

**AN OPTIMALITY-AWARE LIDAR INERTIAL ODOMETRY MODULE
WITH HYBRID CONTINUOUS TIME OPTIMIZATION**

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of priority to Singapore patent application no. 10202303272R which was filed on 20 November 2023, the contents of which are hereby incorporated by reference in its entirety for all purposes.

TECHNICAL FIELD

[0002] This application relates to a LiDAR inertial odometry module that employs a hybrid continuous time optimization technique to mitigate errors introduced by human motion when low-cost and compact wearable mapping systems are used to implement LiDAR inertial odometry methods for collecting LiDAR-based 3D maps of structures and complex environments.

BACKGROUND

[0003] Due to their mobility, miniaturization, and portability, Wearable Mapping Systems (WMS) have emerged as promising tools in fields such as emergency rescue, construction, and Building Information Modeling (BIM). WMS offers an efficient solution for collecting 3D data in Global Navigation Satellite System (GNSS)-denied complex environments that are often inaccessible to wheeled or legged robots. The resulting 3D maps can then be used as prior maps for robot navigation systems.

[0004] Several commercial WMS platforms, such as the backpack-based NavVis, which utilizes multiple scanners, and the handheld GeoSLAM system, which employs a rotation module, have been successfully used to develop prior maps in GNSS-denied environments. For example, those skilled in the art have designed a dual-LiDAR backpack mapping system that uses ground constraints to limit SLAM drift in multi-level buildings. However, while multi-scanner configurations and rotating modules increase the Field of View (FoV) of such systems, they also add weight and operational burden. Furthermore, most existing WMS systems use high-grade Inertial Measurement Units (IMUs), which significantly increase costs, making them inaccessible for non-professional users. As a result, those skilled in the art are constantly striving to develop compact, affordable WMS solutions.

[0005] In low-cost WMS setups, challenges arise from the limited FoV of single-LiDAR configurations and the high noise levels associated with affordable IMUs (e.g., the low cost ICM40609 IMU). Moreover, existing WMS systems generally require users to move slowly to avoid vibrations, which degrades the accuracy of LiDAR-Inertial Odometry (LIO) methods making it difficult for such systems to be used in real-world scenarios. Head-mounted systems have also been adopted in low-cost WMS setups. However, compared to backpack or handheld ones, such systems are more susceptible to vibration but offer the advantage of "what you see is what you get". As a result, those skilled in the art have strived to improve the performance of compact, helmet-based WMS.

[0006] For trajectory formulation and optimization of LIO, existing methods can be generally categorized into two types: Continuous Time Optimization (CTO) and Discrete Time Optimization (DTO). Compared with the traditional DTO approach, the CTO approach has several intrinsic benefits and has been investigated for use in LIO systems in recent years by those skilled in the art. Firstly, a CTO's trajectory could be sampled at any time, which makes it suitable for use in the fusing of asynchronous data (e.g., LiDAR and IMU) with human motion constraints. Secondly, additional variables at the time instances corresponding to a measurement or motion constraint do not need to be introduced, which keeps the number of variables manageable. However, the CTO method assumes smooth, polynomial motions, which may lead to loss of information and an ill-fitted model at high-variation segments, i.e., the motion vibration caused by human motions. Further, the CTO approach tends to increase the model fitting ability by using a higher-order B-spline trajectory or by adding more knots to the active LiDAR frame.

[0007] A zero-velocity detection and update (ZUPT) approach has been proposed by those skilled in the art to constrain human motion, but it was found to be only applicable for foot-mounted IMUs. With the rapid development of deep learning, a lot of data-driven motion-based methods have been proposed to predict IMU bias or IMU velocity to contain the accumulation of low cost IMU errors. Moreover, the motion-based constraints can be integrated into batch optimization models to enhance the system's overall performance and have been applied in loop closure detection techniques as well.

[0008] In theory, attitude (roll and pitch) is observable in LIO and Visual-Inertial Odometry (VIO) due to the gravity vector. However, in practice, issues such as uneven feature distribution or degeneration tends to cause the matching process to converge to a local minima, which then leads to erroneous IMU state estimations and loss of attitude observability. This problem is exacerbated when low-cost IMUs are used or when the IMUs are moving under vibration, as commonly encountered by wearable mapping systems.

[0009] In LiDAR-based systems, a B-spline trajectory is commonly used to model the continuous motion of the system over time in a smooth and mathematically flexible manner. For a k -order B-spline trajectory, each constraint corresponds to k knots, which results in more Jacobian estimations being performed as compared to the DTO during the batch optimization process. Selecting fewer, but more important, feature points can improve B-spline trajectory estimation performance. Conversely, maintaining all feature points in the LIO system could hinder real-time performance. To address this, those skilled in the art have attempted to select feature points according to space distribution, based on different semantic types, or according to the information matrix's spectral attributes using a greedy optimization approach.

[0010] Despite efforts of those skilled in the art, when existing LIO methods are applied to compact WMS, these efforts face two main challenges: (1) The first challenge faced is that human motion vibrations cannot be effectively modeled with low-order splines (e.g., 4th order), and increasing the spline order adds significant computational overhead and uncertainty. (2) The second challenge is that uneven or degenerated LiDAR feature correspondences lead to suboptimal convergence, resulting in long-term drift. These issues cause attitude and Z-axis drift, leading to "bent" point cloud maps that are geometrically consistent locally but unsuitable for robot navigation and planning systems.

SUMMARY

[0011] In one aspect, the present application discloses a module for generating a set of optimized estimated states for generating a LiDAR point cloud map in real time. The module comprises a processing unit, and a non-transitory media readable by the processing unit. The media contains instructions that when executed by the processing unit causes the processing unit to segment sequential data received from an inertial measurement unit (IMU) into a low-frequency part, a high-frequency part and a constant velocity part. The processing unit then

computes hybrid IMU constraints based on data contained within the low-frequency part, the high-frequency part and the constant velocity part and then proceeds to compute an initial guess of system states for an active LiDAR frame by performing an IMU optimization process based on the hybrid IMU constraints. The processing unit then computes an optimal set of LiDAR correspondences based on measurements of the active LiDAR frame and the initial guess of system states before proceeding to compute a set of optimized estimated states for the active LiDAR frame by performing a maximum-a-priori (MAP) optimization process using the optimal set of LiDAR correspondences, the hybrid IMU constraints, and prior constraints. Once this is done, the processing unit then generates the LiDAR point cloud map based on the set of optimized estimated states.

[0012] In another aspect, the present application discloses a method for generating a set of optimized estimated states for generating a motion-corrected LiDAR point cloud map in real time using a computing module. The method comprises the steps of segmenting sequential data received from an inertial measurement unit (IMU) into a low-frequency part, a high-frequency part and a constant velocity part, computing hybrid IMU constraints based on data contained within the low-frequency part, the high-frequency part and the constant velocity part and computing an initial guess of system states for an active LiDAR frame by performing an IMU optimization process based on the hybrid IMU constraints. The method then includes the steps of obtaining an optimal set of LiDAR correspondences based on measurements of the active LiDAR frame and the initial guess of system states and computing the set of optimized estimated states for the active LiDAR frame by performing a maximum-a-priori (MAP) optimization process using the optimal set of LiDAR correspondences, the hybrid IMU constraints, and prior constraints.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Various embodiments of the present disclosure are described below with reference to the following drawings:

Figure 1 illustrates a block diagram of a LiDAR Inertial Odometry (LIO) with Hybrid Continuous Time Optimization (HCTO) in accordance with embodiments of the present disclosure;

Figure 2 illustrates a block diagram representative of a HCTO module of the LIO with HCTO as illustrated in Figure 1 in accordance with embodiments of the present disclosure;

Figure 3 illustrates plots of different motion states in a repetitive motion pattern;

