRACKING OF ARTIFICIAL REALITY INPUT DEVICES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application No. 63/499,263, filed May 15, 2023 and titled "MOTION-MODEL BASED TRACKING OF ARTIFICIAL REALITY INPUT DEVICES," which is herein incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure is directed to motion tracking of computer based input devices.

BACKGROUND

[0003] Artificial reality systems are becoming increasingly ubiquitous with applications in many fields such as computer gaming, health and safety, industrial, and education. As a few examples, artificial reality systems are being incorporated into mobile devices, gaming consoles, personal computers, movie theaters, and theme parks. In general, artificial reality is a form of reality that has been adjusted in some manner before presentation to a user, which may include, e.g., a virtual reality (VR), an augmented reality (AR), a mixed reality (MR), a hybrid reality, or some combination and/or derivatives thereof.

[0004] Typical artificial reality systems include one or more devices for rendering and displaying content to users. As one example, an artificial reality system may incorporate a head mounted display (HMD) worn by a user and configured to output artificial reality content to the user. The artificial reality content may include completely-generated content or generated content combined with captured content (e.g., real-world video and/or images). During operation, the user may utilize a hand-held controller or other input device to interact with applications or interact with the artificial reality system. Aspects of graphical elements within the artificial reality content may be determined based on the position and orientation of the input device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. **1** is a block diagram illustrating an overview of devices on which some implementations of the present technology can operate.

[0006] FIG. **2**A is a wire diagram illustrating a virtual reality headset which can be used in some implementations of the present technology.

[0007] FIG. **2**B is a wire diagram illustrating a mixed reality headset which can be used in some implementations of the present technology.

[0008] FIG. **2**C is a wire diagram illustrating controllers which, in some implementations, a user can hold in one or both hands to interact with an artificial reality environment.

[0009] FIG. **3** is a block diagram illustrating an overview of an environment in which some implementations of the present technology can operate.

[0010] FIG. **4** is a block diagram illustrating components which, in some implementations, can be used in a system employing the disclosed technology.

[0011] FIG. **5** is a block diagram of an XR input device tracking system in accordance with implementations of the disclosed technology.

[0012] FIG. **6** graphically illustrates that stiffness is the inverse of covariance.

[0013] FIG. **7**A illustrates a position estimate of an XR input device using known techniques of relying on IMU data only and a relatively simple double integral of input acceleration.

[0014] FIG. **7**B illustrates a position estimate of an XR input device using the system in accordance to implementations of the invention.

[0015] FIG. **8** is a flow diagram illustrating a process used in some implementations for tracking an XR input controller.

[0016] The techniques introduced here may be better understood by referring to the following Detailed Description in conjunction with the accompanying drawings, in which like reference numerals indicate identical or functionally similar elements.

DETAILED DESCRIPTION

[0017] Aspects of the present disclosure perform accurate position and orientation for artificial reality input devices (e.g., hand/wrist/ankle input devices), even when the devices are moved beyond the field of view of cameras in a head mounted display. Implementations use raw inertial motion unit (IMU) samples and use a physics model to integrate acceleration to get velocity and further integrate velocity to get position when the input device is "visible". Implementations further use a human body kinematics model using a Kalman filter when the input device is not visible ("occluded"), so that the position regulates to the "last known good" position, and the velocity regulates to the "rotation-inferred" velocity. Implementations, when there is direct line-of-sight between cameras and the input device, can use the Kalman filter to estimate the input device's position, where IMU samples are processed for "dead-reckoning", and a "camera-based tracking system" provides high-confidence measurements to correct dead-reckoning errors.

[0018] Implementations allow an artificial reality system to use a pure IMU or vision-based controller tracking system when the confidence value of that tracking system is high, but when the confidence value of that tracking system is lower, the system can use a last known position from that tracking system as an anchor which is modified according to IMU data, where the amount of modification is correlated (logarithmically) to the confidence factor. Thus, in addition to using cameras/video, implementations use other data based on the reality that the input device is attached to the user's hand or appendage, so there are limits to its location. When the "vision" technique is not adequate, because the input device is beyond the field of view of the head mounted display, implementations infer the position and velocity of the input device using IMU data and a model.

[0019] Implementations track an artificial reality input device by receiving, for the input device, video tracking data based on input from a camera and IMU data input from an IMU. Implementations generate, from the video tracking data, a video tracking position and a video tracking velocity, generate, from the IMU data, an IMU orientation and an IMU linear acceleration, and generate, from the IMU orientation and the IMU linear acceleration, an IMU linear velocity. Implementations determine if the video tracking position and the video tracking velocity have a confidence value above a threshold. When the video tracking position and the video tracking velocity confidence value is above the

threshold, implementations input the video tracking position, the video tracking velocity, the IMU orientation and the IMU linear acceleration to a Kalman filter. When the video tracking position and the video tracking velocity confidence value is not above the threshold, implementations input a last known video tracking position, the IMU linear velocity, the IMU orientation and the IMU linear acceleration to the Kalman filter. Implementations then determine, by the Kalman filter, for the input device, a current bias, a current velocity and a current position.

