

COMPUTER-IMPLEMENTED METHOD AND DEVICE FOR MOBILE NODE TRACKING

Field of the Invention

The present invention relates to the field of object localisation and more particularly to a computer-implemented method and a device for mobile node tracking.

Background of the Invention

Multi-agent localization frameworks often utilize either an extended Kalman filter (EKF) or inertial measurement unit (IMU) dead-reckoning (DR) as a solution for tracking. EKF frameworks are usually based on a motion model that can assume certain characteristics regarding an object being tracked. Most models either take the form of constant velocity, constant acceleration or constant jerk. IMU as a DR solution is a common implementation, utilizing gyroscope heading and accelerometer speed to guide a prediction such as EKF in tracking individual node locations.

Classical EKF tracking, however, cannot effectively capture fine random walk movements typical of human motion without the presence of other sensor information and fusion techniques. Additionally, traditional distributed EKF solutions that rely purely on IMU will still diverge with a complexity of $O(n^2)$.

A significant issue with group localization is divergence from ground truth over time. This is due to drift in sensor measurements over time from an IMU or an external sensor. The inherent drift in IMUs over time causes a tracking solution that is solely based on DR IMU to diverge.

Another problem with multi-agent localization is growing complexity as a network increases and a need therefore for suitable processing and power resources for mobile nodes in a system.

There is therefore a need for a computer-implemented method and a device for mobile node tracking with reduced infrastructure requirements and that provide accurate group tracking with reduced divergence for extended trajectories.

Summary of the Invention

Accordingly, in a first aspect, the present invention provides a computer-implemented method for mobile node tracking. The method includes executing on one or more processors the steps of: predicting node positions of a plurality of mobile nodes;
5 receiving ultra-wideband (UWB) ranging data and instantaneous inertial measurement unit (IMU) data from each of the mobile nodes; updating the predicted node positions of the mobile nodes using the UWB ranging data and the instantaneous IMU data; and broadcasting the updated node positions of the mobile nodes from the server to the mobile nodes.

10 In a second aspect, the present invention provides a device for mobile node tracking. The device includes one or more processors and a non-transitory computer-readable memory storing computer program instructions executable by the one or more processors to perform operations for mobile node tracking. The operations include: predicting node positions of a plurality of mobile nodes; receiving ultra-wideband (UWB)
15 ranging data and instantaneous inertial measurement unit (IMU) data from each of the mobile nodes; updating the predicted node positions of the mobile nodes using the UWB ranging data and the instantaneous IMU data; and broadcasting the updated node positions of the mobile nodes to the mobile nodes.

Other aspects and advantages of the invention will become apparent from the
20 following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

25 FIG. 1 is a schematic flow diagram illustrating a computer-implemented method for mobile node tracking in accordance with an embodiment of the present invention;

FIG. 2 is a schematic block diagram illustrating a computer system suitable for implementing the method for mobile node tracking disclosed herein;

FIG. 3 is a graph of environment and ground truth trajectories for a first scenario;

FIG. 4A is a graph of estimated trajectories of mobile nodes in the first scenario obtained using dead-reckoning (DR);

FIG. 4B is a graph of estimated trajectories of the mobile nodes in the first
5 scenario obtained using the adaptive centralised-extended Kalman filter (AC-EKF) method in accordance with an embodiment of the present invention;

FIG. 5A is a zoomed-in trajectory slice for Node 5 in the first scenario obtained using DR estimation;

FIG. 5B is a zoomed-in trajectory slice for Node 5 in the first scenario obtained
10 using the AC-EKF method in accordance with an embodiment of the present invention;

FIG. 6A is a graph of cumulative distribution functions (CDFs) of estimation errors when the nodes in the first scenario are within starting anchor range;

FIG. 6B is a graph of CDFs of estimation errors when the nodes in the first scenario are out of anchor range;

FIG. 6C is a graph of CDFs of estimation errors when the nodes in the first
15 scenario are within ending anchor range;

FIG. 7 is a graph of environment and ground truth trajectories for a second scenario;

FIG. 8A is a graph of estimated trajectories of mobile nodes in the second
20 scenario obtained using dead-reckoning (DR);

FIG. 8B is a graph of estimated trajectories of the mobile nodes in the second scenario obtained using the adaptive centralised-extended Kalman filter (AC-EKF) method in accordance with an embodiment of the present invention;