Figure 4 illustrates the grouping of candidate correspondences in a current LiDAR frame according to embodiments of the present disclosure;

Figure 5 illustrates a Maximum-A-Priori (MAP) optimization in an active window with the 4th order B-spline as an example;

Figure 6 illustrates a block diagram of a processing system for performing embodiments of the present disclosure;

Figure 7 illustrates a flowchart of a process for generating a LiDAR point cloud in real time using a LIO with HCTO module in accordance with embodiments of the present disclosure;

Figure 8 illustrates a prior map constructed by a HCTO process for a robot navigation system in different challenging environments using wearable devices;

Figure 9 illustrates localization errors in the trajectory of a HCTO for a seq-subway-station when a wearable helmet is used to capture the data;

Figure 10 illustrates localization errors in different axes of the HCTO process for a seq-subway-station when a wearable helmet is used to capture the data;

Figure 11 illustrates point clouds constructed by the HCTO module using wearable helmets for a car park sequence and a subway station sequence;

Figure 12 illustrates datasets of a university campus where the intensity rendered point clouds are overlapped on the satellite image of the university campus;

Figure 13a illustrates a side view of a random LiDAR frame in Site 1, where the visualization of the point clouds are generated by the HCTO module;

Figure 13b illustrates a side view of a random LiDAR frame in Site 2, where the visualization of the point clouds are generated by the HCTO module;

Figure 14 illustrates localization error analysis of seq-02 of Site 1 of the university campus;

Figure 15 illustrates localization error analysis of seq-02 of Site 2 of the university campus;

Figure 16 illustrates a visual comparison of the different methods employed at Site 2 of the university campus where the point clouds are rendered by height to have a better visualization of the drift in attitude and z-direction; and

Figure 17 illustrates a prior map constructed using the HCTO technique for an outside view of a four-story apartment, an inside view of the apartment and a long corridor of each floor of the apartment.

DETAILED DESCRIPTION

[0014] The following detailed description is made with reference to the accompanying drawings, showing details and embodiments of the present disclosure for the purposes of illustration. Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments, even if not explicitly described in these other embodiments. Additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0015] In the context of various embodiments, the articles “a”, “an” and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0016] In the context of various embodiments, the term “about” or “approximately” as applied to a numeric value encompasses the exact value and a reasonable variance as generally understood in the relevant technical field, e.g., within 10% of the specified value.

[0017] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0018] As used herein, “comprising” means including, but not limited to, whatever follows the word “comprising”. Thus, use of the term “comprising” indicates that the listed elements are required or mandatory, but that other elements are optional and may or may not be present.

[0019] As used herein, “consisting of” means including, and limited to, whatever follows the phrase “consisting of”. Thus, use of the phrase “consisting of” indicates that the listed elements are required or mandatory, and that no other elements may be present.

[0020] As used herein, an Inertial Measurement Unit “IMU” refers to an electronic device that measures and provides data sequentially on an object’s motion and orientation based on sensors, such as accelerometers and/or gyroscopes, that are provided within the device. The sequential data provided by the IMU may comprise, but are not limited to, linear acceleration data, angular velocity data, orientation data, velocity data or positional data.

[0021] As used herein, a “world frame” refers to a global coordinate system used as a reference for measuring the position and orientation of objects in real space. This global coordinate system remains constant and is typically aligned with a known reference, such as the Earth's surface or a map. A “body frame” as used herein refers to a coordinate system that is attached to a moving object, such as a helmet or a similar wearable item. The body frame's coordinate system moves and rotates with the object, and its origin is often centered at the object's center of mass or another key point.

[0022] One skilled in the art will recognize that certain functional units in this description have been labelled as modules throughout the specification. The person skilled in the art will also recognize that a module may be implemented as circuits, logic chips or any sort of discrete component. Still further, one skilled in the art will also recognize that a module may be implemented in software which may then be executed by a variety of processor architectures. In embodiments of the disclosure, a module may also comprise computer instructions or executable code that may instruct a computer processor to carry out a sequence of events based on instructions received. The choice of the implementation of the modules is left as a design choice to a person skilled in the art and does not limit the scope of the claimed subject matter in any way.

[0023] Further, it should be noted that the following notations are adopted throughout this disclosure. For a vector $\mathbf{p} \in \mathbb{R}^3$, a hat notation $\hat{\mathbf{p}}$ denotes the global optimization-based estimation of $\mathbf{p}$ while the breve notation $\breve{\mathbf{p}}$ denotes the IMU-propagated information. For a matrix $\mathbf{M}$, $\mathbf{M}_{r,c}$ denotes the element at a r^{th} row and c^{th} column of $\mathbf{M}$. The orientation could be represented by rotation matrix $\mathbf{R} \in SO(3)$. The right superscript and subscript are then used to define the coordinate transformation, i.e., $[\mathbf{R}_B^A, \mathbf{p}_B^A]$, transforming a vector from frame B to frame A.

[0024] In this disclosure, reference is made to an i^{th} time segment spanning a period $[t_i, t_{i+1}]$ with a constant duration Δt . A system state $\mathcal{X}(t)$ at time $t \in [t_i, t_{i+1}]$ is defined as:

$$\mathcal{X}(t) = [\mathbf{R}_b^\omega(t), \mathbf{p}_b^\omega(t), \mathbf{b}_a(t), \mathbf{b}_g(t)] \in SO(3) \times \mathbb{R}^9 \quad \dots \text{equation (1)}$$

where $\mathbf{R}_b^\omega(t), \mathbf{p}_b^\omega(t)$ are rotation and translation of body frame as defined in the world frame, a $\mathbf{q}_b^\omega(t)$ is defined as a quaternion corresponding to $\mathbf{R}_b^\omega(t)$, $\mathbf{b}_a(t), \mathbf{b}_g(t)$ are defined as IMU accelerometer and gyroscope biases respectively – which are modeled based on the random-walk model for low-cost IMUs. It should be noted that a $\mathbf{v}_b^\omega(t)$ is not included in the system states (i.e., in equation (1) above) as a B-spline was used to formulate the trajectory.

[0025] **IMU pre-integration step**

[0026] An IMU pre-integration step is commonly used in Discrete-Time-Optimization (DTO) processes where the IMU measurements are integrated over a time duration to obtain relative spatial relationships. Given the i^{th} time segment where $t \in [t_i, t_{i+1}]$, the state variables are constrained by the IMU measurements, i.e., acceleration $\bar{\mathbf{a}}^b(t)$ and gyroscope $\bar{\boldsymbol{\omega}}^b(t)$ in this time segment. The IMU measurements integrated over the time duration may then be defined as:

$$\begin{aligned}\mathbf{p}_b^\omega(t_{i+1}) &= \mathbf{p}_b^\omega(t_i) + \mathbf{v}_b^\omega(t_i)\Delta t + \iint_{t \in [t_i, t_{i+1}]} (\mathbf{R}_{b_t}^\omega(\bar{\mathbf{a}}^b(t) - \mathbf{b}_a(t_i)) - \mathbf{g}^\omega) dt^2, \\ \mathbf{v}_b^\omega(t_{i+1}) &= \mathbf{v}_b^\omega(t_i) + \iint_{t \in [t_i, t_{i+1}]} (\mathbf{R}_{b_t}^\omega(\bar{\mathbf{a}}^b(t) - \mathbf{b}_a(t_i)) - \mathbf{g}^\omega) dt, \\ \mathbf{q}_b^\omega(t_{i+1}) &= \mathbf{q}_b^\omega(t_i) \otimes \iint_{t \in [t_i, t_{i+1}]} 0.5\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}^b(t) - \mathbf{b}_g(t_i))\mathbf{q}_{b_t}^\omega dt \quad \dots \text{equation (2)}\end{aligned}$$

$$\text{where } \boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\omega_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix}, \boldsymbol{\omega}_\times = \begin{bmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{bmatrix} \quad \dots \text{equation (3)}$$