[0020] Embodiments of the disclosed technology may include or be implemented in conjunction with an artificial reality system. Artificial reality or extra reality (XR) is a form of reality that has been adjusted in some manner before presentation to a user, which may include, e.g., virtual reality (VR), augmented reality (AR), mixed reality (MR), hybrid reality, or some combination and/or derivatives thereof. Artificial reality content may include completely generated content or generated content combined with captured content (e.g., real-world photographs). The artificial reality content may include video, audio, haptic feedback, or some combination thereof, any of which may be presented in a single channel or in multiple channels (such as stereo video that produces a three-dimensional effect to the viewer). Additionally, in some embodiments, artificial reality may be associated with applications, products, accessories, services, or some combination thereof, that are, e.g., used to create content in an artificial reality and/or used in (e.g., perform activities in) an artificial reality. The artificial reality system that provides the artificial reality content may be implemented on various platforms, including a head-mounted display (HMD) connected to a host computer system, a standalone HMD, a mobile device or computing system, a "cave" environment or other projection system, or any other hardware platform capable of providing artificial reality content to one or more viewers.

[0021] "Virtual reality" or "VR," as used herein, refers to an immersive experience where a user's visual input is controlled by a computing system. "Augmented reality" or "AR" refers to systems where a user views images of the real world after they have passed through a computing system. For example, a tablet with a camera on the back can capture images of the real world and then display the images on the screen on the opposite side of the tablet from the camera. The tablet can process and adjust or "augment" the images as they pass through the system, such as by adding virtual objects. "Mixed reality" or "MR" refers to systems where light entering a user's eye is partially generated by a computing system and partially composes light reflected off objects in the real world. For example, a MR headset could be shaped as a pair of glasses with a pass-through display, which allows light from the real world to pass through a waveguide that simultaneously emits light from a projector in the MR headset, allowing the MR headset to present virtual objects intermixed with the real objects the user can see. "Artificial reality," "extra reality," or "XR," as used herein, refers to any of VR, AR, MR, or any combination or hybrid thereof.

[0022] Many current implementations of tracking XR controllers utilize embedded light-emitting diode (LED) devices (e.g., in an infrared spectrum) that can be tracked by cameras on the head mounted display or base station. However, there is a desire to reduce the amount of LEDs needed and reduce the size of the head mounted display, which would reduce the number of cameras in the display or cameras on a base station. These changes would make tracking of the XR controllers and other input devices more difficult using current techniques. Further, these "pure camera" or "vision" techniques of tracking have difficulty with blurring when the controllers are moved quickly and have difficulty when the view of the controllers are occluded.

[0023] In contrast, implementations improve the tracking of XR controllers and other XR input devices through the use of raw IMU samples and a physics model to integrate acceleration to get velocity and further integrate velocity to get position when the input device is "visible". Implementations further use a human body kinematics model using a Kalman filter when the input device is not visible/"occluded", so that the position regulates to the "last known good" position, and the velocity regulates to the "rotation-inferred" velocity.

[0024] Several implementations are discussed below in more detail in reference to the figures. FIG. **1** is a block diagram illustrating an overview of devices on which some implementations of the disclosed technology can operate. The devices can comprise hardware components of a computing system **100** that perform accurate position and orientation for XR devices, even when the devices are moved beyond the field of view of tracking cameras. In various implementations, computing system **100** can include a single computing device **103** or multiple computing devices (e.g., computing device **101**, computing device **102**, and computing device **103**) that communicate over wired or wireless channels to distribute processing and share input data. In some implementations, computing system **100** can include a stand-alone headset capable of providing a computer created or augmented experience for a user without the need for external processing or sensors. In other implementations, computing system **100** can include multiple computing devices such as a headset and a core processing component (such as a console, mobile device, or server system) where some processing operations are performed on the headset and others are offloaded to the core processing component. Example headsets are described below in relation to FIGS. **2**A and **2**B. In some implementations, position and environment data can be gathered only by sensors incorporated in the headset device, while in other implementations one or more of the non-headset computing devices can include sensor components that can track environment or position data.

[0025] Computing system **100** can include one or more processor(s) **110** (e.g., central processing units (CPUs), graphical processing units (GPUs), holographic processing units (HPUs), etc.) Processors **110** can be a single processing unit or multiple processing units in a device or distributed across multiple devices (e.g., distributed across two or more of computing devices **101**-**103**).

[0026] Computing system **100** can include one or more input devices **120** that provide input to the processors **110**, notifying them of actions. The actions can be mediated by a hardware controller that interprets the signals received from the input device and communicates the information to the processors **110** using a communication protocol. Each input device **120** can include, for example, a mouse, a keyboard, a touchscreen, a touchpad, a wearable input device (e.g., a haptics glove, a bracelet, a ring, an earring, a necklace, a

watch, etc.), a camera (or other light-based input device, e.g., an infrared sensor), a microphone, or other user input devices.