FIG. 9A is a zoomed-in trajectory slice for Node 1 in the second scenario obtained
25 using DR estimation;

FIG. 9B is a zoomed-in trajectory slice for Node 1 in the second scenario obtained using the AC-EKF method in accordance with an embodiment of the present invention;

FIG. 10A is a graph of cumulative distribution functions (CDFs) of estimation errors in the second scenario at 0-5 minutes of a 20-minute trajectory;

5 FIG. 10B is a graph of CDFs of estimation errors in the second scenario at 5-10 minutes of the 20-minute trajectory;

FIG. 10C is a graph of CDFs of estimation errors in the second scenario at 10-15 minutes of the 20-minute trajectory; and

10 FIG. 10D is a graph of CDFs of estimation errors in the second scenario at 15-20 minutes of the 20-minute trajectory.

Detailed Description of Exemplary Embodiments

The detailed description set forth below in connection with the appended drawings is intended as a description of presently preferred embodiments of the invention, and is not intended to represent the only forms in which the present invention
15 may be practiced. It is to be understood that the same or equivalent functions may be accomplished by different embodiments that are intended to be encompassed within the scope of the invention.

Referring now to FIG. 1, a computer-implemented method 10 for mobile node tracking is shown. More particularly, a flowchart of adaptive centralised-extended
20 Kalman filter (AC-EKF) filtering and fusion is shown. The method 10 for mobile node tracking may be executed on one or more processors of a centralised server.

The method 10 begins at step 12 by predicting node positions of a plurality of mobile nodes. The predicted node positions of the mobile nodes may be predicted based on a 9-state motion model. In FIG. 1, x_k^i represents the predicted EKF state x_k of node
25 i and may be a 9-state motion model, in particular, including 3D positions p_k^x, p_k^y, p_k^z , three-dimensional (3D) velocities v_k^x, v_k^y, v_k^z , heading w_k and acceleration a_k^x, a_k^y .

An initialization process may be performed at step 14 to initialise extended Kalman filter (EKF) states for all the mobile nodes in a tracking space.

At step 16, ultra-wideband (UWB) ranging data and instantaneous inertial measurement unit (IMU) data are received from each of the mobile nodes. The mobile nodes may forward the UWB ranging data and the instantaneous IMU data through UWB channels to the server.

As part of communication flow, each of the mobile nodes carries out UWB ranging with the other mobile nodes and/or anchor nodes in the tracking space. The UWB ranging data may thus be obtained from $N \times N$ nodes, where N represents a number of nodes in the tracking space.

While there may be N nodes in the tracking space with the nodes carrying out UWB ranging with one another, there may be practical ranging losses due, for example, to maximum communication range limitations, packet loss or hardware failure. Accordingly, the UWB ranging data that the server receives may not be a complete $N \times N$ matrix and may have missing values (or 0's) between any two node pairs. To reduce mobile node tracking errors and divergence due to incomplete UWB ranging data, an extended Kalman filter-inertial measurement unit (EKF-IMU) dead-reckoning (DR) fail safe is provided when no UWB ranging or external information source is available. The individual EKF-IMU DR for each node may be computed by the server using the instantaneous IMU data. Advantageously, in the absence of UWB ranging data when UWB ranging is unsuccessful or in the absence of node connectivity, a fail-safe is present in the form of the EKF-IMU DR which serves as a default and this reduces overall complexity to $O(n)$, while maintaining tens of centimeter-level accuracy for group tracking.

At step 18, the predicted node positions of the mobile nodes are updated using the UWB ranging data and the instantaneous IMU data. The process of applying this update is done in a centralized manner at the server, thereby providing a centralised tracking solution. Advantageously, by updating the node positions with UWB ranging information carried out between the mobile nodes and any stationary or dynamically placed UWB anchor nodes, accurate group tracking with tens of centimetres of accuracy may be provided. The update stage of step 18 defined as adaptive centralised-extended

Kalman filter (AC-EKF) update for the entire system of nodes may be carried out by the server based on UWB ranging information available and filtered IMU data. The filtered IMU data corresponds to a clean (not noisy) version of sensor data from the mobile nodes following bias calibration and rotation to a frame of reference for respective ones of the
5 mobile nodes.