where $\mathbf{g}^\omega$ is defined as a gravity vector in the world frame, $\omega_0, \omega_1, \omega_2$ are defined as three elements in vector $\boldsymbol{\omega}$. When the reference frame of integration is changed to b_i , the three IMU pre-integration factors, $\Delta \mathbf{p}_{i+1}^i, \Delta \mathbf{v}_{i+1}^i$ and $\Delta \mathbf{q}_{i+1}^i$, may be rewritten as equation (4) below:

$$\begin{aligned}\Delta \mathbf{p}_{i+1}^i &= \iint_{t \in [t_i, t_{i+1}]} \mathbf{R}_{b_t}^{b_i}(\bar{\mathbf{a}}_t - \mathbf{b}_a(t_i)) dt^2, \\ \Delta \mathbf{v}_{i+1}^i &= \iint_{t \in [t_i, t_{i+1}]} \mathbf{R}_{b_t}^{b_i}(\bar{\mathbf{a}}_t - \mathbf{b}_a(t_i)) dt, \\ \Delta \mathbf{q}_{i+1}^i &= \iint_{t \in [t_i, t_{i+1}]} 0.5\boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}_t - \mathbf{b}_g(t_i))\Delta \mathbf{q}_{b_t}^{b_i} dt \quad \dots \text{equation (4)}\end{aligned}$$

[0027] The three factors in equation (4) may also be used to restrict the relative motion of the spline trajectory which will be used in the subsequent sections of the disclosure.

[0028] **Continuous Time Optimization (CTO) using B-Spline**

[0029] A k^{th} order B-Spline having $N + 1$ knots (or control points) may be formulated using the De Boor-Cox recurrence relation which is defined as:

$$\mathbf{p}(t) = \sum_{i=0}^N B_{i,k}(t) \mathbf{p}_i \quad \dots \text{equation (5)}$$

where $B_{i,k}(t)$ is defined as the B-Spline basis function. The function $u(t) = (t - t_0)/\Delta t - i$ is defined as normalized time elapsed in the window $[t_i, t_{i+1}]$ and is referred to in this disclosure as ' u ' for brevity. The value $\mathbf{p}(u)$ may then be represented as a matrix multiplication:

$$\mathbf{p}(u) = [\mathbf{p}_i, \mathbf{p}_{i+1}, \dots, \mathbf{p}_{i+k-1}] \mathbf{M} \mathbf{u} \quad \dots \text{equation (6)}$$

where $\mathbf{u}_n = u^n$ and $\mathbf{M}$ is the k^{th} order blending matrix which is defined as:

$$\mathbf{M}_{s,n} = \frac{C_{k-1}^n}{(k-1)!} \sum_{l=s}^{k-1} (-1)^{l-s} (k-l-1)^{k-1-n} \quad \dots \text{equation (7)}$$

$s, n \in 0, \dots, k-1$

where $\mathbf{p}(u)$ can be transformed into a cumulative formulation by:

$$\mathbf{p}(u) = \mathbf{p}_i + \sum_{j=1}^{k-1} (\mathbf{p}_{i+j} - \mathbf{p}_{i+j-1}) (\tilde{\mathbf{M}} \mathbf{u})_j \quad \dots \text{equation (8)}$$

where $\tilde{\mathbf{M}}_{j,n} = \sum_{s=j}^{k-1} \mathbf{M}_{s,n}$.

[0030] As the system state knot $\mathbf{x}_i$ is in a Lie group, the cumulative B-Spline of the system with state knots $\{\mathbf{x}_0, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N\}$ may be defined as:

$$\mathcal{X}(u) = \mathcal{X}_i \prod_{j=1}^{k-1} \text{Exp}(\text{Log}(\mathcal{X}_{i+j-1}^{-1} \mathcal{X}_{i+j})(\tilde{\mathbf{M}}\mathbf{u})_j) \quad \dots \text{equation (9)}$$

[0031] Based on the above, the velocity $\dot{\mathbf{p}}_b^\omega(u)$, the acceleration $\ddot{\mathbf{p}}_b^\omega(u)$, and the angular velocity $\dot{\mathbf{R}}_b^\omega(u)$, may be obtained at any time based on the differential of the cumulative B-spline of the system, $\mathcal{X}(u)$. The IMU measurements at these specific times could be used to restrict the spline trajectory by comparing the differential results and raw IMU measurements. The exact details of fusing the IMU information of the CTO are omitted for brevity as it is a standard way that is known to one skilled in the art.

[0032] A block diagram of LiDAR Inertial Odometry (LIO) system 100 with Hybrid Continuous Time Optimization (HCTO) is illustrated in Figure 1 in accordance with embodiments of the disclosure. System 100 comprises wearable sensing device 101 and a computing module 102 for processing data received from device 101. Device 101 is provided with IMU 103 and LiDAR module 106. In embodiments of the disclosure, IMU 103 may comprise, but is not limited to a low-cost IMU such as the ICM40609 IMU module where IMU 103 and LiDAR module 106 may both be mounted on a compact helmet. Computing module 102, is provided with HCTO module 104, motion un-distortion module 108, optimal-aware point feature association module 110, maximum-a-priori (MAP) optimization module 112 and keyframe selection module 114. In general, system 100 is configured to construct hybrid IMU constraints based on sequential data obtained from IMU 103 while removing LiDAR motion distortions from LiDAR data measurements obtained from LiDAR module 106. In embodiments of the disclosure, computing module 102 may be provided within the compact helmet or may be located remotely, within a communicative range of device 101 through wired or wireless means.

[0033] HCTO module 104 is configured to segment sequential data obtained from IMU 103 into a low-frequency part, a high-frequency part and a constant velocity part. Hybrid IMU constraints 105 are then computed based on data contained within each of these parts. HCTO module 104 then performs an IMU optimization process based on the hybrid IMU constraints to obtain an initial guess of system states 107 for an active LiDAR frame as obtained from LiDAR module 106 where distortions caused by the motion of LiDAR module 106 during the scanning process are corrected. Motion un-distortion module 108 and optimal-aware point feature association module 110 then combine to utilize the initial guess of system states 107

and data contained in the active LiDAR frame to generate an optimal set of LiDAR correspondences 111 where the most informative points are selected for further processing. LiDAR correspondences 111, together with IMU hybrid constraints 105, LiDAR factor r_t^L and prior constraints r_t^M are then used by MAP optimization module 112 to compute a set of optimized estimated states 113 for the active LiDAR frame, based on a sliding window. By doing so, a fixed number of LiDAR frames are kept in memory and the system's states are continuously optimized as each new LiDAR frame is received. LiDAR factor r_t^L and prior constraints r_t^M will be defined in greater detail in the subsequent sections. If keyframe selection module 114 determines from optimized estimated states 113 that the current active LiDAR frame is to be selected as a keyframe, this keyframe will then be selected and added to the keyframe database. This step ensures that only important frames are stored. A motion-corrected LiDAR point cloud map may then be generated based on optimized estimated states 113.

[0034] A block diagram representative of modules contained within HCTO module 104 is illustrated in Figure 2 in accordance with embodiments of the present disclosure. HCTO module 104 comprises motion state recognition module 202 that is configured to receive raw IMU measurements 201 from IMU module 103 and to segment raw IMU measurements 201 into a high-frequency part (HFP), a low-frequency part (LFP) and a constant-velocity part (CVP). IMU measurements 201 that have been segmented into a HFP is provided to HFP module 204, which then proceeds to compute HFP factors based on data in the HFP and pre-integration constants or parameters generated by sub-module 214. IMU measurements 201 that have been segmented into a low-frequency part (LFP) is provided to LFP module 206, which then proceeds to compute LFP factors based on data in the LFP and raw measured constraints as generated by sub-module 216. Finally, IMU measurements 201 that have been segmented into a constant-velocity part (CVP) is provided to CVP module 208, which then proceeds to compute CVP factors based on data in the CVP and attitude constraints as generated by sub-module 218. HCTO module 104 then gathers the HFP, LFP and CVP factors to form IMU hybrid constraints 105.