[0027] Processors **110** can be coupled to other hardware devices, for example, with the use of an internal or external bus, such as a PCI bus, SCSI bus, or wireless connection. The processors **110** can communicate with a hardware controller for devices, such as for a display **130**. Display **130** can be used to display text and graphics. In some implementations, display **130** includes the input device as part of the display, such as when the input device is a touchscreen or is equipped with an eye direction monitoring system. In some implementations, the display is separate from the input device. Examples of display devices are: an LCD display screen, an LED display screen, a projected, holographic, or augmented reality display (such as a heads-up display device or a head-mounted device), and so on. Other I/O devices **140** can also be coupled to the processor, such as a network chip or card, video chip or card, audio chip or card, USB, firewire or other external device, camera, printer, speakers, CD-ROM drive, DVD drive, disk drive, etc.

[0028] In some implementations, input from the I/O devices **140**, such as cameras, depth sensors, IMU sensor, GPS units, LiDAR or other time-of-flights sensors, etc. can be used by the computing system **100** to identify and map the physical environment of the user while tracking the user's location within that environment. This simultaneous localization and mapping (SLAM) system can generate maps (e.g., topologies, girds, etc.) for an area (which may be a room, building, outdoor space, etc.) and/or obtain maps previously generated by computing system **100** or another computing system that had mapped the area. The SLAM system can track the user within the area based on factors such as GPS data, matching identified objects and structures to mapped objects and structures, monitoring acceleration and other position changes, etc.

[0029] Computing system **100** can include a communication device capable of communicating wirelessly or wire-based with other local computing devices or a network node. The communication device can communicate with another device or a server through a network using, for example, TCP/IP protocols. Computing system **100** can utilize the communication device to distribute operations across multiple network devices.

[0030] The processors **110** can have access to a memory **150**, which can be contained on one of the computing devices of computing system **100** or can be distributed across of the multiple computing devices of computing system **100** or other external devices. A memory includes one or more hardware devices for volatile or non-volatile storage, and can include both read-only and writable memory. For example, a memory can include one or more of random access memory (RAM), various caches, CPU registers, read-only memory (ROM), and writable non-volatile memory, such as flash memory, hard drives, floppy disks, CDs, DVDs, magnetic storage devices, tape drives, and so forth. A memory is not a propagating signal divorced from underlying hardware; a memory is thus non-transitory. Memory **150** can include program memory **160** that stores programs and software, such as an operating system **162**, motion-model based tracking system **164**, and other application programs **166**. Memory **150** can also include data memory **170** that can include, e.g., data used for tracking input devices, such as previous locations, configuration data, settings, user options or preferences, etc., which can be provided to the program memory **160** or any element of the computing system **100**.

[0031] Some implementations can be operational with numerous other computing system environments or configurations. Examples of computing systems, environments, and/or configurations that may be suitable for use with the technology include, but are not limited to, XR headsets, personal computers, server computers, handheld or laptop devices, cellular telephones, wearable electronics, gaming consoles, tablet devices, multiprocessor systems, microprocessor-based systems, set-top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, or the like.

[0032] FIG. **2**A is a wire diagram of a virtual reality head-mounted display (HMD) **200**, in accordance with some embodiments. The HMD **200** includes a front rigid body **205** and a band **210**. The front rigid body **205** includes one or more electronic display elements of an electronic display **245**, an inertial motion unit (IMU) **215**, one or more position sensors **220**, locators **225**, and one or more compute units **230**. The position sensors **220**, the IMU **215**, and compute units **230** may be internal to the HMD **200** and may not be visible to the user. In various implementations, the IMU **215**, position sensors **220**, and locators **225** can track movement and location of the HMD **200** in the real world and in an artificial reality environment in three degrees of freedom (3DoF) or six degrees of freedom (6DoF). For example, the locators **225** can emit infrared light beams which create light points on real objects around the HMD **200**. As another example, the IMU **215** can include e.g., one or more accelerometers, gyroscopes, magnetometers, other non-camera-based position, force, or orientation sensors, or combinations thereof. One or more cameras (not shown) integrated with the HMD **200** can detect the light points. Compute units **230** in the HMD **200** can use the detected light points to extrapolate position and movement of the HMD **200** as well as to identify the shape and position of the real objects surrounding the HMD **200**.

[0033] The electronic display **245** can be integrated with the front rigid body **205** and can provide image light to a user as dictated by the compute units **230**. In various embodiments, the electronic display **245** can be a single electronic display or multiple electronic displays (e.g., a display for each user eye). Examples of the electronic display **245** include: a liquid crystal display (LCD), an organic light-emitting diode (OLED) display, an active-matrix organic light-emitting diode display (AMOLED), a display including one or more quantum dot light-emitting diode (QOLED) sub-pixels, a projector unit (e.g., microLED, LASER, etc.), some other display, or some combination thereof.

