(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
24 December 2020 (24.12.2020)

(10) **International Publication Number**
WO 2020/252575 A1

(51) **International Patent Classification:**
G01C 21/16 (2006.01) ***G08G 1/16*** (2006.01)
B60W 30/09 (2012.01) ***H04W 4/40*** (2018.01)

(21) **International Application Number:**
PCT/CA2020/050838

(22) **International Filing Date:**
17 June 2020 (17.06.2020)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
62/862,645 17 June 2019 (17.06.2019) US

(72) **Inventor; and**
(71) **Applicant: SETH, Rohit** [CA/CA]; 16282 Side Road 27, Terra Cotta, Ontario L0P 1N0 (CA).

(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, 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,

(54) Title: RELATIVE POSITION TRACKING USING MOTION SENSOR WITH DRIFT CORRECTION

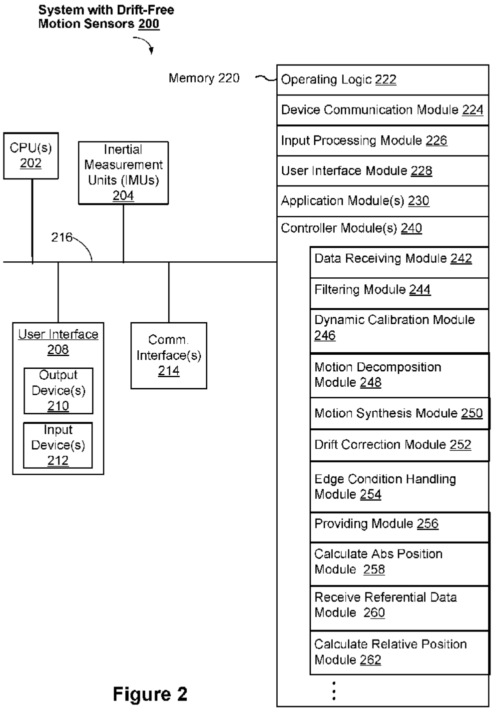


Figure 2

(57) **Abstract:** A method is provided for calculating a position and/or orientation of a first object relative to a second object. The method includes receiving a first object initial absolute position. The method includes sensing, using a first IMUs, motion of the first object and generating sensed motion data of the first object. The method includes generating, using the controller, a motion signal representative of the motion of the first object. The method includes calculating, using the controller, a first object current absolute position using the motion signal and the first object initial absolute position. The method includes receiving, from the second object, a second object current absolute position calculated using a second IMUs associated with the second object. The method includes calculating a relative position and/or orientation of the first object relative to the second object using the first object current absolute position and the second object current absolute position.

WO 2020/252575 A1

[Continued on next page]

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 the identity of the inventor (Rule 4.17(i))*
- *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))*
- *of inventorship (Rule 4.17(iv))*

Published:

- *with international search report (Art. 21(3))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

Relative Position Tracking Using Motion Sensor with Drift Correction

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is related to U.S. Patent No. 9,417,693, entitled "Wearable Wireless HMI Device" and filed on December 8, 2015; U.S. Patent No. 9,846,482, entitled "Wearable Wireless HMI Device" and filed on August 15, 2016; U.S. Patent Application Publication No. 2018/0101231, entitled "Wearable Wireless HMI Device" and filed on December 11, 2017; and U.S. Provisional Application No. 62/690,865 filed on June 27, 2018. The entire contents of these references are incorporated by reference in their entireties.

TECHNICAL FIELD

[0002] The disclosed implementations relate generally to motion sensors and more specifically to a method, system, and device for implementing motions sensors with drift correction, in some implementations, capable of position tracking, more accurate than the Global Positioning System (GPS), and independent of external reference markers, transponders, or satellites.

BACKGROUND

[0003] Motion tracking detects the precise position and location of an object by recognizing rotation (pitch, yaw, and roll) and translational movements of the object. Inertial tracking is a type of motion tracking that uses data from sensors (e.g., accelerometers, gyroscopes, magnetometers, altimeters, and pressure sensors) mounted on an object to measure positional changes of the object. Some of the sensors are inertial sensors that rely on dead reckoning to operate. Dead reckoning is the process of calculating an object's current location by using a previously determined position, and advancing that position based upon known or estimated accelerations, speeds, or displacements over elapsed time and course. While dead reckoning techniques are somewhat effective, they are subject to cumulative error called "drift" Because some IMUs estimate relative position/location by integrating acceleration data twice from an accelerometer, even a small error in acceleration results in compounded, increasing, error in relative position/location that accumulates over time. Similarly, errors in gyroscopic angular velocity data lead to cumulative error in relative angular orientation. Thus, acceleration and gyroscopic data are unreliable, when used in isolation, to estimate orientation and positional changes of an object being tracked using IMUs.

[0004] Traditional solutions that compensate for drift without eliminating it, are too costly and/or unreliable. One such solution merges external reference data, such as from a camera or a GPS, with the data from an IMU mounted on an object being tracked to reset the drift of the IMU data at defined intervals. Fusing IMU data with a GPS signal typically results in a large error margin (e.g., several meters). Other solutions fuse data from multiple IMUs, using a Kalman filter, and weighted averaging to try and reduce orientation-drift using statistical calculations over a large sample or number of IMU sensors with differential measurement techniques. When multiple sensors are used, even if a few of the sensors are obstructed, orientation tracking becomes increasingly unreliable. Location tracking remains unsolved even with the best, current drift-compensation techniques employing multiple sensors with weighted statistical calculations. Furthermore, a very large number of sensors is required to significantly reduce overall drift in a multi-sensor system. More sensors also mean higher cost, greater combined power consumption, and increased latency in sampling and processing of orientation data. Increased latency causes low sampling rates further leading to positional error and a reduction in positional accuracy.

SUMMARY

[0005] Accordingly, there is a need for systems and/or devices for implementing cost-effective, high accuracy, high-speed motion sensors that correct for drift.

[0006] (A1) In one aspect, some implementations include a tracking device for tracking location and orientation of an object. The device comprises one or more sides that form a predetermined shape. The device also comprises a plurality of inertial measurement units (IMU) mounted to the one or more sides of the predetermined shape. Each IMU is configured to detect movement of the object and generate inertial output data representing non-linear acceleration and/or angular velocity of the object. Each IMU includes a first sub-sensor and a second sub-sensor. Each IMU is positioned at a predetermined distance and orientation relative to each other and a center of mass of the tracking device. The device also comprises a controller communicatively coupled to the plurality of IMUs, the controller configured to perform a sequence of steps. The sequence of steps comprises receiving first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs. The sequence of steps also comprises for each IMU: generating calibrated inertial output data based on the first sub-sensor inertial output data and the second sub-sensor inertial output data; and, cross-correlating the first sub-sensor inertial output data with the second sub-sensor inertial output data to identify and remove anomalies from the first sub-sensor inertial output data with the second sub-sensor inertial output data to generate decomposed inertial output data. The sequence of steps also comprises determining the translational and rotational state of the tracking device based on the decomposed inertial output data from each of the IMUs. The sequence of steps also comprises synthesizing first sub-sensor inertial output data and second sub-sensor inertial output data to create IMU synthesized or computed data using a synthesizing methodology based on the translational and rotational state of the tracking device. The sequence of steps also comprises calculating a current tracking device rectified data output (also referred to herein as “drift-free or “drift-corrected”) based on the synthesized movement of each of the IMUs, a predetermined position of each of the IMUs and a predetermined orientation of each of the IMUs. The sequence of steps also comprises calculating a current location and orientation of an object based on a difference between the current object rectified data output, and a previous object drift-free or rectified data output.

[0007] (A2) In some implementations of the tracking device of A1, wherein generating calibrated inertial output data includes applying neural network weights to the first sub-sensor inertial output data and the second sub-sensor inertial output data, wherein the neural network weights are adjusted at a learning rate based on the positional state of the tracking device, calculating a discrepancy value representative of a difference between an actual movement of the object and estimated movement of the object, and removing the discrepancy value from the calibrated inertial output data.

[0008] (A3) In some implementations of the tracking device of A2, the neural network weights applied to the first sub-sensor inertial output data and the second inertial output data are based on historical inertial output data from each of the first and second sub-sensors.

[0009] (A4) In some implementations of the tracking device of A1, the decomposed inertial output data corresponding to the first sub-sensor is calibrated based on the second sub-sensor inertial output data by providing feedback to dynamic-calibration neural network of first sub-sensor.

[0010] (A5) In some implementations of the tracking device of A1, cross-correlating the first sub-sensor inertial output data with the second sub-sensor inertial output data includes applying pattern recognition to the second sub-sensor inertial output data to generate a decomposed inertial output data representative of the first sub-sensor inertial output data.

[0011] (A6) In some implementations of the tracking device of A1, the first sub-sensor inertial output data and second sub-sensor inertial output data are filtered to minimize signal noise through signal conditioning.

[0012] (A7) In some implementations of the tracking device of A1, the first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs is received periodically at less than approximately 1 millisecond (ms) for continuous high sampling rate.

[0013] (A8) In some implementations of the tracking device of A1, the first sub-sensor and the second sub-sensor are each one of: accelerometer, magnetometer, gyroscope, altimeter, and pressure sensor; wherein the first sub-sensor is a different sensor type than the second sub-sensor.

[0014] (A9) In some implementations of the tracking device of A1, the predetermined shape is one of: a plane, a tetrahedron, a cube, or any platonic solid, or any other irregular configurations with known distances and angels between IMUs.

[0015] (A10) In some implementations of the tracking device of A1, at least some of the IMUs used to calculate the rectified IMU data output are oriented at different angles along two different axes relative to each other.

[0016] (A11) In some implementations of the tracking device of A1, calculating the current position and orientation of the object based on the difference between the current rectified IMU output and the previous object rectified IMU output include: identifying an edge condition; and blending the current object rectified IMU output and the previous object rectified IMU output to remove the edge condition using neural networks

[0017] (A12) In another aspect, some implementations include a method of tracking the location and orientation of an object using a tracking device. The tracking device includes one or more sides that define a predetermined shape. The tracking device also includes a plurality of inertial measurement units (IMU) mounted to the one or more sides of the predetermined shape. Each IMU includes a first sub-sensor and a second sub-sensor. Each IMU is positioned at a predetermined distance and orientation relative to each other and a center of mass of the tracking device. The tracking device also includes a controller communicatively coupled to the plurality of IMUs. The method comprises performing a sequence of steps. The sequence of steps includes, at each IMU, detecting movement of the object and generating inertial output data representing acceleration and/or angular velocity of the object. The sequence of steps also includes, at the controller, receiving first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs. The sequence of steps also includes, at the controller, for each IMU: generating calibrated inertial output data based on the first sub-sensor inertial output data and the second sub-sensor inertial output data; cross-correlating the first sub-sensor inertial output data with the second sub-sensor inertial output data to identify and remove anomalies from the first sub-sensor inertial output data with the second sub-sensor inertial output data to generate decomposed inertial output data. The sequence of steps also includes, at the controller, determining a translational and rotational state of the tracking device based on the decomposed inertial output data from each of the IMUs. The sequence of steps also includes, at the controller, synthesizing first sub-sensor inertial output data and second sub-sensor inertial output data to create IMU synthesized or computed data using a synthesizing methodology based on the positional and rotational state of the tracking device. The sequence of steps also includes, at the controller, calculating a current tracking device overall drift-free or rectified data output based on the synthesized movement of each of the IMUs, a predetermined location of each of the IMUs and a predetermined orientation of each of the IMUs. The sequence of steps also includes, at the controller, calculating a current location and orientation of an object based on a difference between the current object overall rectified data and a previous object overall rectified data.

[0018] (A13) In some implementations of the method of (A12), wherein generating calibrated inertial output data includes applying neural network weights to the first sub-sensor inertial output data and the second sub-sensor inertial output data, wherein the neural network weights are adjusted at a learning rate based on the positional state of the tracking device, calculating a discrepancy value representative of a difference between an actual movement of the object and estimated movement of the object, and removing the discrepancy value from the calibrated inertial output data.

[0019] (A14) In some implementations of the method of (A13), the neural network weights applied to the first sub-sensor inertial output data and the second inertial output data are based on historical inertial output data from each of the first and second sub-sensors.

[0020] (A15) In some implementations of the method of (A12), the decomposed inertial output data corresponding to the first sub-sensor is calibrated based on the second sub-sensor inertial output data by providing feedback to dynamic-calibration neural network of first sub-sensor.

[0021] (A16) In some implementations of the method of (A12), cross-correlating the first sub-sensor inertial output data with the second sub-sensor inertial output data includes applying pattern recognition to the second sub-sensor inertial output data to generate a decomposed inertial output data representative of the first sub-sensor inertial output data.

[0022] (A17) In some implementations of the method of (A12), the first sub-sensor inertial output data and the second sub-sensor inertial output data are filtered to minimize signal noise through signal conditioning.

[0023] (A18) In some implementations of the method of (A12), the first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs is received periodically at less than approximately 1 ms for continuous high sampling rate.

[0024] (A19) In some implementations of the method of (A12), the first sub-sensor and the second sub-sensor are each one of: accelerometer, magnetometer, gyroscope, altimeter, and pressure sensor and the first sub-sensor is a different sensor type than the second sub-sensor.

[0025] (A20) In some implementations of the method of (A12), the predetermined shape is one of: a plane, a tetrahedron, a cube or any platonic solid, or any other irregular configurations with known distances and angels between IMUs.

[0026] (A21) In some implementations of the method of (A12), at least some of the IMUs used to calculate the overall drift-free or rectified system output are oriented at different angles along two different axes relative to each other.

[0027] (A22) In some implementations of the method of (A12), calculating the current location and orientation of the object based on the difference between the current object rectified data and the previous object rectified data output include: identifying an edge condition; and blending the current object rectified data output and the previous object rectified data output to remove the edge condition using neural networks.

[0028] (A23) In another aspect, a method is provided for calculating a position of a first object relative to a second. The method is performed at a first object including a controller, a wireless transceiver, and a first plurality of inertial measurement units (IMUs) each mounted in one or more positions and orientations relative to other of the first plurality of IMUs. The first object is configured to receive a first object initial absolute position for the first plurality of IMUs and/or controller. The first object is also configured to sense, using the first plurality of IMUs, motion of the first object and generate sensed motion data of the first object. The first object is also configured to generate, using the controller, a motion signal representative of the motion of the first object, wherein the motion signal is generated by calculating a rectified data output based on sensed motion data from each of the first plurality of IMUs, a predetermined position of each of the first plurality of IMUs and a predetermined orientation of each of the first plurality of IMUs. The first object is also configured to calculate, using the controller, a first object current absolute position using the motion signal generated by the controller and the first object initial absolute position. The first object is also configured to receive, using the wireless transceiver, referential data from the second object, the referential data including a second object current absolute position calculated using a second plurality of IMUs associated with the second object. The first object is also configured to calculate a relative position of the first object relative to the second object using the first object current absolute position and the second object current absolute position, wherein the relative position includes at least one of: (i) a distance between the first object and the second object and (ii) an orientation of the first object relative to the second object.

[0029] (A24) In some implementations of the method of (A23), the referential data includes a third object current absolute position of a third object calculated using a third plurality of IMUs associated with the third object. The first object is also configured to calculate a relative position of the first object relative to the third object using the first object current absolute position and the third object current absolute position, wherein the relative position includes at least one of: (i) a distance between the first object and the third object and (ii) an orientation of the first object relative to the third object.

[0030] (A25) In some implementations of any of the methods (A23)-(A24), the first object is configured to transmit, using the wireless transceiver at the first object, the first object current absolute position of the first object to the second object. The second object is configured to: receive, using a wireless transceiver at the second object, the first object current absolute position of the first object; and calculate, using a controller at the second object, a relative position of the second object relative to the first object using the first object current absolute position and the second object current absolute position, wherein the relative position includes at least one of: (i) a distance between the second object and the first object and (ii) an orientation of the second object to the first object.

[0031] (A26) In some implementations of any of the methods (A23)-(A25), the first plurality of IMUs generates the motion signal using at least one of: shape correction, static calibration, motion decomposition, dynamic calibration, motion synthesis, and edge condition smoothing.

[0032] (A27) In some implementations of any of the methods (A23)-(A26), the first plurality of IMUs includes an accelerometer and/or a gyroscope.

[0033] (A28) In some implementations of any of the methods (A23)-(A27), the first object current absolute position and the second object current absolute position are calculated without an external reference signal.