[0035] Exemplary measured IMU data, which comprises accelerometer outputs, are illustrated in Figure 3. It can be observed from plots 302, 304, 306 that there is a repetitive motion pattern for each step as seen by the accelerometer measurements in the x, y and z axes.

It should be noted that when the commonly used cubic Spline (4th order) was employed, it was only able to capture the linear changes of acceleration, and faced difficulties in representing the raw IMU measurements, especially during vibrations. This is illustrated in plot 308 which illustrates a zoomed view of part of plot 301 where the model was unable to fit the data well in certain parts of the plot. In particular, it can be seen that the low-frequency part (LFP) segment can be effectively modelled using a linear model, whereas the high-frequency part (HFP) segment may not. Accordingly, the trajectory is partitioned into two parts, the HFP and LFP, by motion state recognition module 202 (as shown in Figure 2), where these two parts are delineated by a Root Mean Squared Error (RMSE) σ_{motion} of acceleration measurements within the B-Spline time interval Δt , as derived from linear fitting residuals. If module 202 determines that σ_{motion} of a time segment is a predetermined number of times larger, e.g., three times larger, than the accelerometer measurement error, this time segment is determined as an HFP. The predetermined number may be selected as “three” as this is based on the well-known three-sigma rule. Conversely, if it is determined that σ_{motion} of a time segment is not larger than the predetermined number of time, i.e., it is not three times larger, than the accelerometer measurement error, this time segment is classified as a LFP as the acceleration measurements can be fitted well using a linear model. Additionally, the initial guess of acceleration $\mathbf{a}^\omega(t)$ in the world frame may be obtained based on pure IMU integration at any time t . Hence, a time segment at which the time $\|\mathbf{a}^\omega(t)\| = 0$ will be classified as the CVP.

[0036] In other words, it can also be said that module 202 segments the sequential data from IMU module 103 into individual segments according to a B-Spline time interval Δt . Module 202 then integrates and normalizes acceleration data within each individual segment, whereby an individual segment having normalized integrated acceleration data equal to a zero value is classified as the constant velocity part (CVP). Module 202 then proceeds to fit acceleration measurements contained within each unclassified individual segment to a linear model, and compute a root mean square error (RMSE) for each of the individual segments based on linear fitting residuals. An individual segment is then classified as the low-frequency part (LFP) when it is determined that the RMSE computed for the individual segment is equal to or less than three times an accelerometer measurement error of the IMU, and/or an individual segment is classified as the high-frequency part (HFP) when it is determined that the RMSE computed for the individual segment is more than three times the accelerometer measurement error.

[0037] **Computation of Constant Velocity Part (CVP) Factors**

[0038] The computation of the CVP factors take place in CVP module 208 based on parameters obtained from attitude constraint module 218. Based on the principles behind complementary filters for the estimation of orientation, a force equation of a system may be defined as:

$$\mathbf{R}_b^\omega(t) \left(\mathbf{a}^b(t) + \mathbf{b}_a(t) \right) + \mathbf{g}^\omega = \mathbf{a}^\omega(t) \quad \dots \text{equation (10)}$$

[0039] When $\|\mathbf{a}^\omega(t)\| = 0$, the relationship between the states and the gravity may be defined as follows:

$$\mathbf{R}_b^\omega(t) \left(\mathbf{a}^b(t) + \mathbf{b}_a(t) \right) = -\mathbf{g}^\omega, \text{ if } \|\mathbf{a}^\omega(t)\| = 0 \quad \dots \text{equation (11)}$$

[0040] The condition $\|\check{\mathbf{a}}^\omega(t)\| < 0.3$ is used as a trigger for adding a CVP factor to equation (10) where $\check{\mathbf{a}}^\omega(t)$ is defined as the IMU-predicted acceleration in the world frame, i.e., equation (11). The CVP factor is used to correct the roll and pitch of the state estimate. Under the assumption that $\delta\theta(t)$ is defined as the rotation correction corresponding to an initial guess $\mathbf{R}_b^\omega(t)$, equation (11) above can then be redefined as:

$$-\mathbf{R}_b^\omega(t)^T \left(I - [\delta\hat{\theta}(t)]_\times \right) \mathbf{g}^\omega = \check{\mathbf{a}}^b(t) + \mathbf{b}_a(t) \quad \dots \text{equation (12)}$$

[0041] Based on the above, CVP module 208 then computes the CVP factors r_t^{cv} at time t as:

$$r_t^{cv} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \delta\hat{\theta}(t)_0 \\ \delta\hat{\theta}(t)_1 \end{bmatrix} + \frac{1}{\|\mathbf{g}^\omega\|} \left[\left(\mathbf{R}_b^\omega(t)^T \check{\mathbf{a}}^b(t) + \mathbf{b}_a(t) \right) + \mathbf{R}_b^\omega(t)^T \mathbf{g}^\omega \right]_{1:2} \quad \dots \text{equation (13)}$$

where $\delta\hat{\theta}(t)_0$ and $\delta\hat{\theta}(t)_1$ are defined as the roll and pitch corrections respectively.

[0042] **Computation of High Frequency Part (HFP) Factors**

[0043] As illustrated in plot 308 of Figure 3, the high-frequency part of motion may not be fitted using 4th order B-Spline, which only guarantees the linear changing of acceleration and may not be used to accurately represent the high-frequency motion. Consequently, naively fitting this will lead to a large error and unstable results. Increasing the order of the B-Spline or decreasing the Δt , namely increasing the parameter of the trajectory, could be one solution but such an approach will result in a greatly increased computational load and uncertainty of the system. In this disclosure, instead of increasing the parameter to fit the vibrated motion, the raw IMU measurements are intergrade and HPF factors are computed to constrain the relative motion between the start and end of the high frequency motion, which alleviates the effect of the high-frequency part on sequential B-Spline segments.

[0044] Under the assumption that a high-frequency part has a period $[t_i, t_{i+1}]$ with a duration Δt , the IMU measurements in $[t_i, t_{i+1}]$ are integrated to obtain the position, velocity, and rotation elements (i.e., $\Delta \mathbf{p}_{i+1}^i$, $\Delta \mathbf{v}_{i+1}^i$, $\Delta \mathbf{q}_{i+1}^i$ respectively) of IMU pre-integration according to equation (4). The HFP factors $r_t^{\mathcal{H}F}$ may then be computed by HFP module 204 as:

$$r_t^{\mathcal{H}F} = \begin{bmatrix} \Delta \mathbf{p}_{i+1}^i \\ \Delta \mathbf{v}_{i+1}^i \\ \Delta \mathbf{q}_{i+1}^i \end{bmatrix} - \begin{bmatrix} \mathbf{R}_b^\omega(t_i)^\top \left(\mathbf{p}_b^\omega(t_{i+1}) - \mathbf{p}_b^\omega(t_i) \right) \\ -0.5 \mathbf{g}^\omega \Delta t^2 - \dot{\mathbf{p}}_b^\omega(t_i) \Delta t \\ \mathbf{R}_b^\omega(t_i)^\top (\dot{\mathbf{p}}_b^\omega(t_{i+1}) - \mathbf{g}^\omega \Delta t - \dot{\mathbf{p}}_b^\omega(t_i)) \\ \mathbf{R}_b^\omega(t_i)^\top \mathbf{R}_b^\omega(t_{i+1}) \end{bmatrix} \quad \dots \text{equation (14)}$$

It should be noted that the system's velocity is extracted from the differential of the B-Spline trajectory, setting it apart from the conventional DTO approach and this ensures local trajectory consistency. In the high-frequency part, unlike in traditional CTO approaches, the B-Spline trajectory does not attempt to fit the oscillated acceleration measurements, thereby facilitating stable and swift convergence. Additionally, the step of integrating the raw IMU measurements in the high frequency part can be regarded as applying a special low-pass filter to the raw IMU measurements as this integration effectively suppresses the peak values attributed to jitter. However, the HPF factor described in this disclosure is different from mere low-pass filtering as it incorporated the IMU measurements according to the strapdown IMU integration principle, ensuring the relative motion relationship remains consistent from the beginning to the end of the high-frequency motion.