[0034] In some implementations, the HMD **200** can be coupled to a core processing component such as a personal computer (PC) (not shown) and/or one or more external sensors (not shown). The external sensors can monitor the HMD **200** (e.g., via light emitted from the HMD **200**) which the PC can use, in combination with output from the IMU **215** and position sensors **220**, to determine the location and movement of the HMD **200**.

[0035] FIG. **2**B is a wire diagram of a mixed reality HMD system **250** which includes a mixed reality HMD **252** and a core processing component **254**. The mixed reality HMD **252** and the core processing component **254** can communi-

cate via a wireless connection (e.g., a 60 GHZ link) as indicated by link **256**. In other implementations, the mixed reality system **250** includes a headset only, without an external compute device or includes other wired or wireless connections between the mixed reality HMD **252** and the core processing component **254**. The mixed reality HMD **252** includes a pass-through display **258** and a frame **260**. The frame **260** can house various electronic components (not shown) such as light projectors (e.g., LASERs, LEDs, etc.), cameras, eye-tracking sensors, MEMS components, networking components, etc.

[0036] The projectors can be coupled to the pass-through display **258**, e.g., via optical elements, to display media to a user. The optical elements can include one or more waveguide assemblies, reflectors, lenses, mirrors, collimators, gratings, etc., for directing light from the projectors to a user's eye. Image data can be transmitted from the core processing component **254** via link **256** to HMD **252**. Controllers in the HMD **252** can convert the image data into light pulses from the projectors, which can be transmitted via the optical elements as output light to the user's eye. The output light can mix with light that passes through the display **258**, allowing the output light to present virtual objects that appear as if they exist in the real world.

[0037] Similarly to the HMD **200**, the HMD system **250** can also include motion and position tracking units, cameras, light sources, etc., which allow the HMD system **250** to, e.g., track itself in 3DoF or 6DoF, track portions of the user (e.g., hands, feet, head, or other body parts), map virtual objects to appear as stationary as the HMD **252** moves, and have virtual objects react to gestures and other real-world objects.

[0038] FIG. **2**C illustrates controllers **270** (including controller **276**A and **276**B), which, in some implementations, a user can hold in one or both hands to interact with an artificial reality environment presented by the HMD **200** and/or HMD **250**. The controllers **270** can be in communication with the HMDs, either directly or via an external device (e.g., core processing component **254**). The controllers can have their own IMU units, position sensors, and/or can emit further light points. The HMD **200** or **250**, external sensors, or sensors in the controllers can track these controller light points to determine the controller positions and/or orientations (e.g., to track the controllers in 3DoF or 6DoF). The compute units **230** in the HMD **200** or the core processing component **254** can use this tracking, in combination with IMU and position output, to monitor hand positions and motions of the user. The controllers can also include various buttons (e.g., buttons **272**A-F) and/or joysticks (e.g., joysticks **274**A-B), which a user can actuate to provide input and interact with objects.

[0039] In various implementations, the HMD **200** or **250** can also include additional subsystems, such as an eye tracking unit, an audio system, various network components, etc., to monitor indications of user interactions and intentions. For example, in some implementations, instead of or in addition to controllers, one or more cameras included in the HMD **200** or **250**, or from external cameras, can monitor the positions and poses of the user's hands to determine gestures and other hand and body motions. As another example, one or more light sources can illuminate either or both of the user's eyes and the HMD **200** or **250** can use eye-facing cameras to capture a reflection of this light to determine eye position (e.g., based on set of reflections around the user's cornea), modeling the user's eye and determining a gaze direction.

[0040] FIG. **3** is a block diagram illustrating an overview of an environment **300** in which some implementations of the disclosed technology can operate. Environment **300** can include one or more client computing devices **305**A-D, examples of which can include computing system **100**. In some implementations, some of the client computing devices (e.g., client computing device **305**B) can be the HMD **200** or the HMD system **250**. Client computing devices **305** can operate in a networked environment using logical connections through network **330** to one or more remote computers, such as a server computing device.

[0041] In some implementations, server **310** can be an edge server which receives client requests and coordinates fulfillment of those requests through other servers, such as servers **320**A-C. Server computing devices **310** and **320** can comprise computing systems, such as computing system **100**. Though each server computing device **310** and **320** is displayed logically as a single server, server computing devices can each be a distributed computing environment encompassing multiple computing devices located at the same or at geographically disparate physical locations.

[0042] Client computing devices **305** and server computing devices **310** and **320** can each act as a server or client to other server/client device(s). Server **310** can connect to a database **315**. Servers **320**A-C can each connect to a corresponding database **325**A-C. As discussed above, each server **310** or **320** can correspond to a group of servers, and each of these servers can share a database or can have their own database. Though databases **315** and **325** are displayed logically as single units, databases **315** and **325** can each be a distributed computing environment encompassing multiple computing devices, can be located within their corresponding server, or can be located at the same or at geographically disparate physical locations.

[0043] Network **330** can be a local area network (LAN), a wide area network (WAN), a mesh network, a hybrid network, or other wired or wireless networks. Network **330** may be the Internet or some other public or private network. Client computing devices **305** can be connected to network **330** through a network interface, such as by wired or wireless communication. While the connections between server **310** and servers **320** are shown as separate connections, these connections can be any kind of local, wide area, wired, or wireless network, including network **330** or a separate public or private network.