At step 20, the updated node positions of the mobile nodes are broadcasted from the server to the mobile nodes. More particularly, after updating the predicted node positions with the filtered IMU and UWB ranging data, the server broadcasts the coordinate information or final positions of all the mobile nodes from the server to all
10 nodes for situational awareness. Advantageously, with one-way reporting and coordinate broadcast, overheads may be reduced in this multi-agent localisation scheme. In the exemplary embodiment shown, the coordinate broadcast only broadcasts at step 20 a position state of the state vector x_k , namely $x_k[0:2]$ for all the nodes.

In practice, the UWB distance ranging matrix may not be full, due, for example,
15 to unsuccessful UWB ranging, UWB communication range limits or packet loss during communication. To reduce tracking divergence, ranging losses in the mobile nodes may be identified at step 22 before updating the predicted node positions of the mobile nodes at step 18. This may be by determining a number of non-zero inter-nodal UWB ranging values in the UWB ranging data for each of the mobile nodes as represented, for example,
20 by Equation (1) below:

$$\sum_{j=1..N, j \neq i} (d_m^{ij} \neq 0?) < L, \quad \forall i, i = 1, 2 \dots N; \quad 0 < L \leq N - 1 \quad (1)$$

where d_m^{ij} represents UWB ranging from node i to node j for iteration m , and $j \neq i$. A determination of ranging loss may be made in respect of each of the mobile nodes having less than a predetermined number of non-zero inter-nodal UWB ranging values L in the
25 UWB ranging data. In one exemplary embodiment, the predetermined number of non-zero inter-nodal UWB ranging values in the UWB ranging data may be three (3). In such an embodiment, the server checks at step 22 if node i has successful UWB ranging with at least three (3) out of $N-1$ other mobile nodes and anchor nodes.

In the embodiment shown, $d_m^{ij} = 0$ indicates node i missed UWB ranging with node j . A whitening procedure may be performed at step 24 on each of the mobile nodes identified as having ranging losses. This may be by removing inter-nodal UWB ranging data of the mobile nodes identified as having ranging losses from the UWB ranging data before updating the predicted node positions of the mobile nodes at step 18. More particularly, this may involve removing each row of the inter-nodal UWB ranging data corresponding to respective ones of the mobile nodes identified as having ranging losses from a matrix of the inter-nodal UWB ranging data forming the UWB ranging data provided to the server. In the exemplary embodiment shown, the server carries out the whitening procedure on the ranging matrix for every row i that has fewer than L UWB connections. The whitening procedure removes the nodes that have less than L successful/accurate UWB ranges from the $N \times N$ matrix.

The whitening procedure may be performed on each of the mobile nodes identified as having ranging losses for a predetermined time period. The predetermined time period may correspond to a predetermined number of iterations of the step of updating the predicted node positions of the mobile nodes. In the exemplary embodiment shown, the whitening procedure effectively removes the row corresponding to node i for time window n , where $n = m, m + 1, \dots, m + W$, m represents a current iteration, and W represents a number of iterations to whiten for node i . For example, if $W = 5$, five (5) iterations of the whitening procedure may be carried out for each of the mobile nodes identified as having ranging losses. When the whitening procedure is performed on row i in the UWB ranging matrix, the AC-EKF update effectively defaults to an EKF-IMU update only, which is the tracking possible with EKF and IMU alone. Advantageously, this reduces the overall system complexity for that node that fails to have good UWB ranging. Furthermore, the position of node i may be corrected in future after W iterations and when node i has successful UWB ranging with other nodes again.

It follows from the foregoing that the computer-implemented method 10 for mobile node tracking provides a decision-making flow for a centralized server to compute and

broadcast live coordinate locations of all the mobile nodes. In this centralized framework, processing is offloaded to the centralized server to compute and broadcast node locations. Advantageously, with all processing being handled by the centralized server, computational complexity may be reduced. This allows this infrastructure-less
5 positioning solution to track multiple nodes at tens of centimetres accuracy without divergence. In one or more embodiments, the AC-EKF method 10 may reduce the complexity from $O(n^2)$ to $O(n^2 - mn)$, where $m = \{0, n\}$ represents sparseness of a UWB ranging matrix.

The fusion process 10 fuses IMU and UWB data with the predicted EKF state x_k
10 during the AC-EKF update process and together with strategies for anchor placement and node distribution enables accurate group tracking without divergence for extended trajectories. The mobile nodes may employ strategies to maintain anchor coverage and utilize UWB distance ranging from anchor nodes to rectify any time-based divergence. This framework also allows for dynamic anchors to be dropped by the mobile nodes to
15 maintain time-based divergence in the absence of any existing infrastructure or when the coverage area is larger. Dynamic anchors may also be dropped as another strategy to improve UWB ranging chances. These strategies ensure that the mobile nodes have a higher chance for good UWB ranging. When UWB ranging is present, any time-based divergence may be corrected through the AC-EKF update.