[0034] (A29) In some implementations of any of the methods (A23)-(A28), the first object is a first car and the second object is a second car. The first object is configured to after calculating a relative position of the first car relative to the second car, determine whether the relative position of the first car to the second car meets emergency criteria. The first object is also configured to, in response to determining that the relative position of the first car to the second car meets emergency criteria, cause the first car to perform an evasive maneuver. The evasive maneuver includes braking the first car and/or turning the first car.

[0035] (A30) In some implementations of any of the methods (A23)-(A29), the first object is configured to display, at a user interface associated with the first object, a position of the first object on a graphical representation of a map using the relative position of the first object relative to the second object.

[0036] (A31) In some implementations of any of the methods (A23)-(A28), the first object is a home appliance and the second object is a car. The home appliance is configured to: after calculating a relative position of the car relative to the home appliance, determine whether the relative position of the car to the home appliance meets operational state change criteria; and, in response to determining that the relative position of the car to the home appliance meets operational state change criteria, cause the home appliance to change from an off state to an on state.

[0037] (A32) In another aspect, a system is provided for calculating a position of a first object relative to a second. The system includes a first object including a controller, a wireless transceiver, and a first plurality of inertial measurement units (IMUs). The first object is configured to perform the steps of any of the methods (A23)-(A31).

BRIEF DESCRIPTION OF THE DRAWINGS

[0038] For a better understanding of the various described implementations, reference should be made to the Description of Implementations below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0039] Figures 1A-1F illustrate various configurations of motion sensors mounted on two-dimensional ("2-D") or three-dimensional ("3-D") objects, in accordance with some implementations.

[0040] Figure 2 is a block diagram illustrating a representative system with sensor(s) with drift correction, according to some implementations.

[0041] Figure 3 is a flow diagram illustrating the flow of sensor data through a representative system with drift correction, according to some implementations.

[0042] Figures 4A-4D illustrate a flowchart representation of a method of tracking position and orientation of an object using a tracking device, according to some implementations.

[0043] Figure 5A-5D is a block diagram illustrating the method of calculating the position of an object.

[0044] Figure 6 illustrates a flowchart representation of a method of calculating the position of an object.

DESCRIPTION OF IMPLEMENTATIONS

[0045] Described herein are exemplary implementations for systems, methods and/or devices for implementing cost-effective, high accuracy, high-speed motion sensors that correct for drift. There are a number of different applications for motion sensors that correct for drift, including, but not limited to, gaming systems, smartphones, helmet-mounted displays, military applications, and gesture tracking devices, among others. For example, in U.S. Patent No. 9,417,693 (the "'693 Patent"), incorporated herein by reference in its entirety, different implementations for a wearable wireless human-machine interface (HMI) are described. In the '693 Patent, a user can control a controllable device based on gestures performed by the user using the wearable HMI. In some implementations, a controller to track motion and correct for drift, as described herein, may be connected to the IMUs of the wearable HMI. In some implementations, the controller is attached to or integrated in the wearable HMI. In some implementations, the controller is remote from the wearable HMI but communicatively coupled to the wearable HMI.

[0046] Figures 1A-1F illustrate various configurations of motion sensors mounted on 3D objects, in accordance with some implementations. Motion sensors may be mounted in linear arrays, on planar surfaces, or vertices of a myriad of geometric configurations, formed by any dimensional planar surface, platonic solid, or irregular 3D object. As long as the distances and angles are known between the mounted motion sensors, drift can be eliminated by, among certain methods or portions thereof described herein, resetting the motion sensors' instantaneous measured acceleration, angular velocity, magnetic orientation, and altitude to match the known geometry formed by the physical distances and angles of the motion sensors relative to each other, as further described below in reference to flowcharts 4A-4D below.

[0047] In a linear geometry, as shown in Figure 1A, two sensors 102, 104 are positioned adjacent to each other at a fixed distance 128, and the angles between the two sensors can be considered to be approximately 0 degrees or approximately 180 degrees. As the measured distance or angle between the two fixed sensors 102 and 104, at any given instantaneous reading, deviates from the known distance 128, or angle between them, this drift can be removed and positions of the two motion sensors can be reset to a fairly accurate degree.

[0048] A planar configuration of three (3) or four (4) or more sensors can provide a spatial calculation based on a higher number of IMU readings of instantaneous measurements of all sensors in the array with known physical angles and distances between them. Figure 1B shows a four-sensor configuration with sensors 106, 108, 110, and 112 mounted adjacent to each other in a planar configuration. Planar configurations, such as the configurations shown in Figures 1A and 1B, provide a simpler mathematical model with fairly low demand for computation. However, variations in axial motion detection methods of the physical sensors may affect the accuracy of measurement in different axes of motion and orientation. For example, motion in the Z-axis of a MEMS-based sensor is heavily biased with a gravity vector which may introduce higher variance in the physical motion of the sensor in this axis. Additionally, the Coriolis force, used to calculate Yaw in the z-axis, is also susceptible to larger variance than the X, or Y axis.

[0049] For improved drift correction, a tetrahedron configuration with four (4) sensors, each one mounted on each face of the tetrahedron can provide a blend of multi-axial data resulting in better complementary and compensatory measurement for the gravity vector bias than a single, Z-Axis of all sensors, according to some implementations. Figures 1C and 1D show one such configuration. Figure 1C shows the top – oblique view of a tetrahedron with motion sensors 114, 116, 118 mounted on each of the visible three faces. Figure 1D shows the bottom – oblique view of the tetrahedron shown in Figure 1C showing the additional sensor 120 on the fourth face of the tetrahedron. In this configuration, a component of the X and Y axis is also exposed to the gravity vector from at least three sensors at any given time, permitting a higher degree of accuracy through the removal of the gravity vector from a number of sensors and a number of axes at any instantaneous measurement. Sensors are mounted at angles on each surface, providing a blend of X, Y, and Z axis data for better spatial calculations and drift correction, in accordance with some implementations.

[0050] Furthermore, a cubic configuration will provide a higher sensor count of six (6) sensors on the six (6) surfaces of the cube to stabilize the spatial deviation even further. Figure 1E shows an oblique view of a cubic configuration, according to some implementations. Only three out of the six faces are visible in Figure 1E. Each of the six faces may have a sensor mounted, including the sensors 122, 124, and 126. In some implementations, some, less than all, faces of any object described herein have at least one sensor. In this configuration, each sensor on each face enables a complementary reading between the other sensors on the other faces of the cube. However, as the number of sensors is increased, the latency to read all measurements is also increased in the cubic or higher dimensional solid geometries.

[0051] Motion sensors can also be rotated on opposite faces of the geometric solids to provide an axial blend in any configuration, according to some implementations. Figure 1F shows an oblique view of another configuration of the cuboid in Figure 1E wherein motion sensors are mounted on each face of the cube as before, but sensors may be rotated at an angle between zero (0) and ninety (90) degrees, non-inclusive. For example, sensor 122 may be rotated at an angle of approximately forty-five (45) degrees with respect to the other sensors. Although this method may provide a better analysis of instantaneous motion data, the computation time per measurement-to-calculation output may be longer.

[0052] Figure 2 is a block diagram illustrating a representative system 200 with drift-free sensor(s), according to some implementations. In some implementations, the system 200 includes one or more processing units 202 (e.g., CPUs, ASICs, FPGAs, microprocessors, and the like), one or more communication interfaces 214, memory 220, and one or more communication buses 216 for interconnecting these components (sometimes called a chipset). The type of processing units 202 is chosen to match the requirement of application, including power requirements, according to some implementations. For example, the speed of the CPU should be sufficient to match application throughput.

[0053] In some implementations, the system 200 includes a user interface 208. In some implementations, the user interface 208 includes one or more output devices 210 that enable presentation of media content, including one or more speakers and/or one or more visual displays. In some implementations, user interface 208 also includes one or more input devices 212, including user interface components that facilitate user input such as a keyboard, a mouse, a voice-command input unit or microphone, a touch screen display, a touch-sensitive input pad, a gesture capturing device, or other input buttons or controls. Furthermore, some systems use a microphone and voice recognition or a camera and gesture recognition or a motion device and gesture recognition to supplement or replace the keyboard.

[0054] In some implementations, the system 200 includes one or more Inertial Measurement Unit(s) 204. In some implementations, the IMUs include one or more accelerometers, one or more magnetometers, and/or one or more gyroscopes, and/or altimeters, and/or pressure sensors. In some implementations, the one or more IMUs are mounted on an object that incorporates the system 200 according to a predetermined shape. Figures 1A-1F described above illustrate various exemplary configurations of motion sensors. In some implementations, the initial configuration of the IMUs (e.g., the number of IMUs, the predetermined shape) is also determined based on characteristics of the individual IMUs. For example, the orientation or the axis of the IMUs, and therefore the predetermined shape, are chosen so as to compensate for manufacturing defects. In some implementations, the one or more IMUs are fabricated as CMOS and MEMS system on a chip (SOC) that incorporates the system 200.

[0055] Communication interfaces 214 include, for example, hardware capable of data communications using any of a variety of custom or standard wireless protocols (e.g., IEEE 802.15.4, Wi-Fi, ZigBee, 6LoWPAN, Thread, Z-Wave, Bluetooth Smart, ISA100.11a, WirelessHART, MiWi, etc.) and/or any of a variety of custom or standard wired protocols (e.g., Ethernet, HomePlug, etc.), or any other suitable communication protocol, including communication protocols not yet developed as of the filing date of this document.

[0056] Memory 220 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices; and, optionally, includes non-volatile memory, such as one or more magnetic disk storage devices, one or more optical disk storage devices, one or more flash memory devices, one or more EPROMs, one or more EEPROMs, or one or more other non-volatile solid state storage devices. Memory 220, or alternatively the non-volatile memory within memory 220, includes a non-transitory computer readable storage medium. In some implementations, memory 220, or the non-transitory computer readable storage medium of memory 220, stores the following programs, modules, and data structures, or a subset or superset thereof:

- operating logic 222 including procedures for handling various basic system services and for performing hardware dependent tasks;
- device communication module 224 for connecting to and communicating with other network devices (e.g., network interface, such as a router that provides Internet connectivity, networked storage devices, network routing devices, server system, etc.) connected to one or more networks via one or more communication interfaces 214 (wired or wireless);
- input processing module 226 for detecting one or more user inputs or interactions from the one or more input devices 212 and interpreting the detected inputs or interactions;
- user interface module 228 for providing and displaying a user interface in which settings, captured data, and/or other data for one or more devices (not shown) can be configured and/or viewed;
- one or more application modules 230 for execution by the system 200 for controlling devices, and for reviewing data captured by devices (e.g., device status and settings, captured data, or other information regarding the system 200 and/or other client/electronic devices);
- one or more controller modules 240, which provides functionalities for processing data from the one or more IMUs 204, including but not limited to:
    - data receiving module 242 for receiving data from the one or more IMUs 204 that is to be processed by the controller module(s) 240;
    - filtering module 244 for removing noise from the raw data received by the data receiving module 242;
    - dynamic calibration module 246 for cross-correlating the data between the one or more IMUs 204 (e.g., different gyroscopes and accelerometers of the one or more IMUs 204) to calibrate filtered data for the one or more IMUs 204;

- motion decomposition module 248 that determines positional and rotational state based on the decomposed output for each of the one or more IMUs;
- motion synthesis module 250 for synthesizing motion based on the output of the dynamic calibration module 246 and the motion decomposition module 248;
- drift correction module 252 for correcting drift in the sensor output (e.g., using an Adaptive Continuous Fuzzy Rule (without modus ponens) Bayesian Filter with Trapezoidal Motion Parameters (ACFBT) for the predetermined shape based on the output from the motion synthesis module 250; and
- edge condition handling module 254 that handles complex movements based on the output (e.g., using Artificial Intelligence/Neural Networks/Deep Learning) of the drift correction module 252.
- receiving absolute position module 256 that receives a first object initial absolute position.
- calculate absolute position module 258 that calculates, a first object current absolute position using an output of the IMU and the first object initial absolute position.
- receive referential data module 260 that, in conjunction with communication interface 214, receives, the wireless transceiver referential data from the second object, the referential data may include an second object current absolute position calculated using a second plurality of IMUs associated with the second object
- calculate relative position module 262 that calculates a relative position of the first object relative to the second object using the first object current absolute position and the second object current absolute position, wherein the relative position may include at least one of: (i) a distance between the first object and the second object and (ii) an orientation of the first object relative to the second object.

[0057] In some implementations, the raw data received by the data receiving module 242 from the IMUs include acceleration information from accelerometers, angular velocities from gyroscopes, degrees of rotation of magnetic field from magnetometers, atmospheric pressure from Altimeters, and differential Pressure Sensors. The raw data is received from each of the IMUs sequentially, according to some implementations. In some implementations, the IMU data is received in parallel.

[0058] In some implementations, the filtering module 244 filters the raw data to remove noise from the raw data signals received by the data receiving module 242. The filtering module 244 uses standard signal processing techniques (e.g., low-pass filtering, clipping, etc.) to filter the raw data thereby minimizing noise in sensor data, according to some implementations. The filtering module 244 also computes moving averages and moving variances using historical data from the sensors, according to some implementations.

[0059] In some implementations, the dynamic calibration module 246 uses an Artificial Intelligence (AI) framework (e.g., a neural network framework) to calibrate data from the one or more IMUs 204. For example, one or more “neurons” (typically 3 per sensor) are configured in a neural network configuration to calibrate the filtered data for the one or more IMUs 204. To understand how dynamic calibration works, consider first a static configuration for an object. Let us assume further that the shape of the object (sometimes herein called a predetermined shape) is a cuboid for the sake of explanation. A cuboid-shaped object could be placed on a planar surface six different ways (i.e., on six different faces of the cuboid). So there are six orientations to calibrate on. In this static configuration, the system 200 collects a large number of samples (e.g., approximately 1,000 or more samples) for each of the six orientations. This sampled data is collected and stored in memory 220. Later, when raw data is received, the stored sampled data is used as a baseline to correct any offset error in the raw data during sedentary states (i.e., when the object is not moving). In some implementations that use a neural network, the weights of the network are constantly tuned or adjusted based on the received raw data from the IMUs after offsetting the stored sampled data, according to some implementations. A neural network-based solution provides better estimates than a least squares regression analysis or statistical measures. As an example of how the neural network weights are adjusted dynamically, consider when the object is stationary but the neural network output indicates that the object is moving. The weights are readjusted, through back propagation, such that the output will indicate that the object is stationary. Thus the weights settle during times when the object is stationary. In some implementations, the learning rate of the neural network is maximized during sedentary states (sometimes herein called stationary states), and minimized when the object is in motion. Pattern recognition is used to detect whether the object is moving or is stationary so that the learning rate can be adjusted, according to some implementations. The different stationary and mobile states are used to adjust the weights affecting the accelerometer. In some implementations, known reference to the magnetic north is used to constantly adjust the weights that correspond to the magnetometers. In some implementations, the magnetometer data is also used to correct or settle the weights for the accelerometers when the object is moving because the reference point for the magnetic north and gravity vector are always known. Gyroscope data is more reliable than data from accelerometers because it only requires a single level integration. So the gyroscope data is also used to correct accelerometer weights, according to some implementations. It is noted, in some implementations, that the dynamic calibration module 246 is optional, and a pass-through channel passes the output of the filtering module 244 to the motion synthesis module 250 without dynamic calibration.

[0060] In some implementations, the motion decomposition module 248 uses pattern recognition techniques to eliminate anomalies due to cross-interaction or interference between the sub-sensors, in each IMU. Experimental data is collected for controlled translational and rotational movements of an object. For example, the behavior of the gyroscope is tracked under constant velocity and the pattern is stored in memory. When the gyroscopic data follows the known pattern, the fact that the object is under constant velocity is deduced based on this pattern. Similarly, accelerometer data (e.g., constant gravity vector) can be used to identify patterns to correct errors in gyroscopic data and/or magnetometer data, and magnetometer data can be used to identify patterns to correct errors in accelerometer data and/or gyroscope data, according to some implementations. For example, in some implementations, the motion decomposition module 248 distinguishes between constant velocity state and stationary state of the object. For example, while the object is in a constant velocity state, such as when the object is moving at a constant velocity, the gyroscope registers noise (due to vibrations) which is captured as a signature (or a pattern) and stored in memory. The noise may cause the gyroscope to register that the object is moving at varying velocities rather than a constant velocity. On the other hand, while the object is in a constant velocity state, the accelerometer does not show change in output under constant velocity. Some implementations detect the differences in the behavior (e.g., noise level) of the gyroscope, and/or the absence of change in output of the accelerometer to deduce that the pattern corresponds to an object in a constant velocity state. In such instances, because the object is in a constant velocity state, the motion decomposition module 248 uses a previously calculated velocity for current position measurements as discussed herein.