[0044] FIG. **4** is a block diagram illustrating components **400** which, in some implementations, can be used in a system employing the disclosed technology. Components **400** can be included in one device of computing system **100** or can be distributed across multiple of the devices of computing system **100**. The components **400** include hardware **410**, mediator **420**, and specialized components **430**. As discussed above, a system implementing the disclosed technology can use various hardware including processing units **412**, working memory **414**, input and output devices **416** (e.g., cameras, displays, IMU units, network connections, etc.), and storage memory **418**. In various implementations, storage memory **418** can be one or more of: local devices, interfaces to remote storage devices, or combinations thereof. For example, storage memory **418** can be one or more hard drives or flash drives accessible through a

system bus or can be a cloud storage provider (such as in storage **315** or **325**) or other network storage accessible via one or more communications networks. In various implementations, components **400** can be implemented in a client computing device such as client computing devices **305** or on a server computing device, such as server computing device **310** or **320**.

[0045] Mediator **420** can include components which mediate resources between hardware **410** and specialized components **430**. For example, mediator **420** can include an operating system, services, drivers, a basic input output system (BIOS), controller circuits, or other hardware or software systems.

[0046] Specialized components **430** can include software or hardware configured to perform operations for input device tracking. Specialized components **430** can include an IMU data to velocity generator **434**, video tracking and IMU data regulator **436** and a position and velocity estimator **438** (e.g., a Kalman filter), and components and APIs which can be used for providing user interfaces, transferring data, and controlling the specialized components, such as interfaces **432**. In some implementations, components **400** can be in a computing system that is distributed across multiple computing devices or can be an interface to a server-based application executing one or more of specialized components **430**. Although depicted as separate components, specialized components **430** may be logical or other nonphysical differentiations of functions and/or may be submodules or code-blocks of one or more applications.

[0047] IMU data to velocity generator **434** can estimate/generate linear velocity from the IMU data corresponding to the input device. The linear velocity is estimated relative to the instantaneous rotation center of the IMU. Additional details on generating velocity data from IMU data are provided below in relation to blocks **806** and **808** of FIG. **8**.

[0048] Video tracking and IMU data regulator **436** can determine what position data and velocity data corresponding to the input device is provided to a Kalman filter based on a determination on the reliability of the video tracking data. The reliability of the video tracking data can be based on whether the input device is within view of HMD cameras. Additional details on regulating video tracking and IMU data are provided below in relation to blocks **810**, **812** and **814** of FIG. **8**.

[0049] Position and velocity estimator **438** can determine for the input device its current bias, current velocity and current position. Position and velocity estimator **438** can be implemented by a Kalman filter. Additional details on estimating position and velocity are provided below in relation to block **816** of FIG. **8**.

[0050] Those skilled in the art will appreciate that the components illustrated in FIGS. **1-4** described above, and in each of the flow diagrams discussed below, may be altered in a variety of ways. For example, the order of the logic may be rearranged, substeps may be performed in parallel, illustrated logic may be omitted, other logic may be included, etc. In some implementations, one or more of the components described above can execute one or more of the processes described below.

[0051] FIG. **5** is a block diagram of an XR input device tracking system **500** in accordance with implementations of the disclosed technology. System **500** relies on two sources of input data. Video tracking **502** is any external source of tracking data, which includes camera based vision, including the cameras on the HMD, or LEDs on the controller, or any combination of cameras and LEDs. Video tracking **502** can be implemented by a hybrid tracking pipeline ("HTP") or any other type of external positional tracking system. IMU **504** is the IMU data from the XR controller or other XR input device. IMU **504** data is used to infer/model the velocity and position of the input device when video tracking data cannot be used because, for example, the input device is no longer trackable in the field of view ("FOV") of the HMD (e.g., due to an occluded view of the input device).

[0052] The modeling implemented by system **500** assumes that any smooth trajectory can be approximated by arcs and straight lines, that human arm movement is mostly arcs, and that users prefer an algorithm that approximately tracks their movement over one that fails to track.

[0053] System **500** infers linear velocity from rotational dynamics using a 3 degrees of freedom ("DoF") filter **506** (also referred to as an "orientation filter" or "attitude filter"), a Savitzky-Golay filter **508** and a rotational dynamics module **510**. The 3 DoF filter **506** infers the orientation **552** and the gravity-removed acceleration **553** from the IMU data. The IMU data **504** includes the linear acceleration and angular velocity. The output of rotational dynamics module **510** is the linear velocity **551**, using the assumption that the movement is mostly arcs. However, this assumption results in inaccuracies, so a Kalman filter **514** is implemented for smoothing purposes since the linear velocity **551**, by itself, includes errors. Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, including statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.