[0045] **Computation of Low Frequency Part (LFP) Factors**

[0046] For the low-frequency portion of the motion, all IMU measurements are directly employed in the B-Spline parameter estimation by LFP module 206 to utilize all the available information. Under the assumption that the low-frequency part has a period $[t_i, t_{i+1}]$ with a duration Δt , LFP module 206 may compute the LFP factor $r_t^{\mathcal{LF}}$ for each raw IMU measurement as:

$$r_t^{\mathcal{LF}} = \begin{bmatrix} \ddot{\mathbf{p}}_b^\omega(t) \\ \dot{\mathbf{R}}_b^\omega(t) \end{bmatrix} - \begin{bmatrix} \mathbf{R}_b^\omega(t) (\ddot{\mathbf{a}}^b(t) + \mathbf{b}_a(t)) + \mathbf{g}^\omega \\ \ddot{\omega}^b(t) + \mathbf{b}_g(t) \end{bmatrix} \quad \dots \text{equation (15)}$$

[0047] IMU hybrid constraints 105 comprising the CVP $r_t^{\mathcal{CV}}$, HFP $r_t^{\mathcal{HF}}$ and LFP $r_t^{\mathcal{LF}}$ factors may then be obtained within a LiDAR frame duration (i.e., 0.1s in most scanning configurations) from equations (13) – (15) respectively. IMU optimization module 216 then performs an IMU only optimization based on the IMU hybrid constraints 105 to compute an initial guess of system states $\tilde{\mathbf{X}}(t)$ 107 which is then used to correct LiDAR motion distortion. It should be noted that the IMU only optimization is performed based only on the constraints defined in equations (14) and (15).

[0048] With reference to Figure 1, motion un-distortion module 108 and optimal-aware point feature association module 110 then combine to utilize the initial guess of system states $\tilde{\mathbf{X}}(t)$ 107 and data contained in the active LiDAR frame to generate an optimal set of LiDAR correspondences 111.

[0049] In particular, points in the current active LiDAR frame are projected into the world map coordinate based on the initial guess of system states $\tilde{\mathbf{X}}(t)$ 107. This is to align the newly captured points of the active LiDAR frame with the existing map. For a LiDAR measurement $\mathbf{f}_t$, its closest points are obtained from the map points and their average value μ and plane normal $\mathbf{n}$ are subsequently computed, where these computed values represent an orientation of a surface formed by these points. In embodiments of the disclosure, the map points are maintained using an incremental k-dimensional tree (ikdtree) as provided in keyframe selection module 114. A point-to-plane factor $r_t^{\mathcal{L}}$ is used to represent a relationship between the points in the current LiDAR frame and map's surfaces. The point-to-plane factor $r_t^{\mathcal{L}}$ may then be computed based on a difference in value between a LiDAR measurement in the active LiDAR

frame and a reference surface on an existing LiDAR map, and can be specifically defined as follows:

$$r_t^{\mathcal{L}} = \mathbf{n}^T(\mathbf{R}_b^{\omega}(t)\mathbf{f}_t + \mathbf{p}_b^{\omega}(t) - \mu) \quad \dots \text{equation (16)}$$

[0050] A set of point-to-plane factors in the current LiDAR frame may be represented as $S_{\mathcal{L}} = \{r_t^{\mathcal{L}}\}$, and there are usually more than several thousand LiDAR correspondences in each frame. MAP optimization module 112 performs a MAP optimization on these correspondences in a sliding window, which includes the most recent frames. However, optimizing such a large number of correspondences leads to heavy computational loads for B-Spline estimation and may result in unforeseen degeneration caused by the uneven distribution of the correspondences. To mitigate these issues, module 110 has to carefully select an optimal subset of correspondences $S_{\mathcal{L}}^O \subset S_{\mathcal{L}}$ for the state estimation as this not only improves time efficiency but also odometry accuracy by focusing on well-distributed and high-quality correspondences. The process of selecting this optimal subset is also known in the art as an optimal design.

[0051] As the system's states $\mathcal{X}$ are presented as k^{th} order B-Spline, there will be k knots related to the constant $r_t^{\mathcal{L}}$. As the center knot is the most correlated with the current LiDAR frame according to the B-Spline definition, only the center knot located in the current LiDAR frame time duration is considered for the following optimal analysis to guarantee real-time performance. The constant $r_t^{\mathcal{L}}$ is then linearized and its Jacobian w.r.t $\mathcal{X}$ is calculated as $\mathbf{J}_t = \partial r_t^{\mathcal{L}} / \partial \mathcal{X} |_{\mathcal{X}=\tilde{\mathcal{X}}}$. The information matrix Λ for $S_{\mathcal{L}}^O$ may then be defined as follows:

$$\Lambda = \sum_{\mathcal{L}_t \in S_{\mathcal{L}}^O} \mathbf{J}_t^T \mathbf{J}_t \quad \dots \text{equation (17)}$$

[0052] The information matrix Λ reveals the uncertainties for the variables to be estimated and also the optimality of the subset $S_{\mathcal{L}}^O$. Specifically, the minimum eigenvalue $\lambda_{\min}(\Lambda)$, which is a commonly used metric, is utilized to qualify the optimality of information matrix Λ . The minimum eigenvalue was selected as the metric because it restricts the possible degenerated direction. Hence, in order to achieve the most optimal design for the knots estimation, the objective function $g(S_{\mathcal{L}}^O)$ is thus maximized:

$$\begin{aligned} \arg \max g(S_{\mathcal{L}}^O) \text{ s.t. } S_{\mathcal{L}}^O \subset S_{\mathcal{L}}, |S_{\mathcal{L}}^O| \leq N^O, \\ g(S) = \lambda_{\min}[\Lambda(S)] \end{aligned} \quad \dots \text{equation (18)}$$

where N^O is defined as the desirable size of correspondences remaining in the system when the computational load is taken into consideration. The objective function in equation (18) is a NP-hard problem and could be approximately solved by a stochastic-greedy heuristic. The stochastic-greedy algorithm starts with an empty set. Then, at each step, it selects one element from a random set with a size of $|S_{\mathcal{L}}|/N^O \log(1/\epsilon)$ from the remaining correspondence set, which gains most of the objective function $g(S_{\mathcal{L}}^O)$ where ϵ is the decay factor. The iterative process stops when N^O correspondences are achieved. It should be noted that the set function $g(S)$ is submodular and monotone increasing w.r.t. S . Additionally, $S_{\mathcal{L}}^{O*}$ and $S_{\mathcal{L}}^{O\#}$ are the optimal set and stochastic-greedy heuristic result respectively, and are defined as:

$$\frac{g(S_{\mathcal{L}}^{O*}) - g(S_{\mathcal{L}}^{O\#})}{g(S_{\mathcal{L}}^{O*})} \leq \left(\frac{N^O - 1}{N^O} \right)^{N^O} + \epsilon \leq \frac{1}{e} + \epsilon \quad \dots \text{equation (19)}$$

[0053] Equation 19 elicits a lower bound for the stochastic-greedy heuristic results and it was found that the stochastic-greedy heuristics were able to achieve a better result than the lower bound. It is useful to note that the time complexity of the stochastic heuristic is $O(|S_{\mathcal{L}}| \log(1/\epsilon))$, which is related to $|S_{\mathcal{L}}|$ but independent of N^O .