[0061] In some implementations, the motion decomposition module 248 removes anomalies by observing changes in patterns detected from sensor data, such as when the object stops moving or rotating abruptly, as another effect to correct for anomalies. In some implementations, the motion decomposition module 248 analyzes several distinct stored patterns for correcting anomalies in each of the sensors. In some implementations, the motion decomposition module 248 categorizes the type of translational and/or rotational movements of each IMU of the tracked object and outputs the pattern or the category for the motion synthesis module 250. For example, the motion decomposition module 248 deduces that each IMU is in one of many states, including simple linear motion, simple linear motion with rotation, non-linear motion with simple rotation. In some implementations, output from the motion decomposition module 248 additionally controls the learning rate in the dynamic calibration module 246.

[0062] In some implementations, the motion synthesis module 250 uses the state information (e.g., constant velocity, constant acceleration, changing acceleration, in combination with rotation) from the motion decomposition module 248 to select one or more algorithms. The motion synthesis module 250 subsequently applies the one or more algorithms on the data output from dynamic calibration module 246 to synthesize the motion of the object (sometimes herein referred to as the computation of overall rectified data for the one or more IMUs). For example, if the state information from the motion decomposition module 326 indicates that the object is rotating, the motion synthesis module 250 uses an equation to compute the axis of rotation based on the difference in angular momentum of the IMUs (as indicated by the output of the dynamic calibration module) and the known shape outlined by the predetermined position of the different IMUs. To elaborate on this example, suppose the object is mounted with IMUs in a planar configuration, such as in Figure 1B, with four sensors, each sensor in a corner. Suppose further that the planar configuration positioned vertically in a diamond shape, with the longitudinal axis passing through the top IMU and the bottom IMU. Now, if the planar (diamond-shaped) object is rotated about the longitudinal axis, the side IMUs on either side of the longitudinal axis will share the same angular momentums but will have different angular momentums as compared to the top IMU and the bottom IMU, and the top IMU will have an angular velocity greater than the bottom IMU that is closer to the axis of rotation. The motion synthesis module 250 computes or synthesizes the rotational axis data from the differences in the angular momentums and the known distances between the sensors, based on the shape formed by the IMUs.

[0063] In some implementations, the drift correction module 252 uses shape correction which, in some implementations, to remove drift by re-conforming sensor positions and orientation to the known (sometimes herein called predetermined) shape. In some implementations, the drift correction module 252 computes the skewness in the data by the motion sensors based on the variation in the norms, distances and angles between the sensors. If the variation in the norms exceeds a threshold, the drift correction module 252 generates a correction matrix (sometimes called a drift matrix) to eliminate drift in successive sensor readings. A shape correcting module (not shown) corrects the data output from the dynamic calibration module (sometimes herein called the clean or filtered data) using the correction matrix, by subtracting the predicted drift from the clean data, in a continuous or iterative fashion, according to some implementations. For example, after every reading of sensor data, previously generated and stored data from the drift correction module 252 is used to correct the clean data output from the noise-filtered, and dynamic calibration module, according to some implementations.

[0064] In some implementations, the edge condition handling module 254 handles complex movements (e.g., while spinning along two axes, and moving across on a straight line, say the object also lifts up) and/or transitional movements (e.g., spinning to laterally moving along a straight line) to reduce drift based on the output of the drift correction module 252. In some implementations, the edge condition handling module 254 uses AI to apply probability weightings to compensate for the edge conditions. In some implementations, the edge condition handling module 254 blends a current object common data point (e.g., output by the drift correction module 252) and the previous object common data point (e.g., previous output for a prior sensor reading by the drift correction module 252 that is stored in memory) to remove the edge condition.

[0065] Even though in some complex scenarios, drift may not be completely corrected, perpetual drift or constant drift can be eliminated with the aforementioned modules. Furthermore, in some implementations, the drift observed by the combination of the modules described herein is in the order of centimeters or even millimeters, whereas alternate external reference based drift elimination (e.g., using a GPS) could sometimes result in a drift in the order of meters.

[0066] In some implementations, the one or more controller module(s) 240 includes device-related information. In some implementations, the device related information includes device identifier, and/or device characteristics. In some implementations, the device identifier may identify the device to other objects in the network. In some implementations, the device characteristics include information related to whether the device corresponds to an object that is operated manually or autonomously. In some implementations, the device characteristics include information related to whether the device corresponds to a static object, such as a building or an appliance, or a dynamic object, such as an automobile. In some implementations, the device characteristics include information related to device operational state, such as whether a device is on or off.

[0067] In some implementations, the one or more controller module(s) 240 includes location-related information (e.g., absolute positions) for other objects. Some implementations include specific features (or characteristics or operational states) and/or encoding of such features, of the system. In some implementations, the operational state of an object may change based on certain criteria detected in the network. For example if the device is embedded in a lamp post that has a light bulb that switches on/off, this characteristic is stored in the modules 240. In some implementations, this information relates to objects (e.g., lights) inside buildings, so the location of such objects inside the building are also stored. Similarly, if the device is in a mobile object (e.g., a car), the object's characteristics are also stored in the modules 240. For example, such information may include information on whether the object, such as a car, can switch on/off turn signals, etc. In some implementations, the characteristics described above are communicated to and/or received from other objects using the receive referential data module 260 that, in conjunction with communication interface 214, sends and/or receives, the wireless transceiver referential data from other objects. In some implementations, the modules 240 store information related to other objects up to a maximum or predetermined number of objects, and/or calculates information related to those objects that do not have relevant information stored, based on the stored information.

[0068] Each of the above identified elements may be stored in one or more of the previously mentioned memory devices, and corresponds to a set of instructions for performing a function described above. The above identified modules or programs (i.e., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules may be combined or otherwise rearranged in various implementations. In some implementations, memory 220, optionally, stores a subset of the modules and data structures identified above. Furthermore, memory 220, optionally, stores additional modules and data structures not described above. In some implementations, one or more processing modules and associated data stored in the memory 220 are stored in and executed on a second processing device other than the system with drift-free motion sensors 200 that is configured to receive and process signals produced by the IMUs 204. For example, the second processing device might be a computer system, smart home device or gaming console that executes applications (e.g., computer games) at least some of whose operations are responsive to motion signals provided by the IMUs.

[0069] Figure 3 is a flow diagram illustrating the flow of sensor data through a representative system with drift-free sensor(s), according to some implementations. Raw data from the one or more IMUs 302 (e.g., IMU 0, IMU 1, IMU 2, …, IMU N) is received (324) by the controller 300 (e.g., controller module 240). As previously mentioned, in some implementations, the controller receives the data from the one or more IMUs in parallel (as shown in Figure 3). The received data is output as raw data (304) to the motion decomposition module 326, according to some implementations. In some implementations, the raw data is also input as data 306 to a filtering module 328 which filters the raw data to produce filtered data 310 which is in turn input to a dynamic calibration module 330. In some implementations, the motion decomposition module 326 also controls (314) the learning rate of the dynamic calibration module 330. In some implementations, the motion decomposition module 326 and/or the dynamic calibration module 330 are optional modules. In such cases, the filtered data 310 is input (not shown) to the motion synthesis module. The motion synthesis module 332, in these cases, does not know the pattern or category of motion but iteratively applies one or more algorithms or equations to synthesize motion. In some implementations, steps of the motion decomposition module 326 and the dynamic calibration module 330 execute asynchronously and/or in parallel. The Bayes calculation step 336 uses the output 316 of the motion synthesis module to generate drift correction matrices 320 (as described previously with reference to Figure 2) which is consumed by a shape correction module 334 to correct input in the next iteration (i.e., when and after such data becomes available) of motion synthesis. Although not shown in Figure 3, in some implementations, during a first set of one or more iterations, the shape correction data is not available, and the dynamic calibration output 312 is input to the motion synthesis module 332. In some implementations, the output of the Bayes calculation step 336 (318) is input to an edge conditions module 338 to handle edge conditions (described above in reference to Figure 2) for complex movements and dynamic learning. The output 322 indicates drift-free real motion output of the controller, according to some implementations.

[0070] In some implementations, filtering module 328 includes similar functionality to filtering module 244 in Figure 2; the motion decomposition module 326 includes similar functionality to the motion decomposition module 248 in Figure 2; dynamic calibration module 330 includes similar functionality to dynamic calibration module 246 in Figure 2; shape correction module 334 includes similar functionality to shape correction module described above in the description for Figure 2; the motion synthesis module 332 includes similar functionality to the motion synthesis module 250 in Figure 2; Bayes calculations module 336 includes similar functionality to drift correction module 252 in Figure 2; and the edge conditions module 338 includes similar functionality to the edge condition handling module 254 in Figure 2.

[0071] Figures 4A-4D illustrate a flowchart representation of a method 400 of tracking position and orientation of an object using a tracking device, according to some implementations. In some implementations, the tracking device includes (402) one or more sides that define a predetermined shape, and a plurality of inertial measurement units (IMU) mounted to the one or more sides of the predetermined shape. Further, each IMU includes a first sub-sensor and a second sub-sensor, and each IMU is positioned at a predetermined distance and orientation relative to a center of mass of the tracking system, according to some implementations. Figures

1A-1F described above illustrate various configurations of sensors mounted on 3D objects, according to some implementations. In some implementations, the first sub-sensor and the second sub-sensor of the tracking device (e.g., IMUs 204 of Figure 2) are (404) each one of: accelerometer, magnetometer, gyroscope, altimeter, and pressure sensor and the first sub-sensor is a different sensor type than the second sub-sensor. In some implementations, the predetermined shape of the tracking device is (406) one of: a plane, a tetrahedron, and a cube. In some implementations, the tracking device also includes a controller communicatively coupled to the plurality of IMUs. An example system 200 with IMUs 204 was described above in reference to Figure 2, according to some implementations.

[0072] In some implementations, each IMU of the tracking device detects (408) movement of the object and generates inertial output data representing location and/or orientation of the object. For example, IMUs 204 in Figure 2 or the sensors in Figures 1A-1F use a combination of accelerometers, magnetometers, gyroscopes, altimeters, and/or pressure sensors to detect movement of the object and generate data that represents location and/or orientation of the object.

[0073] In some implementations, the tracking object, at the controller (410), receives (412) first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs. For example, the data receiving module 242 of the system 200 receives the output from the one or more IMUs 204 via the one or more communication buses 216. In some implementations, the controller receives (414) the first sub-sensor inertial output data and the second sub-sensor inertial output data from each of the plurality of IMUs periodically at less than approximately 1 ms for continuous high sampling rate.

[0074] In some implementations, the controller uses a filtering module (e.g., module 244) to filter (416) the first sub-sensor inertial output data and second sub-sensor inertial output data to minimize signal noise.

[0075] Referring now to Figure 4B, the controller performs a sequence of steps 418 for each IMU, according to some implementations. In some implementations, the controller generates (420) calibrated inertial output data based on the first sub-sensor inertial output data and the second sub-sensor inertial output data. For example, the controller uses the dynamic calibration module 246 to generate calibrated inertial output data. In some implementations, the controller calculates the error value using (422) neural network weights to evaluate the first sub-sensor inertial output data and the second sub-sensor inertial output data, wherein the weights are adjusted at a learning rate based on the positional state (e.g., stationary position state) of the tracking device, calculating a discrepancy value representative of a difference between an actual movement of the object and estimated movement of the object, and removing the discrepancy value from the calibrated inertial output data, (e.g., using the output of a motion decomposition module, such as module 248). In some implementations, the controller applies (424) neural network weights to the first sub-sensor inertial output data and the second inertial output data based on historical (e.g., prior or previous) inertial output data from each of the first and second sub-sensors. Although not shown, the controller stores and/or accumulates inertial output data received from the IMUs over time that is later retrieved as historical data.

[0076] Referring next to Figure 4C, the controller uses the dynamic calibration module (e.g., module 246) to cross-correlate (426) the first sub-sensor inertial output data with the second sub-sensor inertial output data to identify and remove anomalies from the first sub-sensor inertial output data with the second sub-sensor inertial output data to generate decomposed inertial output data for each IMU, according to some implementations. In some implementations, the controller calibrates (428) the decomposed inertial output data corresponding to the first sub-sensor based on the second sub-sensor inertial output data. In some implementations, the controller cross-correlates the first sub-sensor inertial output data with the second sub-sensor inertial output data by applying (430) pattern recognition (e.g., by using a motion decomposition module, such as module 248) to the second sub-sensor inertial output data to generate the decomposed inertial output data representative of the first sub-sensor inertial output data.

[0077] Next, referring to Figure 4D, the controller determines (432), using a motion decomposition module (e.g., module 248 described above), a positional and rotational state of the tracking device based on the decomposed inertial output data from each of the IMUs, according to some implementations.

[0078] Subsequently, the controller synthesizes (434), using a motion synthesis module (e.g., module 250 described above), first sub-sensor inertial output data and second sub-sensor inertial output data to create IMU synthesized data using a synthesizing methodology based on the positional and rotational state of the tracking device, according to some implementations.

[0079] In some implementations, the controller calculates (436), using a ACFBT calculation module (not shown), a current tracking device rectified data output based on the data synthesized for each of the IMUs, a predetermined position of each of the IMUs and a predetermined orientation of each of the IMUs to confirm to a predetermined shape. In some implementations (438) of the controller, at least some of the IMUs used to calculate the common data point are oriented at different angles along two different axes relative to each other.

[0080] The controller subsequently calculates (440), using a current position and orientation determination module (e.g., module 252 in Figure 2, or steps 336 and 334 in Figure 3), a current position and orientation of an object based on a difference between the current object rectified data output and a previous object rectified data output, according to some implementations. In some implementations, the controller identifies (442) an edge condition (e.g., complex movements described above) and blends (444), using an edge condition handling module (e.g., module 254 described above), the current object rectified data output and the previous object rectified data output to remove the edge condition.

[0081] It should be understood that the particular order in which the operations in Figures 4A-4D have been described is merely an example and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein.

[0082] Referring next to Figures 5A-5D, there is shown diagrams illustrating an exemplary implementation of calculating the position of an object, relative to another object using the drift free motion sensor system 200-i described herein. In this exemplary implementation, each of the cars 502 in Figures 5A-5D are the “objects”. As shown in Figure 5 and described in more detail below, the drift free motion sensor system 200-i IMU may be connected to different objects (also referred to herein as nodes) in a “smart city” configuration and may each track distances and/or direction of motion of its own moving object as well as other moving objects or nodes in an interconnected mesh network with precision, accuracy, and redundancy so that different objects can maneuver throughout an environment with other objects without colliding. In some implementations, an object may be a vehicle, a cell-phone, a mobile device, a building, a stationary light-pole, among other things. While the drift free motion sensor system 200-i may operate without calibration from other external objects, by adding more objects in a mesh-network configuration, more redundancy and fail-safe options may be created. For example, if some nodes in the mesh network fail to communicate position data, the remaining nodes in the mesh network may take over and compensate for the failed nodes.

[0083] In other implementations, the objects may correspond to other devices including, but not limited to, mobile computing devices, projectiles, helmet mounted displays, gaming consoles, or other devices included in Exhibit A.

[0084] Turning now to Figures 5A-5D, first car 502-1, second car 502-2 and third car 502-3 may each be traversing along a roadway. Each of the cars 502 may have a drift free motion sensor system 200-i to track the position of itself and other cars as the cars continue traversing along the roadway. By utilizing position data of each of the cars, a particular car may either alert a driver or alter a driving path of the particular car in response to a determination that the cars may collide at some point along the roadway. The drift free motion sensor system 200-i may each include a respective controller 300, a wireless transceiver 214, and one or more IMUs 302. The controller 300, in conjunction with the IMUs 302 may be configured to provide drift free orientation and position data.

[0085] Initially, the first car 502-1 may be configured to receive from an external source an initial absolute position (e.g., seed position) to the drift free sensor system 200-1 of the first car 502-1. The initial absolute position may be a coordinate position in e.g., a latitude/longitude format (e.g., XX Latitude and YY Longitude), among others. As used herein, the term absolute position may refer to a position of an object relative a predefined position on the earth. The predefined position on the earth may correspond to a city, province, road, or building.

[0086] While the car 502-1 continues traversing along the roadway, the first car 502-1 may then sense, using the IMU of the drift free sensor system 200-1, that the first car 502-1 is in motion at a velocity of 65 km/hr and has moved 10 meters North.