20 Computer program instructions executable by one or more computer processors to perform operations for mobile node tracking described above may be stored in non-transitory computer-readable memory.

Referring now to FIG. 2, a computer system or device 100 suitable for implementing the method 10 for mobile node tracking is shown. The device 100 for
25 mobile node tracking may be a centralized server.

The computer system 100 includes a processor 102 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 104, read only memory (ROM) 106, random access memory (RAM) 108, input/output (I/O) devices 110, and network connectivity devices 112. The
30 processor 102 may be implemented as one or more CPU chips.

It is understood that by programming and/or loading executable instructions onto the computer system 100, at least one of the CPU 102, the RAM 108, and the ROM 106 are changed, transforming the computer system 100 in part into a particular machine or apparatus having the novel functionality taught by the present disclosure. It is
5 fundamental to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules. Decisions between implementing a concept in software versus hardware typically hinge on considerations of stability of the design and numbers of units to be produced rather than any issues
10 involved in translating from the software domain to the hardware domain. Generally, a design that is still subject to frequent change may be preferred to be implemented in software, because re-spinning a hardware implementation is more expensive than re-spinning a software design. Generally, a design that is stable that will be produced in large volume may be preferred to be implemented in hardware, for example in an
15 application specific integrated circuit (ASIC), because for large production runs the hardware implementation may be less expensive than the software implementation. Often a design may be developed and tested in a software form and later transformed, by well-known design rules, to an equivalent hardware implementation in an application specific integrated circuit that hardwires the instructions of the software. In the same
20 manner as a machine controlled by a new ASIC is a particular machine or apparatus, likewise a computer that has been programmed and/or loaded with executable instructions may be viewed as a particular machine or apparatus.

Additionally, after the system 100 is turned on or booted, the CPU 102 may execute a computer program or application. For example, the CPU 102 may execute
25 software or firmware stored in the ROM 106 or stored in the RAM 108. In some cases, on boot and/or when the application is initiated, the CPU 102 may copy the application or portions of the application from the secondary storage 104 to the RAM 108 or to memory space within the CPU 102 itself, and the CPU 102 may then execute instructions that the application is comprised of. In some cases, the CPU 102 may copy the
30 application or portions of the application from memory accessed via the network connectivity devices 112 or via the I/O devices 110 to the RAM 108 or to memory space within the CPU 102, and the CPU 102 may then execute instructions that the application is comprised of. During execution, an application may load instructions into the CPU 102,

for example load some of the instructions of the application into a cache of the CPU 102. In some contexts, an application that is executed may be said to configure the CPU 102 to do something, for example, to configure the CPU 102 to perform the function or functions promoted by the subject application. When the CPU 102 is configured in this way by the application, the CPU 102 becomes a specific purpose computer or a specific purpose machine.

The secondary storage 104 is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM 108 is not large enough to hold all working data. Secondary storage 104 may be used to store programs which are loaded into RAM 108 when such programs are selected for execution. The ROM 106 is used to store instructions and perhaps data which are read during program execution. ROM 106 is a non-volatile memory device which typically has a small memory capacity relative to the larger memory capacity of secondary storage 104. The RAM 108 is used to store volatile data and perhaps to store instructions. Access to both ROM 106 and RAM 108 is typically faster than to secondary storage 104. The secondary storage 104, the RAM 108, and/or the ROM 106 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

I/O devices 110 may include cameras, printers, video monitors, liquid crystal displays (LCDs), plasma displays, touch screen displays, keyboards, keypads, switches, dials, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices. The network connectivity devices 112 may take the form of modems, modem banks, Ethernet cards, universal serial bus (USB) interface cards, serial interfaces, token ring cards, fiber distributed data interface (FDDI) cards, wireless local area network (WLAN) cards, radio transceiver cards that promote radio communications using protocols such as code division multiple access (CDMA), global system for mobile communications (GSM), long-term evolution (LTE), worldwide interoperability for microwave access (WiMAX), near field communications (NFC), radio frequency identity (RFID), and/or other air interface protocol radio transceiver cards, and other well-known network devices. These network connectivity devices 112 may enable the processor 102 to communicate with the Internet or one or more intranets. With such a network connection, it is contemplated that the processor 102 might receive information

from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using processor 102, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave. Such information, which may include data or instructions to be executed using processor 102 for example, may be received from and outputted to the network, for example, in the form of a computer data baseband signal or signal embodied in a carrier wave. The baseband signal or signal embedded in the carrier wave, or other types of signals currently used or hereafter developed, may be generated according to several methods well-known to one skilled in the art. The baseband signal and/or signal embedded in the carrier wave may be referred to in some contexts as a transitory signal.