[0054] In embodiments of the disclosure, the optimal set of LiDAR correspondences may be obtained as follows. Module 110 first computes mean values and variance values for each dimension of Jacobian matrices, wherein each Jacobian matrix is computed based on the LiDAR measurements received from the LiDAR module that have been linearized with regard to the initial guess of system states $\tilde{\mathbf{X}}(t)$ 107. The dimensions of the Jacobian matrices are then normalized by subtracting each dimension with a corresponding mean value and dividing a result of the subtraction with a corresponding variance value. The normalized Jacobian matrices are then segmented into sets of groups based on a high-dimensional voxelization defined by a predetermined voxel size. One or more rounds of stochastic-greedy selection are then performed to obtain the optimal set of LiDAR correspondences. In this embodiment, the stochastic-greedy selection is performed based on a set of LiDAR point-to-plane factors $r_t^{\mathcal{L}}$,

the Jacobian matrices, the sets of groups, and a minimum eigenvalue until a size of the optimal set of LiDAR correspondences is equal to a predetermined number of LiDAR correspondences.

[0055] In another embodiment of the disclosure, a group-based stochastic-greedy solver is used for selecting the best or optimal subset of LiDAR correspondences $S_{\mathcal{L}}^O$ and this approach is set out in Algorithm 1 below.

Algorithm 1: Group-Based Stochastic-Greedy for Feature Selection Considering Optimality

Input: $S_{\mathcal{L}}, \{\mathbf{J}_t\}, \epsilon, \lambda, N^O, D_v$
Output: Selected LiDAR correspondences $S_{\mathcal{L}}^O$

- 1 Initialization of the set $S_{\mathcal{L}}^O \leftarrow \emptyset$;
- 2 Calculate mean value $\{\mu_0, \mu_1, \dots, \mu_5\}$ and variance $\{\Sigma_0, \Sigma_1, \dots, \Sigma_5\}$ at each dimension for $\{\mathbf{J}_t\}$;
- 3 **foreach** $\mathbf{J}_t \in \{\mathbf{J}_t\}, \bar{\mathbf{J}}_t \in \{\bar{\mathbf{J}}_t\}$ **do**
- 4 $\bar{\mathbf{J}}_{t,0} = (\mathbf{J}_{t,0} - \mu_0)/\Sigma_0; \bar{\mathbf{J}}_{t,1} = (\mathbf{J}_{t,1} - \mu_1)/\Sigma_1$;
- 5 $\bar{\mathbf{J}}_{t,2} = (\mathbf{J}_{t,2} - \mu_2)/\Sigma_2; \bar{\mathbf{J}}_{t,3} = (\mathbf{J}_{t,3} - \mu_3)/\Sigma_3$;
- 6 $\bar{\mathbf{J}}_{t,4} = (\mathbf{J}_{t,4} - \mu_4)/\Sigma_4; \bar{\mathbf{J}}_{t,5} = (\mathbf{J}_{t,5} - \mu_5)/\Sigma_5$;
- 7 **end**
- 8 Divide $\{\mathbf{J}_t\}$ into groups $\{\mathbf{g}_k\}$ using $\{\bar{\mathbf{J}}_t\}$ as feature with voxel size of D_v ;
- 9 **while** $|S_{\mathcal{L}}^O| < N^O$ **do**
- 10 $\mathcal{G} \leftarrow$ subset obtained by random sampling $\lambda \log(1/\epsilon)$ groups from $\{\mathbf{g}_k\}$;
- 11 $\mathcal{R} = \{\mathbf{J}_j\} \leftarrow$ subset obtained the first element in each sampled group $\mathcal{G}$ with $\mathbf{J}_j \in \mathbf{g}_k$;
- 12 **foreach** $\mathbf{J}_j \in \mathcal{R}$ **do**
- 13 Compute $\Lambda_j = \mathbf{J}_j^T \mathbf{J}_j$;
- 14 **end**
- 15 $j^* \leftarrow \arg\max_{j \in \mathcal{R}} \lambda_{\min}[\Lambda(S_{\mathcal{L}}^O) + \Lambda_j]$;
- 16 $\Lambda(S_{\mathcal{L}}^O) \leftarrow \Lambda(S_{\mathcal{L}}^O) + \Lambda_{j^*}$;
- 17 $S_{\mathcal{L}}^O \leftarrow S_{\mathcal{L}}^O \cup \mathbf{J}_{j^*}^T; \mathbf{g}_k \leftarrow \mathbf{g}_k \setminus \mathbf{J}_{j^*}$.
- 18 **end**

[0056] To make the process more efficient, LiDAR correspondences $S_{\mathcal{L}}$ are grouped based on the similarity of their Jacobian vectors $\mathbf{J}_t$. This ensures similar candidate correspondences are not sampled again in each stochastic step. Each Jacobian vector $\mathbf{J}_t$, which may be represented as a 6-dimensional feature (with 3 dimensions for rotation in the form of a quaternion and 3 for translation), contributes to the objective function in similar ways when they are close in value. By regularizing each dimension of the Jacobian $\mathbf{J}_t$ to obtain $\bar{\mathbf{J}}_t$, the

module balances the numerical ranges of the quaternion and translation parts, which helps to group similar Jacobians together. The regularization of these Jacobians take place at Lines 2–7 of Algorithm 1.

[0057] The module then uses a high-dimensional voxelization approach to group the regularized Jacobians $\bar{\mathbf{J}}_t$, into voxel-based clusters, with a specified voxel size D_v to obtain λN^O groups as illustrated in Figure 4. This method effectively segments the LiDAR correspondences into groups by considering the regularized Jacobians $\bar{\mathbf{J}}_t$, thus speeding up the stochastic-greedy search. By storing voxel indexes in a hash table, the grouping process has a linear complexity of $O|S_{\mathcal{L}}|$, making it computationally efficient even when handling large datasets. During each greedy searching step, a random number of candidates $\lambda \log(1/\epsilon)$ are sampled from each group, as opposed to a larger number that would traditionally be used. The module then selects the best candidate (other than $|S_{\mathcal{L}}|/N^O \log(1/\epsilon)$ candidates in the original stochastic-greedy search process) from these groups, and the process continues iteratively until the desired number of optimal correspondences N^O is reached. This iterative process is set out at Lines 9-18 in Algorithm 1.

[0058] To determine the voxel size D_v , a lazy counting strategy (LCS) is proposed. The detailed workings of the LCS approach is omitted for brevity as it is known to one skilled in the art. The LCS first down samples the current candidates with farthest point sampling to the size of λN^O . Then the mean distance between the remaining candidates is calculated to obtain D_v . As the feature distribution in the environment does not dramatically change every frame, LCS is only triggered to update the voxel size D_v when the size of the grouping results differ significantly (20% in practice) from λN^O .