[0087] The first car 502-1 may then generate, using the first plurality of IMUs 200-1 and controller 300, a motion signal representative of the motion of the first car 502-1. The motion signal may be calculated using one or more of the modules of controller 300, as shown in Figure

3 and described herein. In some implementations, the motion signal may be generated by calculating a rectified data output based on sensed motion data from each of the first plurality of IMUs, a predetermined position of each of the first plurality of IMUs and a predetermined orientation of each of the first plurality of IMUs

[0088] As shown in Figure 5B, the first car 502-1 may then calculate, using the controller 300, a first car 502-1 current absolute position. For example, the first car 502-1 may calculate the current absolute position of the first car 502-1 to be XX Latitude and YY + 10m Longitude, using an output of the IMU 200-1 and the first car 502-1 initial absolute position by, for example, summing the output of the IMU with the latitude and longitude coordinate data.

[0089] As shown in Figure 5C, the first car 502-1 may receive referential data from one or more other cars (e.g., car 502-2 and car 502-3). In some embodiments, one or more of the cars (e.g., car 502-3) may include a wireless transmitter that lacks the capability to transmit referential data to the first car. In these implementations, a mesh network may be created whereby the second car 502-2 may relay the referential data from the third car 502-3 to the first car 502-1. For ease of understanding, only three cars are shown, however, in some implementations, the mesh network may include N cars (or objects generally), with each car relays referential/positional data from one car to another car. Third car 502-3 may send referential data of the third car 502-3 to the second car 502-2 and the second car 502-2 may send the referential data of the second car 502-2 and third car 502-3 to the first car 502-1. The referential data of the second car 502-2 may include the current absolute position of the second car 502-2 calculated using a second plurality of IMUs 200-2 associated with the second car 500-2. The referential data of the third car 502-3 may include the current absolute position of the third car 502-3 calculated using a third plurality of IMUs 200-3 associated with the third car 502-3. The first car 502-1 may then receive, using the wireless transceiver of the first car 502-1, referential data from the second car 502-2 and third car 502-3.

[0090] As shown in Figure 5D, the first car 502-1 may calculate a relative position of the first car 502-1 relative to the second car 502-2 and third car 502-3 using the first car 502-1 current absolute position, the second car 502-2 current absolute position and/or the third car 502-3 current absolute position. The relative position may include at least one of: (i) a distance between the first car 502-1 and the second car 502-2 and a distance between the first car 502-1 and the third car 502-3 and (ii) an orientation of the first car 502-1 relative to the second car 502-2 and an orientation of the first car 502-1 relative to the third car 502-3. For example, the first car 502-1 may determine, based on the referential data received from the second car 502-2 and third 502-3 shown in Figure 5C, the distance between the first car 502-1 and the second car 502-2 to be 2 meters and the distance between the first car 502-1 and the third car 502-3 to be 5 meters. In some implementations, the relative positions of the first car 502-1 relative to the second car 502-2 and third car 502-3 may be calculated without using an external reference signal.

[0091] As shown in Figure 5D, the first car 502-1 may calculate a velocity of the second car 502-2 and third car 502-3 using current absolute position of the first car 502-1, current absolute position of the second car 502-2, and/or current absolute position of the third car 502-3. In the example shown in Figure 5D, the velocity of the second car 502-2 and third car 502-3 is 65 km/hr.

[0092] In some implementations, each object may share an observation of a first external object with a second external object. For example, the first car 502-1 may calculate a position of the second car 502-2, and communicate the observation of the position of the second car 502-2 to the third car 502-3. The third car 502-3 may use the observation of the position of the second car 502-2 received from the first car 502-1 rather than, or in addition to, calculating the position of the second car 502-2 independently. Similarly, the third car 502-3 can calculate a position for the second car 502-2 and communicate the same to the first car 502-1.

[0093] In some implementations, the objects are connected as nodes in a mesh network allowing the objects to reinforce calculations with observations from the other objects. With the exchange of information, entropy (loss of information) is decreased over time. In this way, in some implementations, each node (or the car in this example) characterizes positions (either absolute or relative positions) by reconciling its calculated positions with that of the information received from other nodes. In some implementations, the objects behave like an elastic system, reinforcing accurate estimates, and revert to a point of rigidity that reinforce correctness, and in some sense, perpetually and/or continuously self-correcting.

[0094] In some implementations, the drift free sensor system 200-1 stores historical data (e.g., previous 15 states) and applies the historical data for correction. Some implementations store timestamps for the entries, and/or weigh more recent entries for improving accuracy. Some implementations repeat the process of calculating, exchanging, and reconciling information, over several cycles. In some implementations, the drift-free sensor system 200-1 stores data from other objects in the order it is received, and/or with timestamps. Suppose recent data does not correlate with historical data, some implementations use a Hidden Markov Model (HMM) and use Bayesian probabilities to calculate an accurate estimate of absolute positions of other objects. Some implementations store all observations as time-ordered entries and combine or fuse time entries, depending on storage constraints.

[0095] Some implementations recognize new nodes (cars, in our example) and adjust calculation of relative positions accordingly. For example, suppose a node just started or joins the mesh network. The node does not have prior prediction, and initially incurs more errors in its calculation that stabilizes (or collapses errors) over time (e.g., after 2-3 iterations). Other nodes in the network also recognize the new node, and weigh the information from the node accordingly. In some implementations one or more nodes do not have own sensors but are merely calculating and/or relaying information based on information received from other objects or nodes. In some implementations, other nodes in the network recognize the nodes without own sensors, and weigh information obtained from such nodes accordingly (e.g., provide lower weight to such observations).

[0096] In some implementations, the drift free sensor system 200-1 may determine that the relative position of the first car 502-1 to the second car 502-2 meets emergency criteria. For example, the second car 502-2 may be swerving towards the first car 502-1, such that the first car 502-1 and second car 502-2 may collide at some point in the future. In response, the drift free sensor system 200-1 may either alert the driver and/or cause the first car 502-1 to perform an evasive maneuver, wherein the evasive maneuver includes braking the first car and/or turning the first car 502-1 to avoid the second car 502-2.

[0097] In some implementations, the drift sensor system 200-1 may control one or more objects based on the prediction of direction, position, orientation, and/or acceleration of moving objects. For example, the drift sensor system 200-1 may switch on/off a home’s electrical systems (e.g., cooling systems) or an oven inside a home, in response to an object (e.g., the first car 502-1) moving towards the home or an object inside the home (e.g., a toaster oven) or in response to the drift sensor system 200-1 calculating a relative position of the external object and detecting that a distance between the external object and the home (or object inside the home) is within a predetermined threshold. To further illustrate, in another context, a lamp post, on a city street, or a driveway of a home, may switch on (switch off) automatically in response to detecting that a car is approaching (leaving) the lamp post. Similarly, such systems can be used for traffic flow analysis, to predict number of moving objects (e.g., cars, humans with wearable devices, mobile phones) in an area. In the context of Internet of Things (IoT) environment, IoT devices may be controlled via the drift sensor system 200-1, a controller on a motherboard coupled to the system 200-1, or a communication controller that is communicatively coupled (e.g., using a wireless service) to the IoT device.

[0098] In some implementations, the car 502-1 may include a user interface that displays a graphical representation of a map. The drift free sensor system 200-1 may display a position of the car 502-1 on the graphical representation of the map using the relative position of the first object relative to the second object.

[0099] In some implementations, the drift free sensor system 200-1 may utilize map data to calibrate position data.

[00100] In some implementations, the drift free sensor system 200-1 may calculate a velocity of the first car 502-1 using the referential data from the second car 502-2. In some implementations, the drift free sensor system 200-1 may be connected to the on-board diagnostic (OBD) system of the first car 502-1 to receive velocity data from the OBD system. The velocity data may be used by the motion synthesis module 250 as state information to select one or more motion synthesis algorithms as described herein.

[00101] In some implementations, the drift free sensor system 200-1 may utilize the referential data to update (e.g., calibration or redundancy check) the absolution position of the first car 502-1. For example, the drift free sensor system 200-1 may triangulate the absolute position of the first car 502-1 using the referential data.

[00102] In some implementations, the drift free sensor system 200-1 may calculate velocities of other cars or objects based on change in relative positions of the objects over time. For example, suppose the second car is at a relative position p1 at time t1, and at a relative position p2 at time t2, the drift free sensor system 200-1 may calculate the relative velocity of the second car by dividing the absolute difference between p1 and p2 by the difference between t1 and t2.

[00103] Figure 6 illustrates a flowchart representation of a method 600 of calculating the position of a first object relative to a second object, according to some implementations. For example, and with reference to Figures 5A-5D, first car 502-1 calculates the position of first car 502-1 relative to second car 502-2 and third car 502-3.

[00104] In some implementations, the method is implemented at a first object. The first object may be a static object, such as a light post, traffic light or a building. The first object may be a moving object, such as a car, mobile device, gaming console or projectile. The first object may be a drift free motion sensor system 200 that includes including a controller (e.g., controller 300), a wireless transceiver (e.g., communication interface 214), and a first plurality of inertial measurement units (IMUs) each mounted in one or more positions and orientations relative to other of the first plurality of IMUs. Example orientations/positions of the IMUs are shown in Figures 1A-1F.

[00105] In some implementations, the first object is configured to receive (602) a first object initial absolute position. The first object initial absolute position may be an initial seed position to initiate the IMUs of the first object. The initial absolute position may be in a latitude/longitude format (e.g., XX Latitude and YY Longitude), among others.

[00106] In some implementations, the first object is configured to sense (604), using the first plurality of IMUs (e.g. IMU 200-1), motion of the first object. For example, as shown in Figure 5A, the first car 502-1 may sense, using the IMU of the drift free sensor system 200-1, that the first car 502-1 is in motion and has moved 10 meters North.

[00107] In some implementations, the first object is configured to generate (606) a motion signal representative of the motion of the first object. The motion signal may be calculated using one or more of the modules of controller 300, as shown in Figure 3 and described herein. In some implementations, the motion signal may be generated by calculating a rectified data output based on sensed motion data from each of the first plurality of IMUs, a predetermined position of each of the first plurality of IMUs and a predetermined orientation of each of the first plurality of IMUs.

[00108] In some implementations, the first object is configured to calculate (608), using the controller (e.g., controller 300), a first object current absolute position using an output of the IMU and the first object initial absolute position. For example, as shown in Figure 5B, the first car 502-1 may calculate the current absolute position of the first car 502-1 to be XX Latitude and YY + 10m Longitude, using an output of the IMU 200-1 and the first car 502-1 initial absolute position by, for example, summing the output of the IMU with the latitude and longitude coordinate data.

[00109] In some implementations, the first object is configured to receive (610), using the wireless transceiver (e.g., communication interface 214), referential data from the second object. In some implementations, the referential data includes an second object current absolute position calculated using a second plurality of IMUs associated with the second object. For example, as shown in Figure 5C, the referential data of the second car 502-2 may include the current absolute position of the second car 502-2 calculated using a second plurality of IMUs associated with the second car 500-2.

[00110] In some implementations the first object is configured to calculate (612) a relative position of the first object relative to the second object using the first object current absolute position and the second object current absolute position. In some implementations, the relative position includes at least one of: (i) a distance between the first object and the second object and (ii) an orientation of the first object relative to the second object. For example, as shown in figure 5D, the first car 502-1 may determine, based on the referential data received from the second car 502-2 shown in Figure 5C, the distance between the first car 502-1 and the second car 502-2 to be 2 meters.

[00111] In some implementations, the first object uses a radio communication signal to calculate (e.g., using radiometry) estimated distance of the first object relative to the second object. In some implementations, the distance measurement includes at least one of: (i) a distance measurement between the first object and the second object, by the first object, (ii) a distance measurement between the first object and the second object, by the second object that is relayed through data transmission, using the wireless transceiver, from the second object to the first object, and (iii) a distance measurement between the first object and the second object, by the second object that is relayed through data transmission, using the wireless transceiver, from the first object to the second object. In some implementations, one or more of such measurement methods are used independently for overall rectified relative position estimation. In some implementations, one or more of such measurements are used in communication with IMU rectified output to eliminate error in the device network.

[00112] In some implementations, the referential data from the second object includes a third object current absolute position of a third object calculated using a third plurality of IMUs associated with the third object. The first object is configured to calculate a relative position of the first object relative to the third object using the first object current absolute position and the third object current absolute position. The relative position includes at least one of: (i) a distance between the first object and the third object, (ii) an orientation of the first object relative to the third object. For example, as shown in Figure 5C, a mesh network may be created such that the third car 502-3 may send the referential data of the third car 502-3 to the second car 502-2 and the second car 502-2 may send the referential data of the third car 502-3 to the first car 502-1. The referential data of the third car 502-3 may include the current absolute position of the third car 502-3 calculated using a third plurality of IMUs 200-3 associated with the third car 502-3.

[00113] In some implementations, the first plurality of IMUs generate the motion signal using at least one of: shape correction, static calibration, motion decomposition, dynamic calibration, motion synthesis, and edge condition smoothing.

[00114] In some implementations, the controller contains additional sensors for other internal and/or environmental conditions which typically includes automobile sensors, such as temperature sensors, GPS. In some implementations, additional sensor data is transmitted by the controller using the wireless transceiver in independent data packets or included in the data packets with the reference signal.

[00115] In some implementations, the first object is a first car and the second object is a second car, the method further comprising: after calculating a relative position of the first car relative to the second car, determining whether the relative position of the first car to the second car meets emergency criteria; and in response to determining that the relative position of the first car to the second car meets emergency criteria, causing the first car to perform an evasive maneuver, wherein the evasive maneuver includes braking the first car and/or turning the first car. For example, with reference to Figures 5A-5D, the second car 502-2 may be swerving towards the first car 502-1, such that the first car 502-1 and second car 502-2 may collide at some point in the future. In response, the drift free sensor system 200-1 may either alert the driver and/or cause the first car 502-1 to perform an evasive maneuver, wherein the evasive maneuver includes braking the first car and/or turning the first car 502-1 to avoid the second car 502-2.

[00116] In some implementations, the first object is further configured to display, at a user interface associated with the first object, a position of the first object on a graphical representation of a map using the relative position of the first object relative to the second object. For example, with reference to Figures 5A-5D, the drift free sensor system 200-1 may display a position of the car 502-1 on the graphical representation of the map using the relative position of the first object relative to the second object.

[00117] Some implementations use the device-related information described above (e.g., device identifier, device characteristics, manual or autonomous operation, static or dynamic object, on/off state, etc.) in calculating and/or displaying positions, velocities, and/or orientations of the objects.

[00118] In some implementations, the first object is a home appliance and the second object is a car. The home appliance is configured to, after calculating a relative position of the car relative to the home appliance, determine whether the relative position of the car to the home appliance meets operational state change criteria; and, in response to determining that the relative position of the car to the home appliance meets operational state change criteria, cause the home appliance to change from an off state to an on state.

[00119] In the foregoing description, reference is made in detail to implementations, examples of which are illustrated in the accompanying drawings. Also, in the foregoing detailed description, numerous specific details are set forth in order to provide a thorough understanding of the various described implementations. However, it will be apparent to one of ordinary skill in the art that the various described implementations may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the implementations.

[00120] It will also be understood that, although the terms first, second, etc. are, in some instances, used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first electronic device could be termed a second electronic device, and, similarly, a second electronic device could be termed a first electronic device, without departing from the scope of the various described implementations. The first electronic device and the second electronic device are both electronic devices, but they are not necessarily the same electronic device.

[00121] The terminology used in the description of the various described implementations herein is for the purpose of describing particular implementations only and is not intended to be limiting. As used in the description of the various described implementations and the appended claims, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[00122] As used herein, the term “if” is, optionally, construed to mean “when” or “upon” or “in response to determining” or “in response to detecting” or “in accordance with a determination that,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” is, optionally, construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event]” or “in accordance with a determination that [a stated condition or event] is detected,” depending on the context.

[00123] Although some of various drawings illustrate a number of logical stages in a particular order, stages that are not order dependent may be reordered and other stages may be combined or broken out. While some reordering or other groupings are specifically mentioned, others will be obvious to those of ordinary skill in the art, so the ordering and groupings presented herein are not an exhaustive list of alternatives. Moreover, it should be recognized that the stages could be implemented in hardware, firmware, software or any combination thereof.

[00124] The foregoing description, for purpose of explanation, has been described with reference to specific implementations. However, the illustrative discussions above are not intended to be exhaustive or to limit the scope of the claims to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The implementations were chosen in order to best explain the principles underlying the claims and their practical applications, to thereby enable others skilled in the art to best use the implementations with various modifications as are suited to the particular uses contemplated.