[0054] Savitzky-Golay filter **508** separates/filters user intentional movement from other noise and jerks. Filter **508** performs smoothing and differentiation filtering on a sliding window. In some implementations, the window size is 100 ms, which introduces fixed latency of ½ the window size (e.g., 50 ms). Filter **508** extracts human movement: 98% human motion information <10 Hz in FFT spectrum. For smoothing, filter **508** assumes a 4th order polynomial for generalized position, a 3rd order for velocity and a 2nd order for acceleration, which is used to determine linear acceleration and angular velocity. For differentiation, filter **508** computes derivatives by differentiating corresponding polynomials, which is used for angular acceleration. Savitzky-Golay filter **508** can also output a "not moving" signal **590** when it is determined, from the IMU data **504**, that the controller is not moving.

[0055] Input to Kalman filter **514** is the device orientation **552** and linear acceleration **553** (both inferred from IMU **504** by filter **506**) and the linear velocity **551** generated by rotational dynamics module **510**. The output estimated by Kalman filter **514** is a linear position **520**, a linear velocity **518** and an accelerometer bias **516**.

[0056] System **500** further includes a position and velocity regulator **512**. When there are "reliable" measurements from video tracking **502**, regulator **512** passes through, to filter **514**, position **561** and velocity **562**, which are the same as estimated position **571** and estimated velocity **572** from

video tracking **502**. In this case, position **520** will generally converge, via **580**, to be the same as the position of video tracking **502**.

[0057] Regulator **512** determines whether the measurements from video tracking **502** are reliable based on a confidence level. Video tracking **502** outputs an estimated position **571** and an estimated velocity **572**, as well as a corresponding confidence level. For example, video tracking **502** can implement an internal Kalman filter, which would produce an "uncertainty" (i.e., covariance) in connection of the position and velocity, and the confidence could be defined as the inverse of that uncertainty (i.e., higher uncertainty relates to lower confidence, and vice versa). Example implementations further include additional "sanity check" video tracking measurements, in case the confidence levels it produces are not reliable themselves (e.g., Kalman filter **514** becoming "over confident"). One example is to check the Mahalanobis distances between the video tracking estimates and velocity **518** and position **520**, and reject incoming measurements if the Mahalanobis distances exceeds pre-defined thresholds.

[0058] Rotational dynamics module **510** only outputs velocity **551** in example implementations. When video tracking **502** data is reliable (e.g., confidence level greater than a pre-defined threshold), regulator **512** uses video tracking estimated position **571** and estimated velocity **572** as input to filter **514**. When video tracking **502** data is not reliable, regulator **512** uses the last known position from video tracking **502**, since only velocity and not position is output from module **510**, as an anchor, but with low confidence and uses a "spring" effect to always return to the last known position using anchoring with some stiffness of the spring, represented by the "covariance" or the "confidence level." FIG. **6** graphically illustrates that stiffness is the inverse of covariance. The higher the covariance, the less confident about the position. The "spring" allows for movement, but prevents a "flying away" indefinitely from the last known position.

[0059] In general, regulator **512** operates to output target position **561**, target velocity **562**, and corresponding covariance by applying the following rules:

[0060] Target position update rule—

[0061] When Video tracking is tracking: snap to the video tracking position

[0062] When Device becomes stationary: anchor at the current Kalman filter position estimate

[0063] Target position covariance update rule—

[0064] When Video tracking is tracking: snap to the video tracking covariance

[0065] When Device is stationary: snap to predefined value

[0066] When Device is moving: use exponential ramp until hitting maximum covariance

[0067] Target velocity and covariance update rule—

[0068] When Video tracking is tracking: snap to the video tracking velocity and covariance

[0069] When Device is stationary: use zero-velocity with predefined covariance

[0070] When Device is moving: use rotation-inferred velocity with predefined covariance

[0071] In implementations, regulator **512** further provides:

[0072] Small covariance when stationary—

[0073] Users may see constant offset but not drift

[0074] Anchors to Kalman filter position when becoming stationary

[0075] Cannot recover from accumulated error

[0076] Similar behavior to: marginalization of constraint graphs

[0077] Growing covariance when starts to move.

[0078] Users may feel movement is reduced

[0079] Good for "fine adjustment", like mouse cursor

[0080] Large covariance if moving continuously

[0081] Users feel movement is related but may observe overshoot/undershoot and/or mild drift

[0082] Very gentle pull to the anchor position→suppresses significant drift

[0083] Stronger pull→no long-term drift, but some short-term drift

[0084] FIG. **7**A illustrates a position estimate of an XR input device using known techniques of relying on IMU data only and a relatively simple double integral of input acceleration. As shown, over time on the x-axis, the x, y and z positions rapidly drift/diverge/flyaway, indicating inaccuracy of the estimated position.

[0085] In contrast, FIG. **7**B illustrates a position estimate of an XR input device using system **500** in accordance to implementations of the invention. As shown, the x, y and z positions do not drift over time as with the known solutions, indicating a more accurate position estimate.