The processor 102 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk based systems may all be considered secondary storage 104), flash drive, ROM 106, RAM 108, or the network connectivity devices 112. While only one processor 102 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors. Instructions, codes, computer programs, scripts, and/or data that may be accessed from the secondary storage 104, for example, hard drives, floppy disks, optical disks, and/or other device, the ROM 106, and/or the RAM 108 may be referred to in some contexts as non-transitory instructions and/or non-transitory information.

In an embodiment, the computer system 100 may comprise two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the computer system 100 to provide the functionality of a number of servers that is not directly bound to the number of computers in the computer system 100. For example, virtualization software may provide twenty virtual servers on four physical computers. In

an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. Cloud computing may be supported, at least in part, by virtualization software. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third party provider. Some cloud computing environments may comprise cloud computing resources owned and operated by the enterprise as well as cloud computing resources hired and/or leased from a third party provider.

10 In an embodiment, some or all of the functionality disclosed above may be provided as a computer program product. The computer program product may comprise one or more computer readable storage medium having computer usable program code embodied therein to implement the functionality disclosed above. The computer program product may comprise data structures, executable instructions, and other computer
15 usable program code. The computer program product may be embodied in removable computer storage media and/or non-removable computer storage media. The removable computer readable storage medium may comprise, without limitation, a paper tape, a magnetic tape, magnetic disk, an optical disk, a solid state memory chip, for example analog magnetic tape, compact disk read only memory (CD-ROM) disks, floppy disks,
20 jump drives, digital cards, multimedia cards, and others. The computer program product may be suitable for loading, by the computer system 100, at least portions of the contents of the computer program product to the secondary storage 104, to the ROM 106, to the RAM 108, and/or to other non-volatile memory and volatile memory of the computer system 100. The processor 102 may process the executable instructions and/or data
25 structures in part by directly accessing the computer program product, for example by reading from a CD-ROM disk inserted into a disk drive peripheral of the computer system 100. Alternatively, the processor 102 may process the executable instructions and/or data structures by remotely accessing the computer program product, for example by downloading the executable instructions and/or data structures from a remote server
30 through the network connectivity devices 112. The computer program product may comprise instructions that promote the loading and/or copying of data, data structures, files, and/or executable instructions to the secondary storage 104, to the ROM 106, to

the RAM 108, and/or to other non-volatile memory and volatile memory of the computer system 100.

In some contexts, the secondary storage 104, the ROM 106, and the RAM 108 may be referred to as a non-transitory computer readable medium or a computer readable storage media. A dynamic RAM embodiment of the RAM 108, likewise, may be referred to as a non-transitory computer readable medium in that while the dynamic RAM receives electrical power and is operated in accordance with its design, for example during a period of time during which the computer system 100 is turned on and operational, the dynamic RAM stores information that is written to it. Similarly, the processor 102 may comprise an internal RAM, an internal ROM, a cache memory, and/or other internal non-transitory storage blocks, sections, or components that may be referred to in some contexts as non-transitory computer readable media or computer readable storage media.

Experimental Results

Simulations were performed to demonstrate the capability of the adaptive centralised-extended Kalman filter (AC-EKF) method 10 in maintaining group tracking in two (2) realistic environments using the following noisy measurements: $\sigma_g = 5e^{-2^\circ}/s$, $\sigma_a = 0.1m/s^2$, $\sigma_{UWB} = 5cm$, where g represents gyroscope, a represents accelerometer and UWB represents UWB ranging. The node trajectories have been designed as random walk with an average walking speed of 1 metre per second (m/s) and a maximum ultra-wideband (UWB) communication range of 15 metres (m).