[0059] Once the optimal set of LiDAR correspondences 111 have been obtained, it is then provided to MAP optimization module 112 which performs a maximum-a-priori (MAP) optimization using the optimal set of LiDAR correspondences, the hybrid IMU constraints and prior constraints $\mathbf{r}_t^{\mathcal{M}}$ to obtain the set of optimized estimated states $\hat{\mathbf{X}}(t)$ 113 for the active LiDAR frame as illustrated in Figure 5. The MAP optimization equation is defined as follows:

$$f(\hat{\mathcal{X}}) = \sum_{t \in \Phi} \|r_t^{\mathcal{H}\mathcal{F}}\|_{\Sigma^{\mathcal{H}\mathcal{F}}}^2 + \sum_{t \in \Phi} \|r_t^{\mathcal{L}\mathcal{F}}\|_{\Sigma^{\mathcal{L}\mathcal{F}}}^2 + \sum_{t \in \Phi} \|r_t^{\mathcal{C}\mathcal{V}}\|_{\Sigma^{\mathcal{C}\mathcal{V}}}^2 + \sum_{t \in \Phi} \|r_t^{\mathcal{L}}\|_{\Sigma^{\mathcal{L}}}^2 + \|r_t^{\mathcal{M}}\|_{\Sigma^{\mathcal{M}}}^2$$

...equation (20)

[0060] This equation involves a weighted sum of several error terms, each corresponding to different factors affecting the system's state. Specifically, it includes the IMU hybrid constraints, i.e., the high-frequency part (HFP) factor $r_t^{\mathcal{H}\mathcal{F}}$, the low-frequency part (LFP) factor $r_t^{\mathcal{L}\mathcal{F}}$, the constant velocity part (CVP) factor $r_t^{\mathcal{C}\mathcal{V}}$, the LiDAR factor $r_t^{\mathcal{L}}$, and the prior constraint $r_t^{\mathcal{M}}$. These factors computed at time t for all data points in the active sensor data duration $\Phi = [t_i, t_{i+m}]$, are weighted by their corresponding covariance matrices Σ which reflect the uncertainty associated with the measurements. Additionally, the term $r_t^{\mathcal{M}}$ represents the prior constraint for the first $k-1$ knots in the active window, and this term is used to marginalize out past states from the sliding window and maintain efficiency without losing track of prior information.

[0061] In each optimization step, when a new LiDAR frame is received, the system formulates the optimization problem using k^{th} -order B-Spline to represent the trajectory of the system. The trajectory segment during the time window $[t_i, t_{i+m}]$ is influenced by a set of knots, $[\mathcal{X}_i, \dots, \mathcal{X}_{i+k-1}]$ defined as active nodes (as illustrated in Figure 5) that are to be optimized within the sliding window. This ensures that the optimization focuses on the relevant segment of the trajectory and the sliding window technique balances the computational load by limiting the number of knots being optimized at any one time.

[0062] To achieve better accuracy and to handle the system's dynamics, each factor's covariance matrix is carefully designed. For instance, $\Sigma^{\mathcal{C}\mathcal{V}}, \Sigma^{\mathcal{H}\mathcal{F}}, \Sigma^{\mathcal{L}\mathcal{F}}, \Sigma^{\mathcal{L}}$ and $\Sigma^{\mathcal{M}}$ are the covariance matrices corresponding to the factors. $\Sigma^{\mathcal{C}\mathcal{V}}, \Sigma^{\mathcal{H}\mathcal{F}}, \Sigma^{\mathcal{L}\mathcal{F}}$ are related to the raw IMU measurements and account for the IMU's inherent noise, while $\Sigma^{\mathcal{L}}$ is based on the accuracy of the LiDAR sensor (typically around 3 cm for Livox LiDAR systems) and $\Sigma^{\mathcal{M}}$ is set according to the marginalized prior constraint – for minimizing errors when older knots are removed from the optimization window. It should be noted that the IMU measurement is propagated to obtain the initial poses while the system states presented in knots of B-Spline are initialized with IMU-only optimization.

[0063] Once the optimized estimated states 113 are obtained, keyframe selection module 114 then selects a new keyframe when the system experiences a significant change in translation or orientation that exceeds a certain threshold. This ensures that the keyframe captures a meaningful state of the system. The selected keyframes are then stored in a keyframe database provided within module 114, which contains keyframes distributed over time and space, representing important moments in the system's movement. The points in each keyframe are integrated into the local map, which is managed using an ikdtree (incremental k-d tree) for efficient storage and retrieval of spatial data.

[0064] In accordance with embodiments of the present disclosure, a block diagram representative of components of processing system 600 that may be provided within computing module 102, HCTO module 104 and/or any of the modules shown in Figures 1 and 2 to carry out the computing and processing functions in accordance with embodiments of the disclosure. One skilled in the art will recognize that the exact configuration of each processing system provided within these modules may be different and the exact configuration of processing system 600 may vary and the arrangement illustrated in Figure 6 is provided by way of example only.

[0065] In embodiments of the disclosure, processing system 600 may comprise controller 601 and user interface 602. User interface 602 is arranged to enable manual interactions between a user and the computing module as required and for this purpose includes the input/output components required for the user to enter instructions to provide updates to each of these modules. A person skilled in the art will recognize that components of user interface 602 may vary from embodiment to embodiment but will typically include one or more of display 640, keyboard 635 and optical device 636.

[0066] Controller 601 is in data communication with user interface 602 via bus 615 and includes memory 620, processing unit or processor 605 mounted on a circuit board that processes instructions and data for performing the method of this embodiment, an operating system 606, an input/output (I/O) interface 630 for communicating with user interface 602 and a communications interface, in this embodiment in the form of a network card 650. Network card 650 may, for example, be utilized to send data from these modules via a wired or wireless network to other processing devices or to receive data via the wired or wireless network.

Wireless networks that may be utilized by network card 650 include, but are not limited to, Wireless-Fidelity (Wi-Fi), Bluetooth, Near Field Communication (NFC), cellular networks, satellite networks, telecommunication networks, Wide Area Networks (WAN) and etc.

[0067] Memory 620 and operating system 606 are in data communication with processor 605 via bus 610. The memory components include both volatile and non-volatile memory and more than one of each type of memory, including Random Access Memory (RAM) 623, Read Only Memory (ROM) 625 and a mass storage device 645, the last comprising one or more solid-state drives (SSDs). One skilled in the art will recognize that the memory components described above comprise non-transitory computer-readable media and shall be taken to comprise all computer-readable media except for a transitory, propagating signal. Typically, the instructions are stored as program code in the memory components but can also be hardwired. Memory 620 may include a kernel and/or programming modules such as a software application that may be stored in either volatile or non-volatile memory.

[0068] Herein the term “processor” or “processing unit” is used to refer generically to any device or component that can process such instructions and may include: a microprocessor, a processing unit, a microcontroller, a programmable logic device or other computational device. That is, processor 605 may be provided by any suitable logic circuitry for receiving inputs, processing them in accordance with instructions stored in memory and generating outputs (for example to the memory components or on display 640). In this embodiment, processor 605 may be a single core or multi-core processor with memory addressable space. In one example, processor 605 may be multi-core, comprising—for example—an 8 core CPU. In another example, it could be a cluster of CPU cores operating in parallel to accelerate computations.

[0069] A flowchart which sets out the process for generating a set of optimized estimated states for generating a motion-corrected LiDAR point cloud map in accordance with embodiments of the present disclosure is illustrated in Figure 7. In embodiments of the disclosure, process 700 as illustrated in Figure 7 may be performed by computing module 102 or any combination of modules provided within computing module 102.

[0070] Process 700 begins at step 702 with process 700 segmenting sequential IMU data obtained from an IMU module into a low-frequency part, a high-frequency part and a constant

velocity part. Process 700 then proceeds to compute hybrid IMU constraints based on data contained within the low-frequency part, the high-frequency part and the constant velocity part. This takes place at step 704. At step 706, process 700 then computes an initial guess of system states for an active LiDAR frame by performing an IMU optimization process based on the hybrid IMU constraints. Subsequently, at step 708, process 700 computes an optimal set of LiDAR correspondences based on measurements of the active LiDAR frame and the initial guess of system states. At step 710, process 700 computes the set of optimized estimated states for the active LiDAR frame by performing a maximum-a-priori (MAP) optimization process using the optimal set of LiDAR correspondences, the hybrid IMU constraints, and prior constraints. In embodiments of the disclosure, at step 712, process 700 then goes on to generate motion corrected LiDAR point clouds based on the set of optimized states generated at step 710.