[0086] Although example implementations can use an XR controller, such as controller **270**, the functionality can track other XR input devices that include an IMU and are anchored to the user (e.g., a smart watch, an ankle bracelet, etc.). Example implementations can also be an IMU only tracking system, without HTP, as long as there is also an external positional tracking system to estimate position and optionally velocity.

[0087] FIG. **8** is a flow diagram illustrating a process used in some implementations for tracking an XR input controller.

[0088] Process **800** receives HTP data at **802** (e.g., video tracking **502**) and IMU data at **803** (e.g., IMU **504**). The HTP data is any external source of tracking data, which includes camera based vision, including the cameras on the HMD, or LEDs on the controller, or any combination of cameras and LEDs. The IMU data is generated from the IMU in the controller or other input device.

[0089] At block **804**, from the video tracking data, a video tracking estimated position **571** and video tracking estimated velocity **572** is generated. The video tracking data is a tracking of the movement of the input device, which directly provides the velocity and position.

[0090] At block **806**, from the IMU data, an IMU orientation **552** and IMU linear acceleration **553** is generated. A 3 DoF filter **506** infers the IMU orientation and the gravity-removed linear acceleration from the IMU data.

[0091] At block **808**, an IMU linear velocity **551** is generated from the IMU orientation and the IMU linear acceleration. The IMU linear velocity is based on rotational dynamics using 3DoF filter **506**, Savitzky-Golay filter **508** and rotational dynamics module **510**. The output of rotational dynamics module **510** is the linear velocity **551**, using the assumption that the movement of the input device is mostly arcs.

[0092] At block **810**, it is determined if the video tracking data (i.e., video tracking estimated position **571** and video tracking estimated velocity **572**) is reliable. For example, it is determined if the video tracking data has a confidence value above a threshold. The confidence value is related to a covariance, and is small when the input device is stationary, growing when the input device starts to move, and large if the input device is moving continuously.

[0093] If yes at block **810**, at block **812** the video tracking position **571**, video tracking velocity **572**, IMU orientation **552** and IMU linear acceleration **553** are input to Kalman filter **514**.

[0094] If no at block **810**, at block **814** the last known video tracking position **520**, IMU linear velocity **551**, IMU orientation **552** and IMU linear acceleration **553** are input to Kalman filter **514**.

[0095] At block **816**, Kalman filter **514** determines the current bias **516**, velocity **518** and position **520** of the input device, which provides the current tracking of the input device.

[0096] Reference in this specification to "implementations" (e.g., "some implementations," "various implementations," "one implementation," "an implementation," etc.) means that a particular feature, structure, or characteristic described in connection with the implementation is included in at least one implementation of the disclosure. The appearances of these phrases in various places in the specification are not necessarily all referring to the same implementation, nor are separate or alternative implementations mutually exclusive of other implementations. Moreover, various features are described which may be exhibited by some implementations and not by others. Similarly, various requirements are described which may be requirements for some implementations but not for other implementations.

[0097] As used herein, being above a threshold means that a value for an item under comparison is above a specified other value, that an item under comparison is among a certain specified number of items with the largest value, or that an item under comparison has a value within a specified top percentage value. As used herein, being below a threshold means that a value for an item under comparison is below a specified other value, that an item under comparison is among a certain specified number of items with the smallest value, or that an item under comparison has a value within a specified bottom percentage value. As used herein, being within a threshold means that a value for an item under comparison is between two specified other values, that an item under comparison is among a middle-specified number of items, or that an item under comparison has a value within a middle-specified percentage range. Relative terms, such as high or unimportant, when not otherwise defined, can be understood as assigning a value and determining how that value compares to an established threshold. For example, the phrase "selecting a fast connection" can be understood to mean selecting a connection that has a value assigned corresponding to its connection speed that is above a threshold.

[0098] As used herein, the word "or" refers to any possible permutation of a set of items. For example, the phrase "A, B, or C" refers to at least one of A, B, C, or any combination thereof, such as any of: A; B; C; A and B; A and C; B and C; A, B, and C; or multiple of any item such as A and A; B, B, and C; A, A, B, C, and C; etc.

[0099] Although the subject matter 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 specific features or acts described above. Specific embodiments and implementations have been described herein for purposes of illustration, but various modifications can be made without deviating from the scope of the embodiments and implementations. The specific features and acts described above are disclosed as example forms of implementing the claims that follow. Accordingly, the embodiments and implementations are not limited except as by the appended claims.

[0100] Any patents, patent applications, and other references noted above are incorporated herein by reference. Aspects can be modified, if necessary, to employ the systems, functions, and concepts of the various references described above to provide yet further implementations. If statements or subject matter in a document incorporated by reference conflicts with statements or subject matter of this application, then this application shall control.

I/We claim:

1. A method for tracking an artificial reality input device, the method comprising:

receiving, for the input device, video tracking data and inertial motion unit (IMU) data based on motion input;

generating, from the video tracking data, a video tracking position and a video tracking velocity;

generating, from the IMU data, an IMU orientation and an IMU linear acceleration;

generating, from the IMU orientation and the IMU linear acceleration, an IMU linear velocity;

determining that the video tracking position and the video tracking velocity is below a reliability threshold;

in response to the determining, inputting a last known video tracking position, the IMU linear velocity, the IMU orientation and the IMU linear acceleration to a Kalman filter; and

determining, by the Kalman filter and for the input device, a current bias, a current velocity and a current position for the input device.