Referring now to FIG. 3, environment and ground truth trajectories for a first scenario are shown. In the first scenario, three (3) UWB anchors are placed at each corner of a 60 square metre (m^2) warehouse. Two (2) groups of four (4) mobile nodes are tracked within the warehouse as the mobile nodes move from one working quadrant to an opposite quadrant. The entire simulated trajectories of the mobile nodes span 20 minutes (min).

Referring now to FIGS. 4A and 4B, entire trajectories of the mobile nodes estimated with dead-reckoning (DR) and the adaptive centralised-extended Kalman filter

(AC-EKF) method 10 are shown. More particularly, trajectory estimation with DR is shown in FIG. 4A and trajectory estimation with AC-EKF is shown in FIG. 4B.

Referring now to FIGS. 5A and 5B, clearer zoomed-in trajectory slices for one of the nodes (Node 5) are shown. More particularly, FIG. 5A shows a zoomed-in trajectory slice for Node 5 using DR estimation and FIG. 5B shows a zoomed-in trajectory slice for Node 5 using AC-EKF estimation. In FIGS. 5A and 5B, the ground truth (GT) is represented by dots and estimations by a dashed line, as indicated in the legend. As can be seen from FIG. 5A, a stark difference is observed between the GT and DR estimated trajectory as a result of divergence over time. However, as can be seen from FIG. 5B, the AC-EKF estimation overlaps the GT almost entirely. This can be represented quantitatively by measuring the estimation errors for each trajectory as shown in FIGS. 6A through 6C.

Referring now to FIGS. 6A through 6C, cumulative distribution function (CDF) comparisons of the estimation errors based on location are shown. More particularly, FIG. 6A shows CDFs of estimation errors when the nodes are within starting anchor range, FIG. 6B shows CDFs of estimation errors when the nodes are out of anchor range, and FIG. 6C shows CDFs of estimation errors when the nodes are within ending anchor range.

Divergence is a measure of an increase in tracking error over time. As can be seen from the estimation errors plotted for the time period when the node groups are within their starting quadrant and in range of the static anchors shown in FIG. 6A, both DR and AC-EKF perform well with only 5 centimetres (cm) of difference in the estimations. However, as time progresses and the nodes start to move away from the anchors, the difference grows to more than 1.5 m (90th-percentile error) as can be seen in FIG. 6C.

Another important observation is a small divergence (approximately 13 cm in the 90th-percentile error) noted for AC-EKF when the nodes have no anchors in range as seen from FIG. 6B and a quick convergence seen in FIG. 6C when the nodes return to range with opposite quadrant anchors.

As can be observed from FIGS. 6A through 6C, the IMU DR drifts by more than 1500 millimetres (mm) in its 90th-percentile error from the start of the trajectory to the end of the trajectory in Scenario 1. On the other hand, the AC-EKF, in comparison, only drifts by 60.25 mm in its 90th-percentile estimation error over 20 minutes of tracking, which is very stable. Moreover, when comparing FIG. 6B to FIG. 6C, it can be observed that the AC-EKF estimation quickly improves from 188.82 mm to 118.12 mm which corresponds to fast convergence in the presence of anchors. This demonstrates that the AC-EKF achieves slow and negligible divergence due to UWB ranging update and fast convergence in the presence of stationary anchors.

Referring now to FIG. 7, environment and ground truth trajectories for a second scenario are shown. In the second scenario, a building with a single-entry point that has three (3) UWB anchors placed outside is simulated. The mobile nodes are thought of as firefighters wearing UWB tags that are carrying out a search-and-rescue operation as shown in FIG. 7. With a maximum UWB communication range set at 15 m, certain nodes may travel out of range, while other intermediary nodes may search in regions closer to the anchors. This setup demonstrates that with no added infrastructure during the search-and-rescue operation, the group tracking does not diverge. 20-minute trajectories were simulated in order to have a realistic drift in the order of metres and to demonstrate convergence capability with the UWB ranging and AC-EKF method 10.

Referring now to FIGS. 8A and 8B, trajectories of the mobile nodes estimated with dead-reckoning (DR) and the adaptive centralised-extended Kalman filter (AC-EKF) method 10 are shown. More particularly, trajectory estimation with DR is shown in FIG. 8A and trajectory estimation with AC-EKF is shown in FIG. 8B. From FIGS. 8A and 8B, the overall trajectory estimation for the mobile nodes obtained using DR and AC-EKF can be observed and a divergence, particularly for DR estimation of Node 5, can be seen.