What is claimed is:

1. A method for calculating a position of a first object relative to a second, the method comprising:

at a first object including a controller, a wireless transceiver, and a first plurality of inertial measurement units (IMUs) each mounted in one or more positions and orientations relative to other of the first plurality of IMUs, wherein the first object is configured to:

receive a first object initial absolute position for the first plurality of IMUs or the controller;

sense, using the first plurality of IMUs, motion of the first object and generate sensed motion data of the first object;

generate, using the controller, a motion signal representative of the motion of the first object, wherein the motion signal is generated by calculating a rectified data output based on sensed motion data from each of the first plurality of IMUs, a predetermined position of each of the first plurality of IMUs and a predetermined orientation of each of the first plurality of IMUs;

calculate, using the controller, a first object current absolute position using the motion signal generated by the controller and the first object initial absolute position;

receive, using the wireless transceiver, referential data from a second object, the referential data including an second object current absolute position calculated using a second plurality of IMUs associated with the second object; and

calculate a relative position of the first object relative to the second object using the first object current absolute position and the second object current absolute position, wherein the relative position includes at least one of: (i) a distance between the first object and the second object and (ii) an orientation of the first object relative to the second object.

2. The method of claim 1, wherein the referential data includes a third object current absolute position of a third object calculated using a third plurality of IMUs associated with the third object, wherein the first object is configured to:

calculate a relative position of the first object relative to the third object using the first object current absolute position and the third object current absolute position, wherein the relative position includes at least one of: (i) a distance between the first object and the third object and (ii) an orientation of the first object relative to the third object.

3. The method of any of claims 1-2,
wherein the first object is configured to:

transmit, using the wireless transceiver at the first object, the first object current absolute position of the first object to the second object;

wherein the second object is configured to:

receive, using a wireless transceiver at the second object, the first object current absolute position of the first object; and

calculate, using a controller at the second object, a relative position of the second object relative to the first object using the first object current absolute position and the second object current absolute position, wherein the relative position includes at least one of: (i) a distance between the second object and the first object and (ii) an orientation of the second object to the first object.

4. The method of any of claims 1-3, wherein the first plurality of IMUs generates the motion signal using at least one of: shape correction, static calibration, motion decomposition, dynamic calibration, motion synthesis, and edge condition smoothing.

5. The method of any of claims 1-4, wherein the first plurality of IMUs includes an accelerometer or a gyroscope.

6. The method of any of claims 1-5, wherein the first object current absolute position and the second object current absolute position are calculated without an external reference signal.

7. The method of any of claims 1-6, wherein the first object is a first car and the second object is a second car, wherein the first object is configured to:

after calculating a relative position of the first car relative to the second car, determine whether the relative position of the first car to the second car meets emergency criteria; and

in response to determining that the relative position of the first car to the second car meets emergency criteria, cause the first car to perform an evasive maneuver, wherein the evasive maneuver includes braking the first car or turning the first car.

8. The method of any of claims 1-7, wherein the first object is configured to:

display, at a user interface associated with the first object, a position of the first object on a graphical representation of a map using the relative position of the first object relative to the second object.

9. The method of any of claims 1-6, wherein the first object is a home appliance and the second object is a car, wherein the home appliance is configured to:

after calculating a relative position of the car relative to the home appliance, determine whether the relative position of the car to the home appliance meets operational state change criteria; and

in response to determining that the relative position of the car to the home appliance meets operational state change criteria, cause the home appliance to change from an off state to an on state.

10. A system for calculating a position of a first object relative to a second, the system comprising:

a first object including a controller, a wireless transceiver, and a first plurality of inertial measurement units (IMUs), wherein the first object is configured to perform the steps of any of claims 1-9.

11. A non-transitory_computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first object including a controller, a wireless transceiver, and a first plurality of inertial measurement units (IMUs), cause the first object to perform the steps of any of claims 1-9.

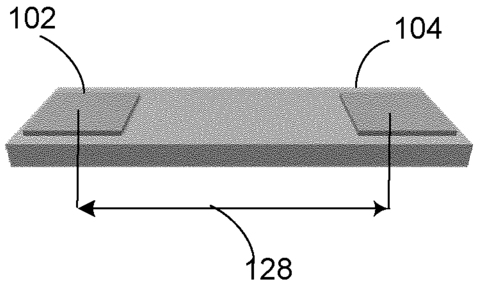


Figure 1A

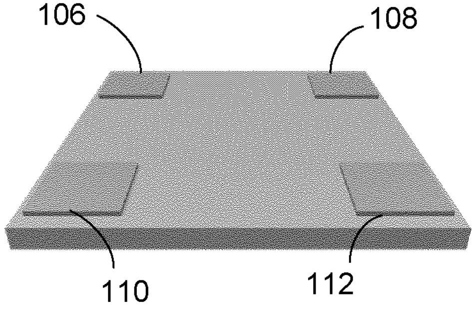


Figure 1B

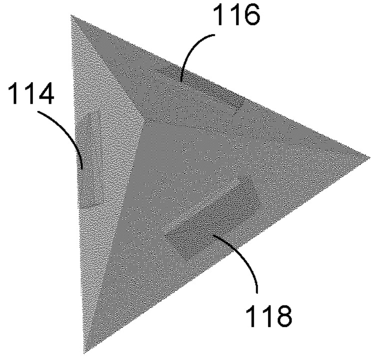


Figure 1C

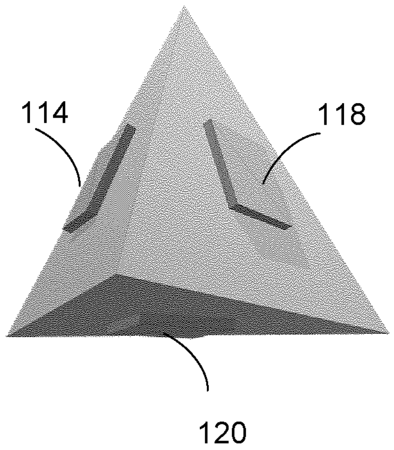


Figure 1D

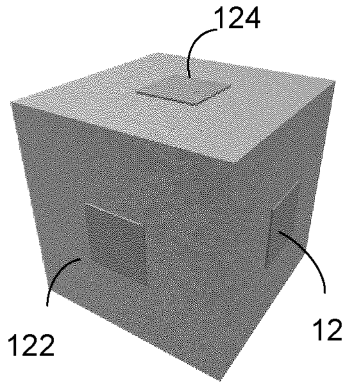


Figure 1E

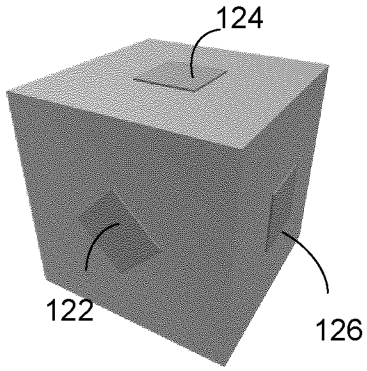


Figure 1F

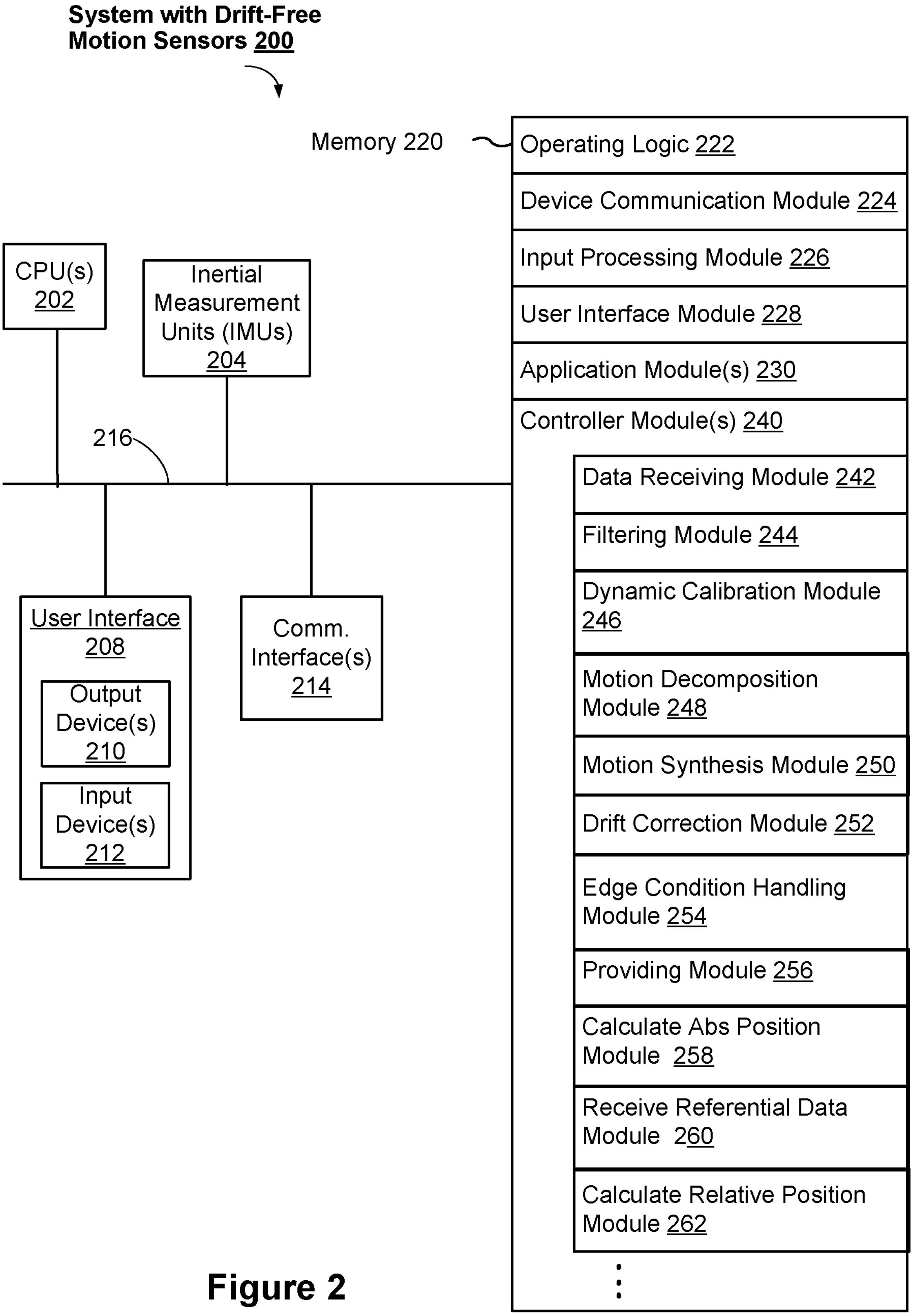

System with Drift-Free Motion Sensors 200
Memory 220
Operating Logic 222
Device Communication Module 224
Input Processing Module 226
User Interface Module 228
Application Module(s) 230
Controller Module(s) 240
Data Receiving Module 242
Filtering Module 244
Dynamic Calibration Module 246
Motion Decomposition Module 248
Motion Synthesis Module 250
Drift Correction Module 252
Edge Condition Handling Module 254
Providing Module 256
Calculate Abs Position Module 258
Receive Referential Data Module 260
Calculate Relative Position Module 262
CPU(s) 202
Inertial Measurement Units (IMUs) 204
216
User Interface 208
Output Device(s) 210
Input Device(s) 212
Comm. Interface(s) 214

Figure 2

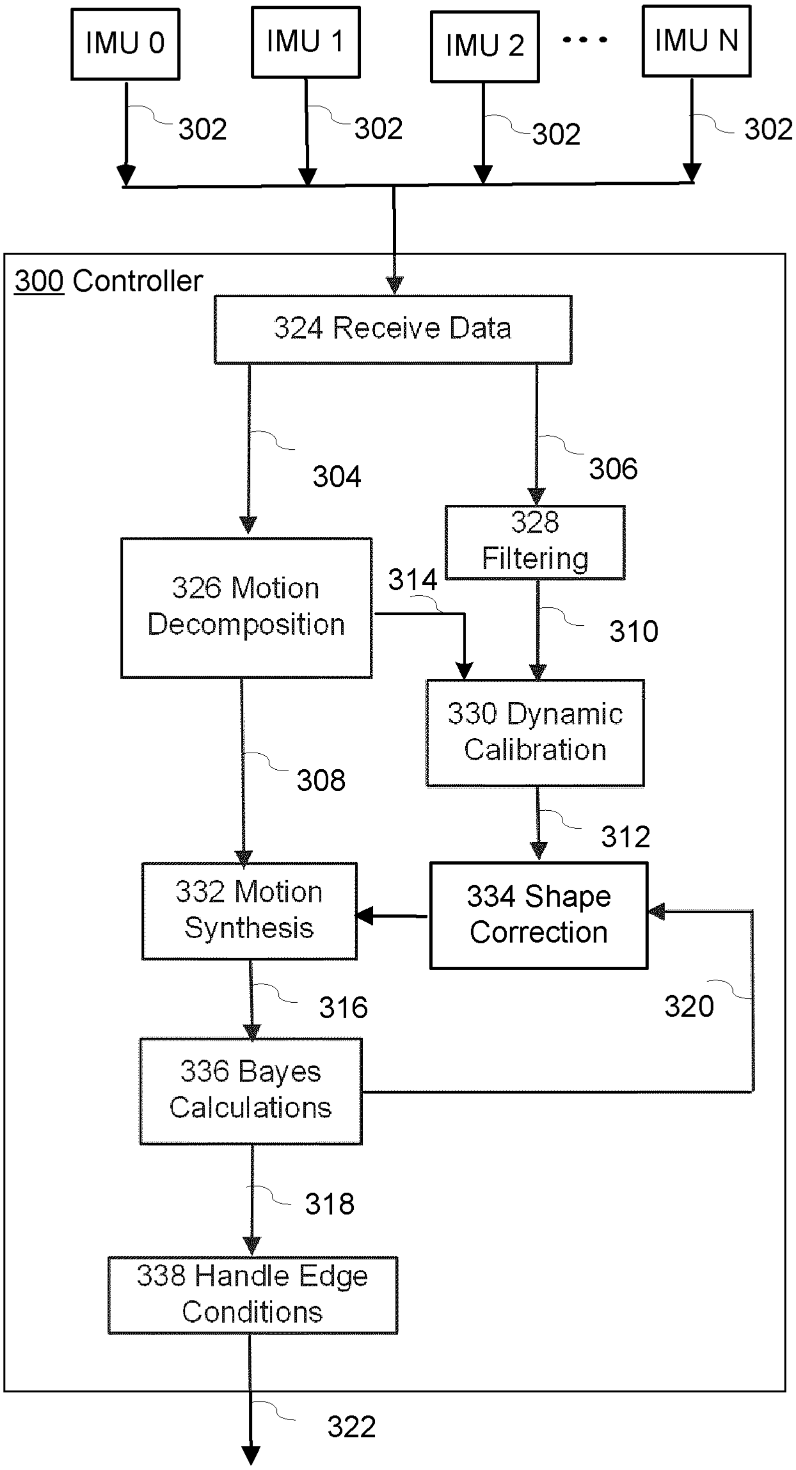

IMU 0
IMU 1
IMU 2
• • •
IMU N
302
302
302
302
300 Controller
324 Receive Data
304
306
328 Filtering
326 Motion Decomposition
314
310
330 Dynamic Calibration
308
312
332 Motion Synthesis
334 Shape Correction
316
320
336 Bayes Calculations
318
338 Handle Edge Conditions
322