2. The method of claim **1**, further comprising:

determining that the video tracking position and the video tracking velocity has risen above the reliability threshold; and

in response to the determining that the video tracking position and the video tracking velocity has risen above the reliability threshold, inputting the video tracking position, the video tracking velocity, the IMU orientation and the IMU linear acceleration to the Kalman filter.

3. The method of claim **1**, wherein the determining that the video tracking position and the video tracking velocity is below the reliability threshold comprises determining that a confidence level, generated by a second Kalman filter, is below the threshold.

4. The method of claim **3**, further comprising evaluating the confidence level based on a Mahalanobis distance.

5. The method of claim **1**, wherein the video tracking data is generated based on light, from a light-emitting diode, being received by a camera.

6. The method of claim 1, wherein the IMU linear acceleration is generated from a rotational dynamics module that assumes movement of the input device comprises mostly arcs.

7. The method of claim 1, further comprising determining, using a Savitzky-Golay filter, that the input device is being intentionally moved by a user.

8. The method of claim 1, wherein the generating, from the IMU data, the IMU orientation and the IMU linear acceleration comprises using a three degrees of freedom filter.

9. The method of claim 1, wherein the determining that the video tracking position and the video tracking velocity is below the reliability threshold is based on the input device being occluded from view of a camera.

10. A computer-readable storage medium storing instructions that, when executed by a computing system, cause the computing system to perform a process for tracking an artificial reality input device, the process comprising:

receiving, for the input device, video tracking data and inertial motion unit (IMU) data based on motion input;

generating, from the video tracking data, a video tracking position and a video tracking velocity;

generating, from the IMU data, an IMU orientation and an IMU linear acceleration;

generating, from the IMU orientation and the IMU linear acceleration, an IMU linear velocity;

determining that the video tracking position and the video tracking velocity is below a reliability threshold;

in response to the determining, inputting a last known video tracking position, the IMU linear velocity, the IMU orientation and the IMU linear acceleration to a Kalman filter; and

determining, by the Kalman filter and for the input device, a current bias, a current velocity and a current position for the input device.

11. The computer-readable storage medium of claim 10, wherein the process comprises:

determining that the video tracking position and the video tracking velocity has risen above the reliability threshold; and

in response to the determining that the video tracking position and the video tracking velocity has risen above the reliability threshold, inputting the video tracking position, the video tracking velocity, the IMU orientation and the IMU linear acceleration to the Kalman filter.

12. The computer-readable storage medium of claim 10, wherein the determining that the video tracking position and the video tracking velocity is below the reliability threshold comprises determining that a confidence level, generated by a second Kalman filter, is below the threshold.

13. The computer-readable storage medium of claim 12, wherein the process further comprises evaluating the confidence level based on a Mahalanobis distance.

14. The computer-readable storage medium of claim 10, wherein the video tracking data is generated based on light, from a light-emitting diode, being received by a camera.

15. The computer-readable storage medium of claim 10, wherein the IMU linear acceleration is generated from a rotational dynamics module that assumes movement of the input device comprises mostly arcs.

16. The computer-readable storage medium of claim 10, wherein the process further comprises determining, using a Savitzky-Golay filter, that the input device is being intentionally moved by a user.

17. The computer-readable storage medium of claim 10, wherein the generating, from the IMU data, the IMU orientation and the IMU linear acceleration comprises using a three degrees of freedom filter.

18. The computer-readable storage medium of claim 10, wherein the determining that the video tracking position and the video tracking velocity is below the reliability threshold is based on the input device being occluded from view of a camera.

19. A computing system for tracking an artificial reality input device, the computing system comprising:

one or more processors; and

one or more memories storing instructions that, when executed by the one or more processors, cause the computing system to perform a process comprising:

receiving, for the input device, video tracking data and inertial motion unit (IMU) data based on motion input;

generating, from the video tracking data, a video tracking position and a video tracking velocity;

generating, from the IMU data, an IMU orientation and an IMU linear acceleration;

generating, from the IMU orientation and the IMU linear acceleration, an IMU linear velocity;

determining that the video tracking position and the video tracking velocity is below a reliability threshold;

in response to the determining, inputting a last known video tracking position, the IMU linear velocity, the IMU orientation and the IMU linear acceleration to a Kalman filter; and

determining, by the Kalman filter and for the input device, a current bias, a current velocity and a current position for the input device.

20. The computing system of claim 19, wherein the process further comprises:

determining that the video tracking position and the video tracking velocity has risen above the reliability threshold; and

in response to the determining that the video tracking position and the video tracking velocity has risen above the reliability threshold, inputting the video tracking position, the video tracking velocity, the IMU orientation and the IMU linear acceleration to the Kalman filter.

* * * * *