Referring now to FIGS. 9A and 9B, clearer zoomed-in trajectory slices for one of the nodes (Node 1) are shown. More particularly, FIG. 9A shows a zoomed-in trajectory slice for Node 1 using DR estimation and FIG. 9B shows a zoomed-in trajectory slice for Node 1 using AC-EKF estimation. On zooming in to Node 1, a comparative difference in the quality of estimation can be seen from FIGS. 9A and 9B.

Referring now to FIGS. 10A through 10D, cumulative distribution function (CDF) comparisons of the estimation errors based on time-windows are shown. More particularly, FIG. 10A shows CDFs of estimation errors at 0-5 minutes of a 20-minute trajectory, FIG. 10B shows CDFs of estimation errors at 5-10 minutes of the 20-minute trajectory, FIG. 10C shows CDFs of estimation errors at 10-15 minutes of the 20-minute trajectory and FIG. 10D shows CDFs of estimation errors at 15-20 minutes of the 20-minute trajectory. FIGS. 10A through 10D encapsulate the quantitative divergence over time by splitting the 20-minute trajectory into four (4) windows of 5 minutes.

It can be observed from FIGS. 10A through 10D that while the DR estimation has a near constant-drift with time, the AC-EKF estimation actually improves over time as the geometry of the nodes is more spaced out (comparing FIG. 10B to that of FIG. 10A) and stabilizes (does not diverge) as time increases (as seen in FIGS. 10C and 10D) with 90th-percentile errors within 6 cm for most of the trajectory. It is clear from FIGS. 10A through 10D that the AC-EKF is stable at 60 mm for even extended periods of time, while the IMU DR results consistently gets worse (or diverges) over time.

Overall, the estimations using the AC-EKF method are within 10 cm for both scenarios when nodes are within range of static anchors or intermediary nodes that can range with the anchors and are within 20 cm for scenarios without any anchors. This demonstrates the flexibility and robustness of the AC-EKF fusion solution for group tracking. As described above, the simulation parameters are constrained to just twelve (12) anchors at the corners of the 60 m² warehouse – three (3) at each of the four (4) corners of the warehouse, with a smaller communication range of 15 m and only two (2) groups of four (4) mobile nodes in the first scenario and three (3) anchors with twelve (12) mobile nodes in the second scenario, for 20-minute random walks with 90th-percentile errors below 20 cm. Even better performance may be expected with more anchors, greater coverage and greater connectivity with more mobile nodes.

As is evident from the foregoing discussion, the present invention provides a computer-implemented method and a device for mobile node tracking with reduced infrastructure requirements and that provide accurate group tracking with reduced divergence for extended trajectories. Advantageously, the computer-implemented method and the device of the present invention provide a framework for infrastructure-less positioning and a fusion solution to track a group of mobile nodes even for extended

trajectories within tens of centimetres accuracy. The fusion solution includes individual dead reckoning (DR) EKF-IMU (Extended Kalman Filter – Inertial Measurement Unit) tracking and an adaptive centralized EKF (AC-EKF) update based on UWB distance ranging. Further advantageously, because processing is handled by a centralized server
5 that is privy to all information available to the system of nodes, the need for a processing unit on mobile nodes is eliminated and this significantly reduces power consumption requirements as well. With one-way reporting and coordinate broadcast, overheads for the multi-agent localization scheme is also reduced.

While preferred embodiments of the invention have been illustrated and
10 described, it will be clear that the invention is not limited to the described embodiments only. Numerous modifications, changes, variations, substitutions and equivalents will be apparent to those skilled in the art without departing from the scope of the invention as described in the claims.

Further, unless the context clearly requires otherwise, throughout the description
15 and the claims, the words "comprise", "comprising" and the like are to be construed in an inclusive as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to".