Figure 3

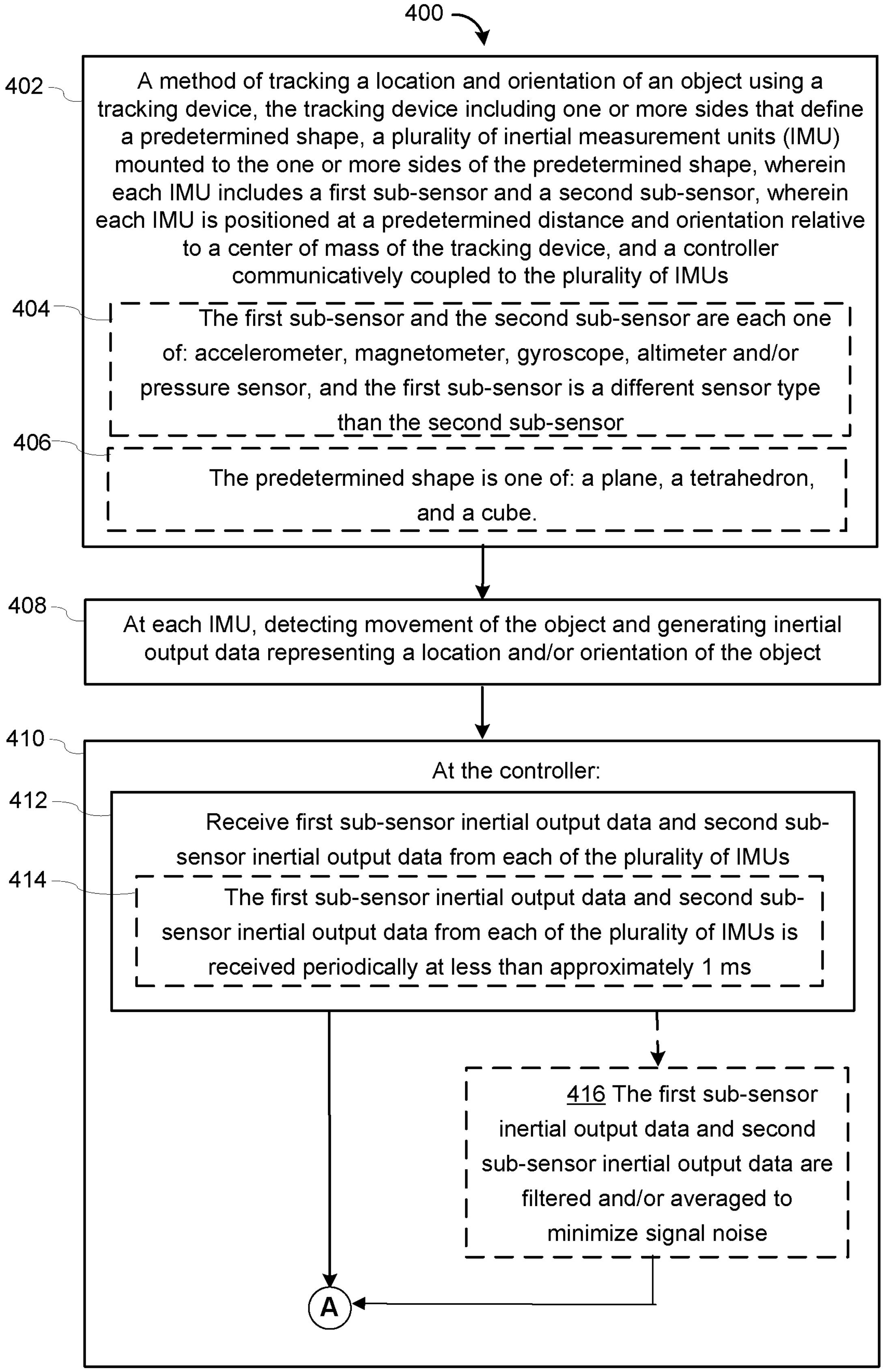

400
402
A method of tracking a location and orientation of an object using a tracking device, the tracking device including one or more sides that define a predetermined shape, a plurality of inertial measurement units (IMU) mounted to the one or more sides of the predetermined shape, wherein each IMU includes a first sub-sensor and a second sub-sensor, wherein each IMU is positioned at a predetermined distance and orientation relative to a center of mass of the tracking device, and a controller communicatively coupled to the plurality of IMUs
404
The first sub-sensor and the second sub-sensor are each one of: accelerometer, magnetometer, gyroscope, altimeter and/or pressure sensor, and the first sub-sensor is a different sensor type than the second sub-sensor
406
The predetermined shape is one of: a plane, a tetrahedron, and a cube.
408
At each IMU, detecting movement of the object and generating inertial output data representing a location and/or orientation of the object
410
At the controller:
412
Receive first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs
414
The first sub-sensor inertial output data and second sub-sensor inertial output data from each of the plurality of IMUs is received periodically at less than approximately 1 ms
416 The first sub-sensor inertial output data and second sub-sensor inertial output data are filtered and/or averaged to minimize signal noise
A

Figure 4A

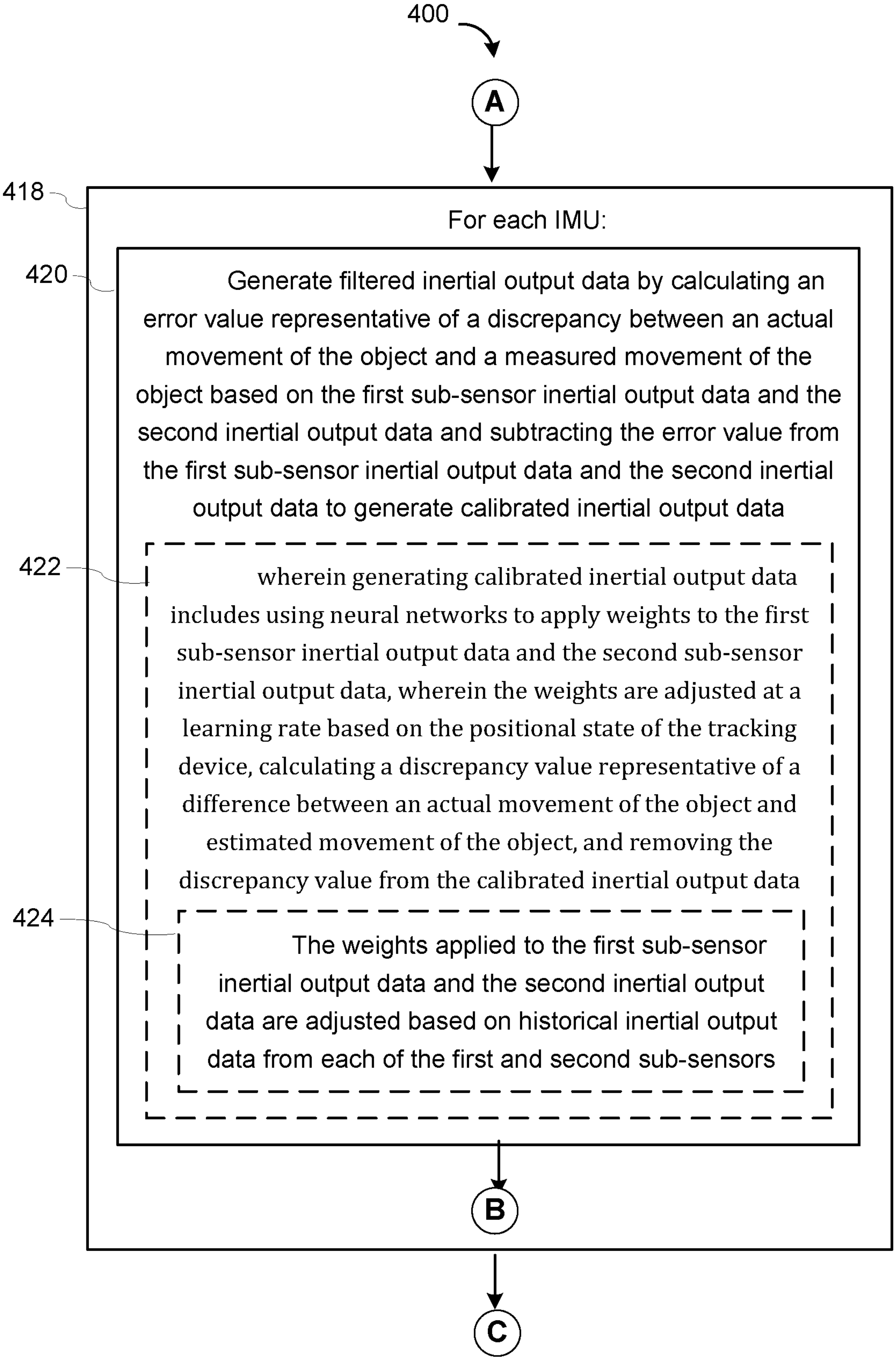
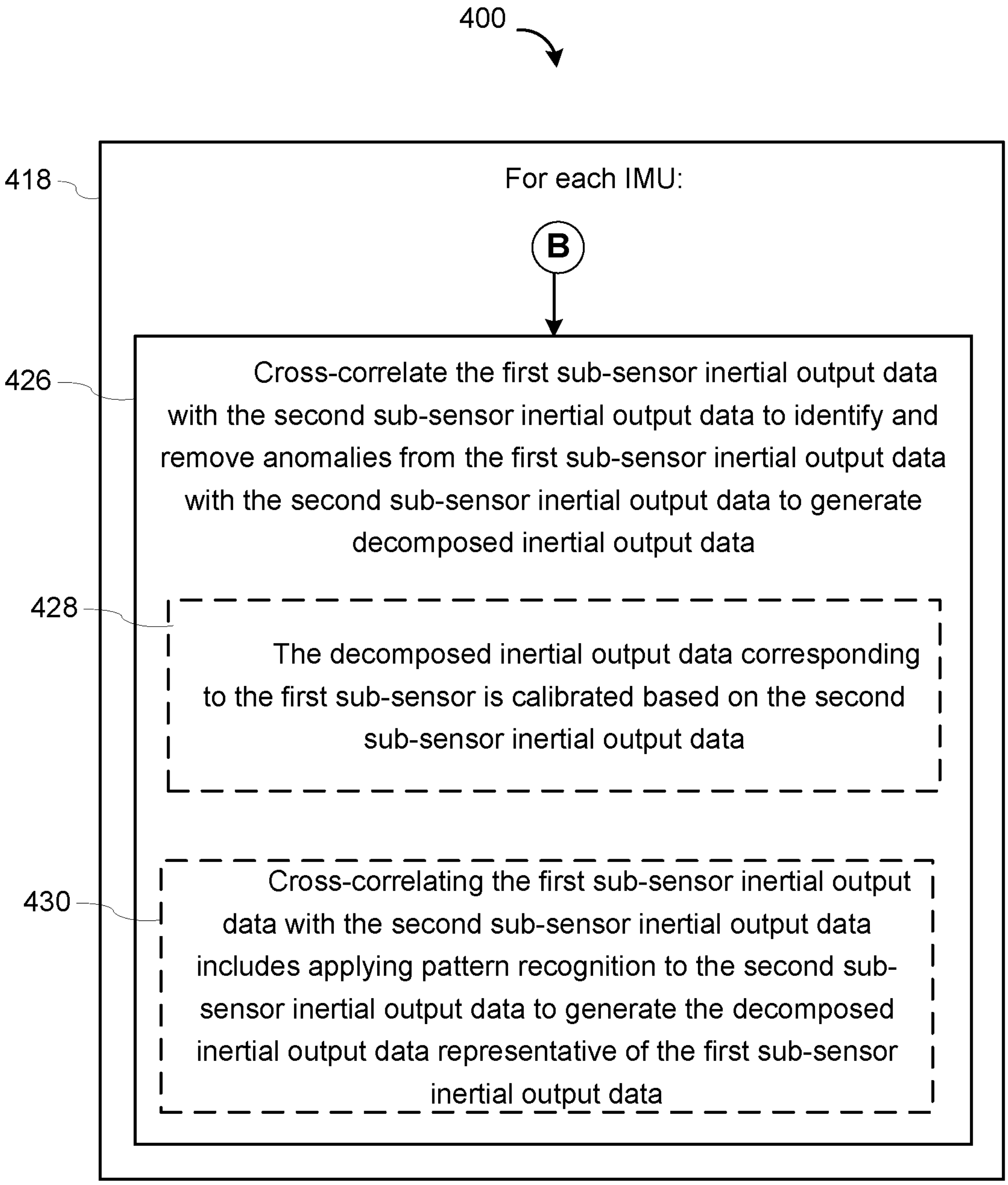

400
A
418
For each IMU:
420
Generate filtered inertial output data by calculating an error value representative of a discrepancy between an actual movement of the object and a measured movement of the object based on the first sub-sensor inertial output data and the second inertial output data and subtracting the error value from the first sub-sensor inertial output data and the second inertial output data to generate calibrated inertial output data
422
wherein generating calibrated inertial output data includes using neural networks to apply weights to the first sub-sensor inertial output data and the second sub-sensor inertial output data, wherein the weights are adjusted at a learning rate based on the positional state of the tracking device, calculating a discrepancy value representative of a difference between an actual movement of the object and estimated movement of the object, and removing the discrepancy value from the calibrated inertial output data
424
The weights applied to the first sub-sensor inertial output data and the second inertial output data are adjusted based on historical inertial output data from each of the first and second sub-sensors
B
C

Figure 4B



Cross-correlate the first sub-sensor inertial output data with the second sub-sensor inertial output data to identify and remove anomalies from the first sub-sensor inertial output data with the second sub-sensor inertial output data to generate decomposed inertial output data

The decomposed inertial output data corresponding to the first sub-sensor is calibrated based on the second sub-sensor inertial output data

Cross-correlating the first sub-sensor inertial output data with the second sub-sensor inertial output data includes applying pattern recognition to the second sub-sensor inertial output data to generate the decomposed inertial output data representative of the first sub-sensor inertial output data

Figure 4C

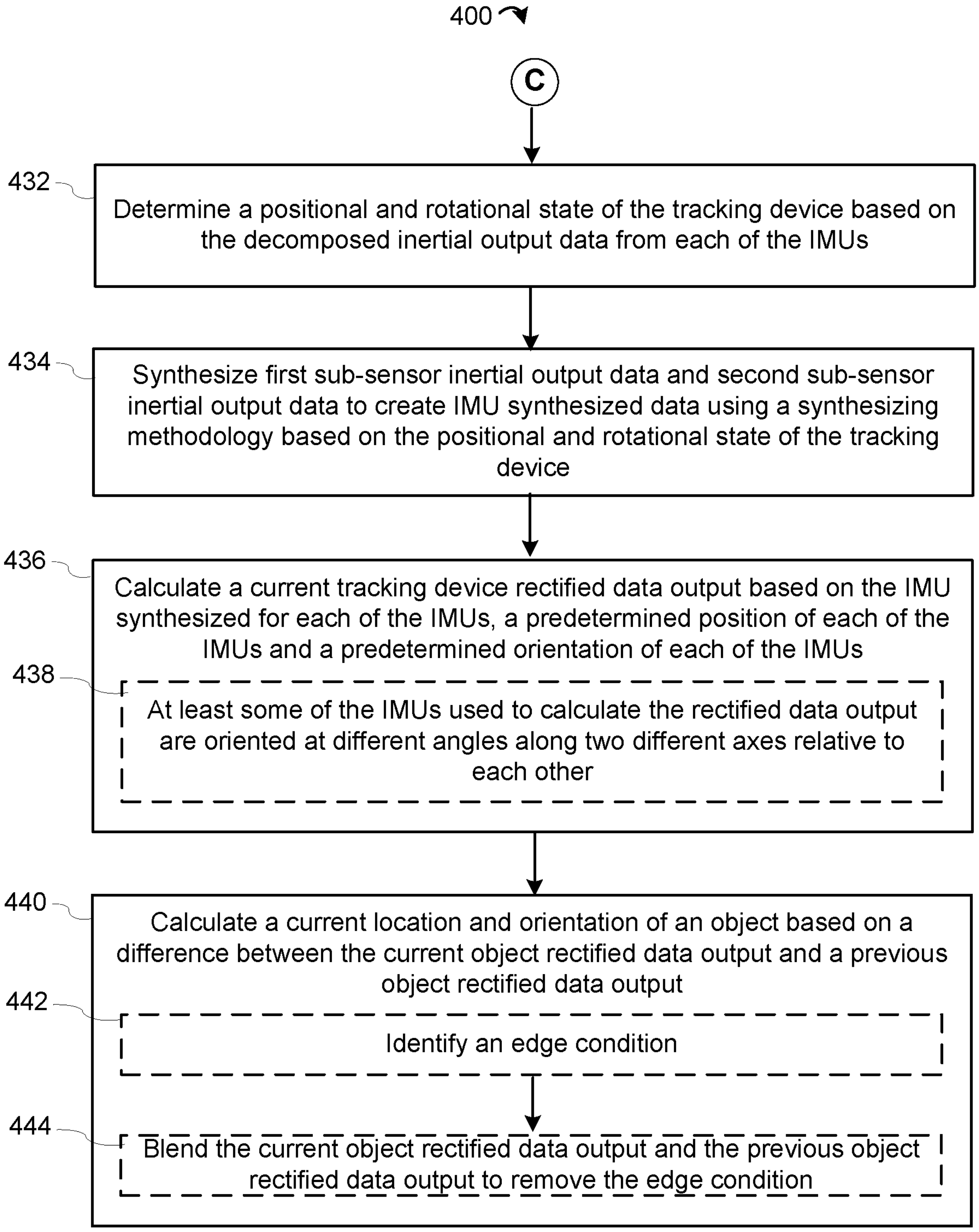

400
C
432
Determine a positional and rotational state of the tracking device based on the decomposed inertial output data from each of the IMUs
434
Synthesize first sub-sensor inertial output data and second sub-sensor inertial output data to create IMU synthesized data using a synthesizing methodology based on the positional and rotational state of the tracking device
436
Calculate a current tracking device rectified data output based on the IMU synthesized for each of the IMUs, a predetermined position of each of the IMUs and a predetermined orientation of each of the IMUs
438
At least some of the IMUs used to calculate the rectified data output are oriented at different angles along two different axes relative to each other
440
Calculate a current location and orientation of an object based on a difference between the current object rectified data output and a previous object rectified data output
442
Identify an edge condition
444
Blend the current object rectified data output and the previous object rectified data output to remove the edge condition

Figure 4D

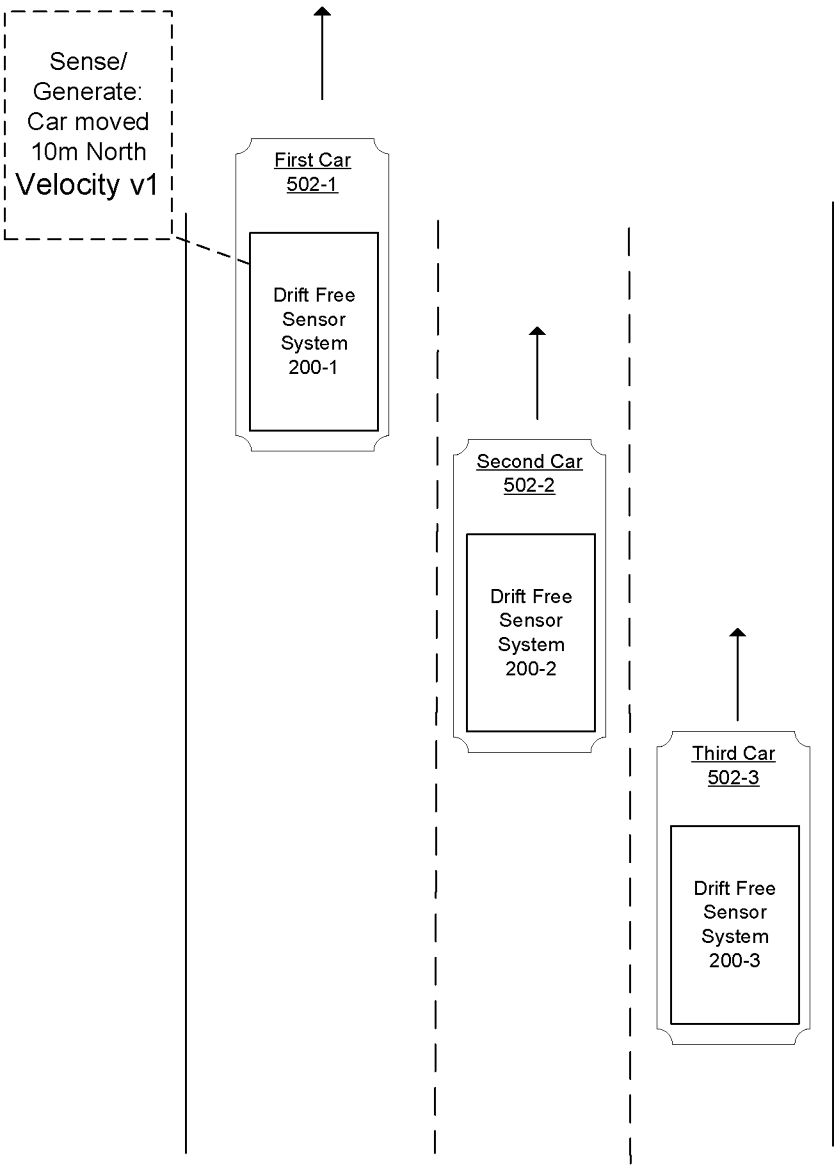

Sense/
Generate:
Car moved
10m North
Velocity v1
First Car
502-1
Drift Free
Sensor
System
200-1
Second Car
502-2
Drift Free
Sensor
System
200-2
Third Car
502-3
Drift Free
Sensor
System
200-3

Figure 5A

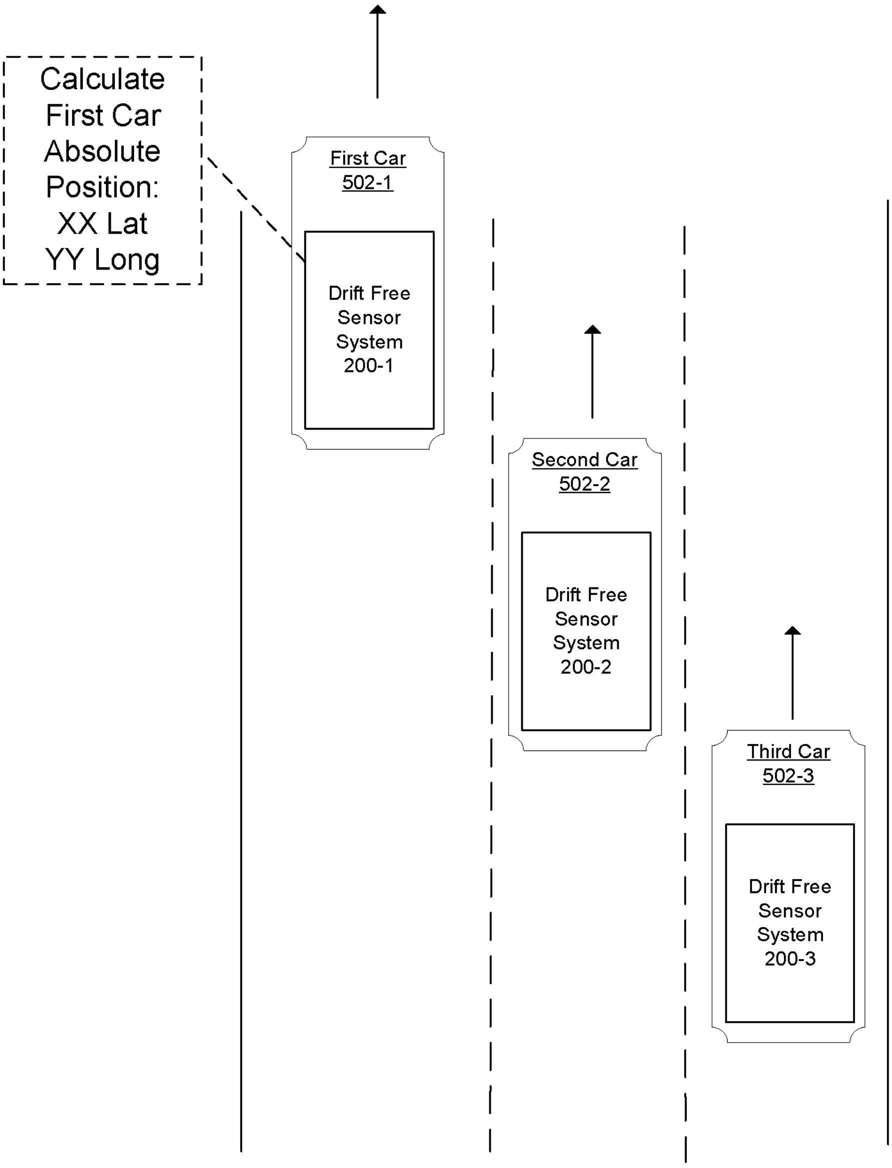

Calculate
First Car
Absolute
Position:
XX Lat
YY Long
First Car
502-1
Drift Free
Sensor
System
200-1
Second Car
502-2
Drift Free
Sensor
System
200-2
Third Car
502-3
Drift Free
Sensor
System
200-3

Figure 5B

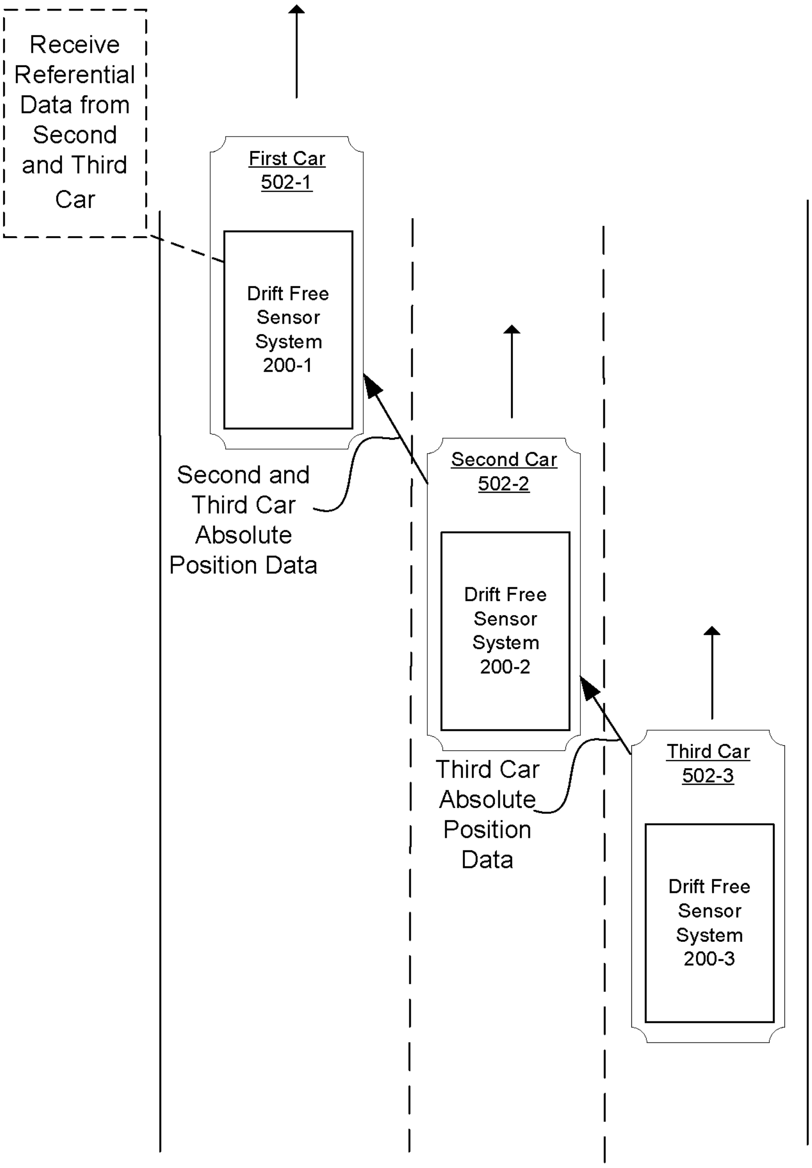

Receive Referential Data from Second and Third Car
First Car 502-1
Drift Free Sensor System 200-1
Second and Third Car Absolute Position Data
Second Car 502-2
Drift Free Sensor System 200-2
Third Car Absolute Position Data
Third Car 502-3
Drift Free Sensor System 200-3

Figure 5C

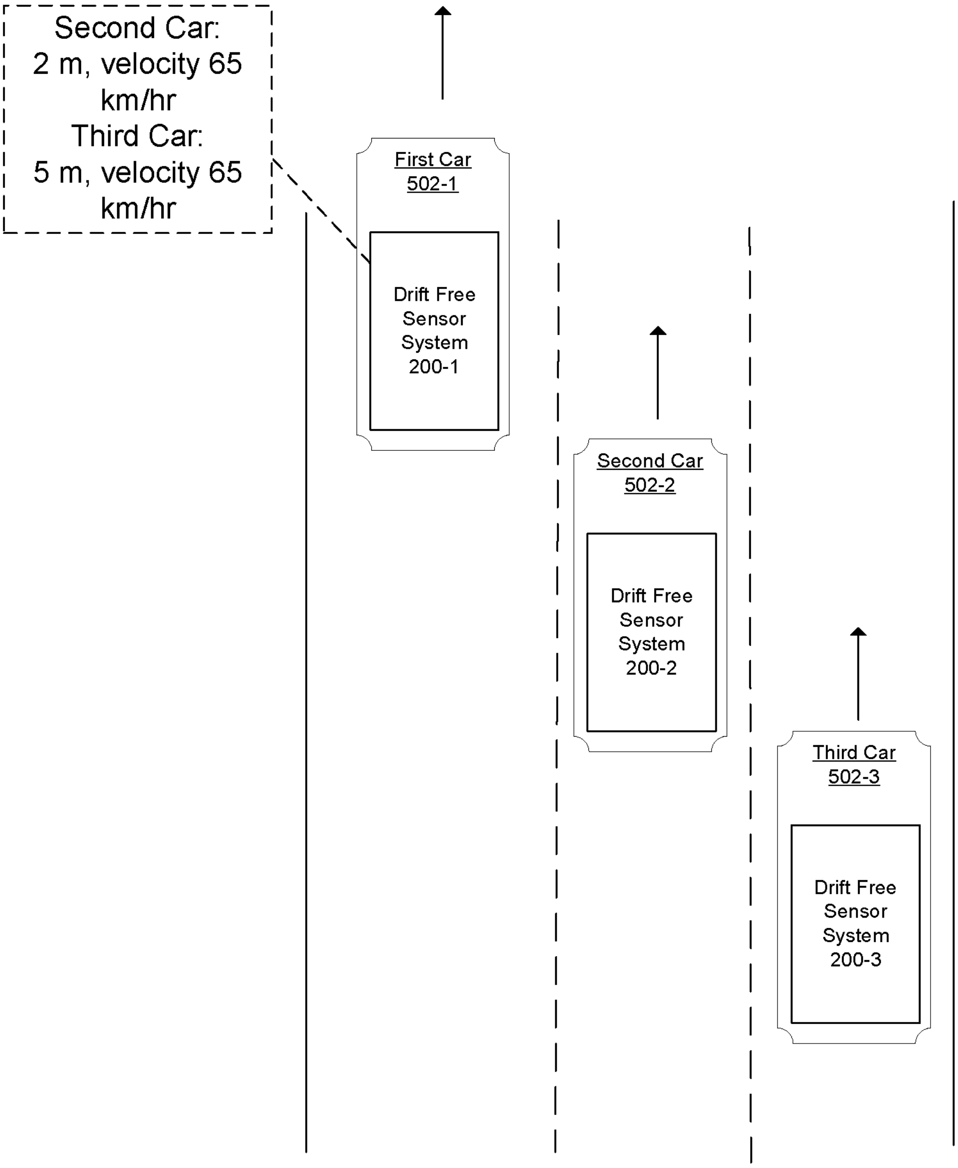

Second Car:
2 m, velocity 65 km/hr
Third Car:
5 m, velocity 65 km/hr
First Car 502-1
Drift Free Sensor System 200-1
Second Car 502-2
Drift Free Sensor System 200-2
Third Car 502-3
Drift Free Sensor System 200-3