CLAIMS

1. A computer-implemented method for mobile node tracking, comprising executing on one or more processors of a server the steps of:
 - 5 predicting node positions of a plurality of mobile nodes;
receiving ultra-wideband (UWB) ranging data and instantaneous inertial measurement unit (IMU) data from each of the mobile nodes;
updating the predicted node positions of the mobile nodes using the UWB ranging data and the instantaneous IMU data; and
 - 10 broadcasting the updated node positions of the mobile nodes from the server to the mobile nodes.
2. The computer-implemented method for mobile node tracking according to claim 1, further comprising identifying ranging losses in the mobile nodes before updating the
15 predicted node positions of the mobile nodes.
3. The computer-implemented method for mobile node tracking according to claim 2, wherein the step of identifying ranging losses in the mobile nodes comprises:
 - determining a number of non-zero inter-nodal UWB ranging values in the UWB
20 ranging data for each of the mobile nodes.
4. The computer-implemented method for mobile node tracking according to claim 3, wherein the step of identifying ranging losses in the mobile nodes further comprises:
 - making a determination of ranging loss in respect of each of the mobile nodes
25 having less than a predetermined number of non-zero inter-nodal UWB ranging values in the UWB ranging data.
5. The computer-implemented method for mobile node tracking according to claim 4, wherein the predetermined number of non-zero inter-nodal UWB ranging values in the
30 UWB ranging data is three (3).
6. The computer-implemented method for mobile node tracking according to any one of claims 2 to 5, further comprising performing a whitening procedure on each of the mobile nodes identified as having ranging losses.

7. The computer-implemented method for mobile node tracking according to claim 6, wherein the step of performing the whitening procedure comprises:
removing inter-nodal UWB ranging data of the mobile nodes identified as having ranging losses from the UWB ranging data before updating the predicted node positions of the mobile nodes.
8. The computer-implemented method for mobile node tracking according to claim 7, wherein the step of removing the inter-nodal UWB ranging data of the mobile nodes comprises:
removing each row of the inter-nodal UWB ranging data corresponding to respective ones of the mobile nodes identified as having ranging losses from a matrix of the inter-nodal UWB ranging data forming the UWB ranging data provided to the server.
9. The computer-implemented method for mobile node tracking according to any one of claims 6 to 8, wherein the whitening procedure is performed on each of the mobile nodes identified as having ranging losses for a predetermined time period.
10. The computer-implemented method for mobile node tracking according to claim 9, wherein the predetermined time period corresponds to a predetermined number of iterations of the step of updating the predicted node positions of the mobile nodes.
11. A device for mobile node tracking, comprising:
one or more processors; and
a non-transitory computer-readable memory storing computer program instructions executable by the one or more processors to perform operations for mobile node tracking, the operations comprising:
predicting node positions of a plurality of mobile nodes;
receiving ultra-wideband (UWB) ranging data and instantaneous inertial measurement unit (IMU) data from each of the mobile nodes;
updating the predicted node positions of the mobile nodes using the UWB ranging data and the instantaneous IMU data; and
broadcasting the updated node positions of the mobile nodes to the mobile nodes.

12. The device for mobile node tracking according to claim 11, wherein the operations further comprise identifying ranging losses in the mobile nodes before updating the predicted node positions of the mobile nodes.

5

13. The device for mobile node tracking according to claim 12, wherein the operation of identifying ranging losses in the mobile nodes comprises:

determining a number of non-zero inter-nodal UWB ranging values in the UWB ranging data for each of the mobile nodes.

10

14. The device for mobile node tracking according to claim 13, wherein the operation of identifying ranging losses in the mobile nodes further comprises:

making a determination of ranging loss in respect of each of the mobile nodes having less than a predetermined number of non-zero inter-nodal UWB ranging values

15 in the UWB ranging data.

15. The device for mobile node tracking according to claim 14, wherein the predetermined number of non-zero inter-nodal UWB ranging values in the UWB ranging data is three (3).

20

16. The device for mobile node tracking according to any one of claims 12 to 15, wherein the operations further comprise performing a whitening procedure on each of the mobile nodes identified as having ranging losses.

25 17. The device for mobile node tracking according to claim 16, wherein the operation of performing the whitening procedure comprises:

removing inter-nodal UWB ranging data of the mobile nodes identified as having ranging losses from the UWB ranging data before updating the predicted node positions of the mobile nodes.

30

18. The device for mobile node tracking according to claim 17, wherein the operation of removing the inter-nodal UWB ranging data of the mobile nodes comprises:

removing each row of the inter-nodal UWB ranging data corresponding to respective ones of the mobile nodes identified as having ranging losses from a matrix of the inter-nodal UWB ranging data forming the UWB ranging data provided to the server.

5 19. The device for mobile node tracking according to any one of claims 16 to 18, wherein the whitening procedure is performed on each of the mobile nodes identified as having ranging losses for a predetermined time period.

20. The device for mobile node tracking according to claim 19, wherein the
10 predetermined time period corresponds to a predetermined number of iterations of the step of updating the predicted node positions of the mobile nodes.