Figure 5D

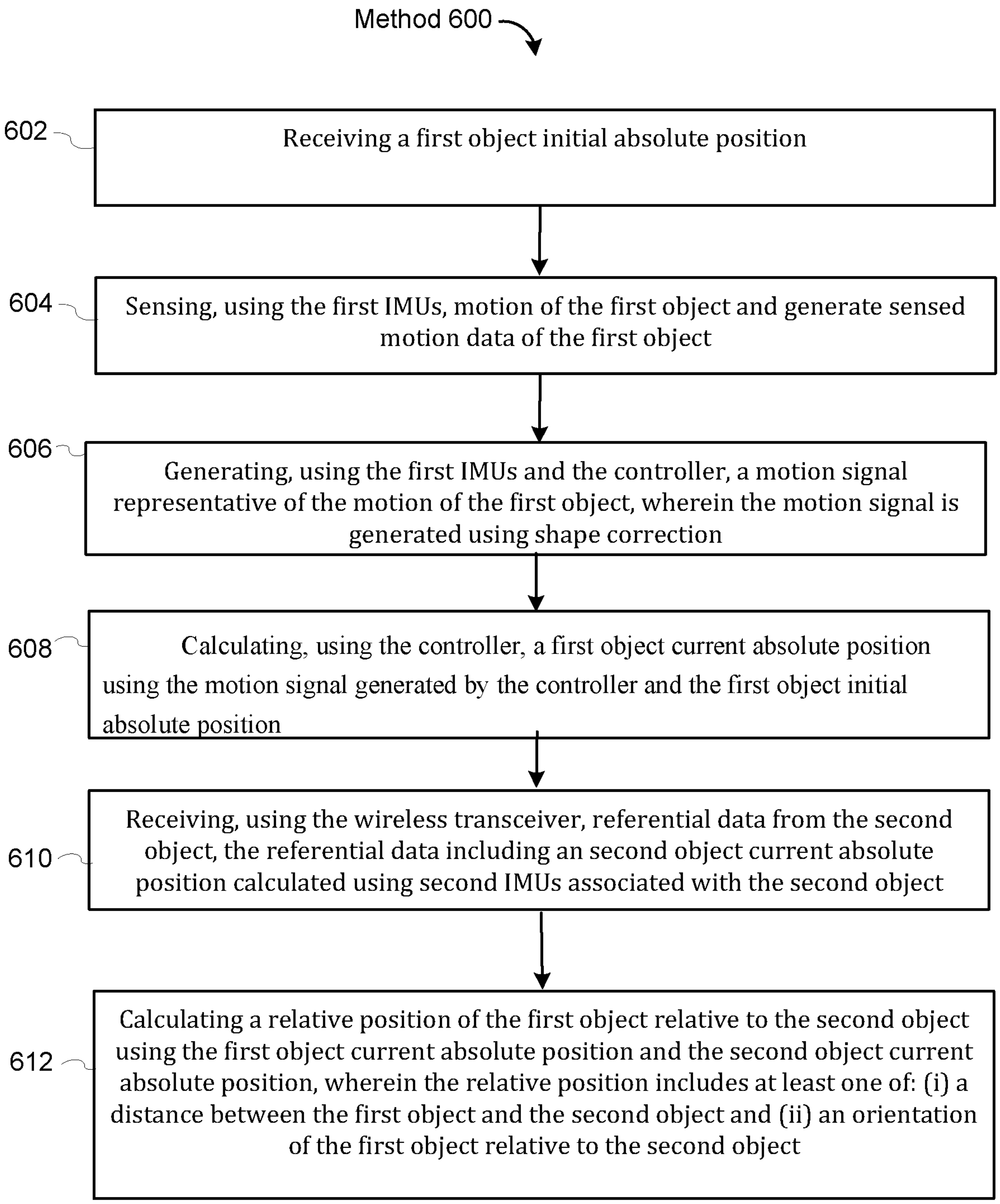

Method 600
602
Receiving a first object initial absolute position
604
Sensing, using the first IMUs, motion of the first object and generate sensed motion data of the first object
606
Generating, using the first IMUs and the controller, a motion signal representative of the motion of the first object, wherein the motion signal is generated using shape correction
608
Calculating, using the controller, a first object current absolute position using the motion signal generated by the controller and the first object initial absolute position
610
Receiving, using the wireless transceiver, referential data from the second object, the referential data including an second object current absolute position calculated using second IMUs associated with the second object
612
Calculating a relative position of the first object relative to the second object using the first object current absolute position and the second object current absolute position, wherein the relative position includes at least one of: (i) a distance between the first object and the second object and (ii) an orientation of the first object relative to the second object

Figure 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2020/050838

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G01C 21/16*** (2006.01) , ***B60W 30/09*** (2012.01) , ***G08G 1/16*** (2006.01) , ***H04W 4/40*** (2018.01)

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)
IPC: G01C 21/16 (2006.01) , B60W 30/09 (2012.01) , G08G 1/16 (2006.01) , H04W 4/40 (2018.01)

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Database: (Keywords: IMUs, absolute position, relative position, motion sensor, motion signal, drift correction, drift, vehicle, car, calculating a position, motion tracking, second object, smart city, autonomous vehicle, distance, relative distance)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014/0168009 (Peake) 19 June 2014 (19-06-2014) (abstract)	1-11
A	US 2007/0005609 (Breed) 4 January 2007 (04-01-2007) (abstract, paragraph [0217], [0218])	1-11

☐ 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
"D"	document cited by the applicant in the international application
"E"	earlier application or patent but published on or after the international filing date
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O"	document referring to an oral disclosure, use, exhibition or other means
"P"	document published prior to the international filing date but later than the priority date claimed
"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&"	document member of the same patent family

Date of the actual completion of the international search
26 August 2020 (26-08-2020)

Date of mailing of the international search report
26 August 2020 (26-08-2020)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 819-953-2476

Authorized officer

Camran Syed (819) 635-5801

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2014168009A1	19 June 2014 (19-06-2014)	US2014168009A1	19 June 2014 (19-06-2014)
		EP2932194A1	21 October 2015 (21-10-2015)
		EP2932194B1	06 March 2019 (06-03-2019)
		WO2014099403A1	26 June 2014 (26-06-2014)
US2007005609A1	04 January 2007 (04-01-2007)	US2007005609A1	04 January 2007 (04-01-2007)
		US7629899B2	08 December 2009 (08-12-2009)
		AU1526299A	07 June 1999 (07-06-1999)
		AU1820200A	05 June 2000 (05-06-2000)
		AU1958801A	25 June 2001 (25-06-2001)
		AU2323692A	11 February 1993 (11-02-1993)
		AU3522500A	28 September 2000 (28-09-2000)
		AU5964001A	20 November 2001 (20-11-2001)
		AU2003237024A1	19 January 2004 (19-01-2004)
		AU2003237024A8	19 January 2004 (19-01-2004)
		CN1561640A	05 January 2005 (05-01-2005)
		CN1666232A	07 September 2005 (07-09-2005)
		CN1666232B	26 May 2010 (26-05-2010)
		CN105807350A	27 July 2016 (27-07-2016)
		DE4447960B4	27 September 2007 (27-09-2007)
		DE4492128C2	02 January 2003 (02-01-2003)
		DE4492128T1	27 June 1996 (27-06-1996)
		DE10084638T1	02 May 2002 (02-05-2002)
		DE10149206A1	06 February 2003 (06-02-2003)
		DE19882381T1	13 July 2000 (13-07-2000)
		DE19983715T1	07 March 2002 (07-03-2002)
		DE69828585D1	17 February 2005 (17-02-2005)
		DE69828585T2	19 January 2006 (19-01-2006)
		EP0952933A1	03 November 1999 (03-11-1999)
		EP0952933A4	02 February 2000 (02-02-2000)
		EP0952933B1	12 January 2005 (12-01-2005)
		EP1069000A1	17 January 2001 (17-01-2001)
		EP1433326A1	30 June 2004 (30-06-2004)
		EP1520258A2	06 April 2005 (06-04-2005)
		GB0111362D0	04 July 2001 (04-07-2001)
		GB2360097A	12 September 2001 (12-09-2001)
		GB2360097B	23 April 2003 (23-04-2003)
		GB0116062D0	22 August 2001 (22-08-2001)
		GB2363638A	02 January 2002 (02-01-2002)
		GB2363638B	10 March 2004 (10-03-2004)
		GB0117003D0	05 September 2001 (05-09-2001)
		GB2363769A	09 January 2002 (09-01-2002)
		GB2363769B	18 June 2003 (18-06-2003)
		GB0123688D0	21 November 2001 (21-11-2001)
		GB2373117A	11 September 2002 (11-09-2002)
		GB2373117B	16 February 2005 (16-02-2005)
		GB0200235D0	20 February 2002 (20-02-2002)
		GB2369737A	05 June 2002 (05-06-2002)
		GB2369737B	02 February 2005 (02-02-2005)
		GB0307874D0	14 May 2003 (14-05-2003)
		GB2383415A	25 June 2003 (25-06-2003)
		GB2383415B	23 February 2005 (23-02-2005)
		GB0422015D0	03 November 2004 (03-11-2004)
		GB2405279A	23 February 2005 (23-02-2005)
		GB2405279B	27 April 2005 (27-04-2005)
		GB0426923D0	12 January 2005 (12-01-2005)
		GB2406170A	23 March 2005 (23-03-2005)

GB2406170B	04 May 2005 (04-05-2005)
GB0426924D0	12 January 2005 (12-01-2005)
GB2406646A	06 April 2005 (06-04-2005)
GB2406646B	18 May 2005 (18-05-2005)
GB0426925D0	12 January 2005 (12-01-2005)
GB2405931A	16 March 2005 (16-03-2005)
GB2405931B	27 April 2005 (27-04-2005)
GB0426926D0	12 January 2005 (12-01-2005)
GB2406171A	23 March 2005 (23-03-2005)
GB2406171B	04 May 2005 (04-05-2005)
GB9509090D0	02 August 1995 (02-08-1995)
GB2289332A	15 November 1995 (15-11-1995)
GB2289332B	06 January 1999 (06-01-1999)
GB9509088D0	02 August 1995 (02-08-1995)
GB2289542A	22 November 1995 (22-11-1995)
GB2289542B	26 August 1998 (26-08-1998)
GB9510402D0	19 July 1995 (19-07-1995)
GB2289653A	29 November 1995 (29-11-1995)
GB2289653B	26 August 1998 (26-08-1998)
GB9510408D0	19 July 1995 (19-07-1995)
GB2289786A	29 November 1995 (29-11-1995)
GB2289786B	16 September 1998 (16-09-1998)
GB9611795D0	07 August 1996 (07-08-1996)
GB2301906A	18 December 1996 (18-12-1996)
GB2301906B	29 September 1999 (29-09-1999)
GB9611810D0	07 August 1996 (07-08-1996)
GB2301922A	18 December 1996 (18-12-1996)
GB2301922B	22 December 1999 (22-12-1999)
GB9624734D0	15 January 1997 (15-01-1997)
GB2308102A	18 June 1997 (18-06-1997)
GB2308102B	10 November 1999 (10-11-1999)
GB9802263D0	01 April 1998 (01-04-1998)
GB2323340A	23 September 1998 (23-09-1998)
GB2323340B	18 July 2001 (18-07-2001)
GB9816593D0	30 September 1998 (30-09-1998)
GB2324864A	04 November 1998 (04-11-1998)
GB2324864B	06 January 1999 (06-01-1999)
GB9927633D0	19 January 2000 (19-01-2000)
GB2340978A	01 March 2000 (01-03-2000)
GB2340978B	27 December 2000 (27-12-2000)
JPH08175305A	09 July 1996 (09-07-1996)
JP3960394B2	15 August 2007 (15-08-2007)
JPH08198044A	06 August 1996 (06-08-1996)
JP3993253B2	17 October 2007 (17-10-2007)
JP2005531845A	20 October 2005 (20-10-2005)
JP4936662B2	23 May 2012 (23-05-2012)
JP2011081823A	21 April 2011 (21-04-2011)
JP5143212B2	13 February 2013 (13-02-2013)
JPH09175316A	08 July 1997 (08-07-1997)
JPH09189710A	22 July 1997 (22-07-1997)
JPH09240407A	16 September 1997 (16-09-1997)
JPH09501120A	04 February 1997 (04-02-1997)
JP2001508732A	03 July 2001 (03-07-2001)
JP2002501459A	15 January 2002 (15-01-2002)
JP2003532959A	05 November 2003 (05-11-2003)
JP2004518104A	17 June 2004 (17-06-2004)
JP2004522932A	29 July 2004 (29-07-2004)
JP2005306376A	04 November 2005 (04-11-2005)
JP2005505209A	17 February 2005 (17-02-2005)

JP2016071366A	09 May 2016 (09-05-2016)		
		KR20040037145A	04 May 2004 (04-05-2004)
		SE0100186D0	24 January 2001 (24-01-2001)
		SE0100186L	27 March 2001 (27-03-2001)
		SE523753C2	11 May 2004 (11-05-2004)
		TW541255B	11 July 2003 (11-07-2003)
		TW201612560A	01 April 2016 (01-04-2016)
		TWI626479B	11 June 2018 (11-06-2018)
		USRE37260E	03 July 2001 (03-07-2001)
		USRE37736E	11 June 2002 (11-06-2002)
		USRE39868E	09 October 2007 (09-10-2007)
		USRE40073E	19 February 2008 (19-02-2008)
		US5155307A	13 October 1992 (13-10-1992)
		US5231253A	27 July 1993 (27-07-1993)
		US5233141A	03 August 1993 (03-08-1993)
		US5326133A	05 July 1994 (05-07-1994)
		US5389751A	14 February 1995 (14-02-1995)
		US5441301A	15 August 1995 (15-08-1995)
		US5505485A	09 April 1996 (09-04-1996)
		US5629681A	13 May 1997 (13-05-1997)
		US5653462A	05 August 1997 (05-08-1997)
		US5653464A	05 August 1997 (05-08-1997)
		US5684701A	04 November 1997 (04-11-1997)
		US5694320A	02 December 1997 (02-12-1997)
		US5746446A	05 May 1998 (05-05-1998)
		US5748473A	05 May 1998 (05-05-1998)
		US5772238A	30 June 1998 (30-06-1998)
		US5809437A	15 September 1998 (15-09-1998)
		US5822707A	13 October 1998 (13-10-1998)
		US5829782A	03 November 1998 (03-11-1998)
		US5835613A	10 November 1998 (10-11-1998)
		US5842716A	01 December 1998 (01-12-1998)
		US5845000A	01 December 1998 (01-12-1998)
		US5848802A	15 December 1998 (15-12-1998)
		US5863068A	26 January 1999 (26-01-1999)
		US5901978A	11 May 1999 (11-05-1999)
		US5943295A	24 August 1999 (24-08-1999)
		US6009970A	04 January 2000 (04-01-2000)
		US6039139A	21 March 2000 (21-03-2000)
		US6078854A	20 June 2000 (20-06-2000)
		US6081757A	27 June 2000 (27-06-2000)
		US6088640A	11 July 2000 (11-07-2000)
		US6116639A	12 September 2000 (12-09-2000)
		US6134492A	17 October 2000 (17-10-2000)
		US6141432A	31 October 2000 (31-10-2000)
		US6149194A	21 November 2000 (21-11-2000)
		US6168198B1	02 January 2001 (02-01-2001)
		US6175787B1	16 January 2001 (16-01-2001)
		US6179326B1	30 January 2001 (30-01-2001)
		US6186537B1	13 February 2001 (13-02-2001)
		US6206129B1	27 March 2001 (27-03-2001)
		US6209909B1	03 April 2001 (03-04-2001)
		US6234519B1	22 May 2001 (22-05-2001)
		US6234520B1	22 May 2001 (22-05-2001)
		US6242701B1	05 June 2001 (05-06-2001)
		US6250668B1	26 June 2001 (26-06-2001)

US6253134B1	26 June 2001 (26-06-2001)
US6254127B1	03 July 2001 (03-07-2001)
US6270116B1	07 August 2001 (07-08-2001)
US6279946B1	28 August 2001 (28-08-2001)
US6283503B1	04 September 2001 (04-09-2001)
US6324453B1	27 November 2001 (27-11-2001)
US2001015547A1	23 August 2001 (23-08-2001)
US6325414B2	04 December 2001 (04-12-2001)
US6326704B1	04 December 2001 (04-12-2001)
US2001000886A1	10 May 2001 (10-05-2001)
US6328126B2	11 December 2001 (11-12-2001)
US6330501B1	11 December 2001 (11-12-2001)
US6343810B1	05 February 2002 (05-02-2002)
US6370475B1	09 April 2002 (09-04-2002)
US6393133B1	21 May 2002 (21-05-2002)
US6397136B1	28 May 2002 (28-05-2002)
US6405132B1	11 June 2002 (11-06-2002)
US6412813B1	02 July 2002 (02-07-2002)
US6419265B1	16 July 2002 (16-07-2002)
US6422595B1	23 July 2002 (23-07-2002)
US2001029416A1	11 October 2001 (11-10-2001)
US6442465B2	27 August 2002 (27-08-2002)
US6442504B1	27 August 2002 (27-08-2002)
US2002059022A1	16 May 2002 (16-05-2002)
US6445988B1	03 September 2002 (03-09-2002)
US6452870B1	17 September 2002 (17-09-2002)
US2002082756A1	27 June 2002 (27-06-2002)
US6459973B1	01 October 2002 (01-10-2002)
US6474683B1	05 November 2002 (05-11-2002)
US2001002451A1	31 May 2001 (31-05-2001)
US6484080B2	19 November 2002 (19-11-2002
US2002029103A1	07 March 2002 (07-03-2002)
US6507779B2	14 January 2003 (14-01-2003)
US2001024032A1	27 September 2001 (27-09-2001)
US6513830B2	04 February 2003 (04-02-2003)
US2001042976A1	22 November 2001 (22-11-2001)
US6513833B2	04 February 2003 (04-02-2003)
US2001020777A1	13 September 2001 (13-09-2001)
US6517107B2	11 February 2003 (11-02-2003)
US6526352B1	25 February 2003 (25-02-2003)
US6529809B1	04 March 2003 (04-03-2003)
US6532408B1	11 March 2003 (11-03-2003)
US2001015548A1	23 August 2001 (23-08-2001)
US6533316B2	18 March 2003 (18-03-2003)
US2001003168A1	07 June 2001 (07-06-2001)
US6553296B2	22 April 2003 (22-04-2003)
US2001054516A1	27 December 2001 (27-12-2001)
US6555766B2	29 April 2003 (29-04-2003)
US2001028163A1	11 October 2001 (11-10-2001)
US6557889B2	06 May 2003 (06-05-2003)