[0071] In embodiments of the disclosure, during the process of segmenting the sequential data into the low-frequency part (LFP), the high-frequency part (HFP) and the constant velocity part (CVP), process 700 further segments the sequential data into individual segments according to a B-Spline time interval. Process 700 then integrates and normalizes acceleration data within each individual segment, whereby an individual segment having normalized integrated acceleration data equal to a zero value is classified as the constant velocity part before fitting acceleration measurements contained within each unclassified individual segment to a linear model, and computing a root mean square error (RMSE) for each of the individual segments based on linear fitting residuals. Process 700 then classifies an individual segment as the low-frequency part when it is determined that the RMSE computed for the individual segment is equal to or less than three times an accelerometer measurement error of the IMU and classifies an individual segment as the high-frequency part when it is determined that the RMSE computed for the individual segment is more than three times the accelerometer measurement error.

[0072] In embodiments of the disclosure, during the process of computing the hybrid IMU constraints, process 700 further computes the CVP factors based on CVP parameters derived during a time interval of the CVP, the CVP parameters comprising roll and pitch correction factors, CVP rotation matrices defining an orientation of a body frame in a world frame, CVP acceleration parameters in the body frame, CVP IMU accelerometer biases, and a gravity vector in the world frame.

[0073] The IMU pre-integration parameters comprise pre-integrated position parameters, pre-integrated velocity parameters and pre-integrated rotation parameters

[0074] In embodiments of the disclosure, during the process of computing the hybrid IMU constraints, process 700 further computes the HFP factors based on HFP parameters derived during a time interval of the HFP, the HFP parameters comprising IMU pre-integration parameters, HFP rotation matrices defining an orientation of the body frame in the world frame, positions and velocities of the body frame in the world frame, the gravity vector in the world frame and the B-Spline time interval Δt . The IMU pre-integration parameters comprise pre-integrated position parameters, pre-integrated velocity parameters and pre-integrated rotation parameters.

[0075] In embodiments of the disclosure, during the process of computing the hybrid IMU constraints, process 700 further computes the LFP factors based on LFP parameters derived during a time interval of the LFP, the LFP parameters comprising accelerations and angular velocities of the body frame in the world frame, LFP rotation matrices defining an orientation of the body frame in the world frame, LFP acceleration parameters in the body frame, LFP IMU accelerometer biases, the gravity vector in the world frame and gyroscope parameters in the body frame.

[0076] In summary, applying existing LIO methods (such as the state of the art Fast-Lio2 model) to compact WMS faces two primary challenges. First, human motion vibrations are difficult to model using low-order splines, such as the commonly used 4th order spline. While increasing the spline order or adding more knots could address this, it would significantly increase computational load and uncertainty in the system. Second, uneven or degenerated LiDAR feature correspondences can lead to the matching process converging on a local minimum, resulting in long-term drift as shown in plot 801 of Figure 8. These issues cause attitude and Z-direction drift leading to distorted "bent" point clouds such as that illustrated in plot 802. Such "bent" point clouds are locally consistent but unsuitable for use as prior maps in planning and navigation systems, such as for controlling the navigation systems of delivery robots. It should be noted that plots 801 and 802 were generated based on the Fast-Lio2 model.

[0077] The proposed LIO with HCTO as described in this disclosure is suitable for real-time point cloud mapping as it separates the Low-Frequency Part (LFP) and High-Frequency Part (HFP) of human motion using sequential IMU measurements. Raw IMU constraints are then applied for the computation of the LFP factors and IMU pre-integration constraints are used for the computation of the HFP factors to handle severe vibrations. This allows efficient IMU fusion without increasing system variables or uncertainty. The proposed LIO with HCTO system also identifies the Constant Velocity Part (CVP) from IMU data to correct attitude drift using CVP constraints on the spline trajectory. Additionally, the LIO with HCTO system features an optimal design-based feature selection scheme with a group-based stochastic-greedy solver, which improves real-time performance and odometry accuracy, particularly in degenerated environments. The effectiveness of the LIO with HCTO is can be seen from plot 803 which illustrates a motion-corrected LiDAR point cloud that has been generated using the LIO with HCTO system described in this disclosure.

[0078] **Embodiment of the LIO with HCTO based on a wearable mapping system**

[0079] In embodiments of the disclosure, the performance of the LIO with HCTO system is described using a public wearable sensing dataset (WHU-Helmet) as disclosed in “WHU-helmet: A helmet-based multi-sensor SLAM dataset for the evaluation of real-time 3D mapping in large-scale GNSS-denied environments. IEEE Trans. Geosci. Remote Sens.” and using in-house wearable sensing datasets. The performance of the LIO with HCTO method is compared with existing state-of-the-art (SOTA) methods. The first SOTA method is Fast-Lio2, which uses filter-based state estimation with the ikdtree managing the whole map points in an incremental way. The second SOTA method is DLIO, which is a direct LIO using continuous-time motion un-distortion. The third SOTA method is SLICT, which estimates the system states using CTO. The fourth SOTA method is CLINS, which fuses high-frequency and asynchronous sensor data effectively. Video recordings of experiments performed based on the LIO with HCTO system may be found on the project page of HCTO: “<https://github.com/kafeiyin00/HCTO>”.

[0080] The proposed HCTO module was implemented in C++ and Robot Operating System (ROS) where the order of the B-Spline is set to 4. The time segment length Δt for the B-Spline was set to 0.05 and the decay factor ϵ for the stochastic-greedy was set to 0.1. The size of the

selected correspondences N^o was set to 500 and the grouping factor λ was set to 2 in order to obtain 1000 (λN^o) correspondence groups in each LiDAR frame.

[0081] An additional in-house WHU-Helmet2 dataset was also collected using a helmet-based wearable system. However, the main difference in the hardware configuration between WHU-Helmet and the in-house built version WHU-Helmet2 is the type of LiDAR scanner used. The Livox Avia is used in WHU-Helmet, which has a longer observation range but with a limited sight view ($70.4^\circ \times 77.2^\circ$) as compared to the Livox Mid360 (with a sight view of $360^\circ \times 59^\circ$) which was used in the LIO with HCTO system of the present disclosure. The limited view also poses a challenge for LIO systems in indoor environments. For this experiment, two sequences were selected: the car park and subway station scenes, from the WHU-Helmet dataset to evaluate the performance of the proposed method.

[0082] Both car parks and subway stations are common scenes, which need prior maps for robot-based delivery systems. For the evaluation of the performance of the various methods, an Absolute Translation Error (ATE) was used as the measurement metric and the results are listed in Table 1 below.

Table 1
ATE of HCTO and other methods on public WHU-Helmet Dataset (Unit [m]). The best results are in **BOLD**, second best results are underlined, x denotes diverge.

Dataset	Fast-Lio2	DLIO	SLICT	CLINS	HCTO (Ours)
seq-car-park (~0.6 km, W)	x	x	x	<u>1.16</u>	0.504
seq-subway-station (~0.8 km, W)	<u>3.55</u>	x	x	3.94	1.410

W for Walking.

[0083] As the two scenarios contained too many degenerated scenes like the corridor and elevator passage, most of the existing methods diverged. The localization error of seq-subway-station in WHU-Helmet for the proposed HCTO is plotted in Figures 9 and 10. From these plots, it can be seen that the proposed LIO with HCTO system was able to achieve good performance in these degenerated scenes. The point cloud maps constructed by the proposed LIO with HCTO system for the subway station and car park are illustrated in Figure 11. Based on a simple visual inspection, it can be said that the point cloud maps constructed by the proposed LIO with HCTO system achieved high accuracy and did not diverge in the narrow

corridors and elevator passage. The experimental results demonstrated the potential for the proposed method to be used to generate the prior map in car parks and subway stations.

[0084] In-house University Campus Dataset

[0085] A university campus dataset, comprising two study sites are shown in Figure 12. The dataset was collected using the in-house built compact helmet system where Site 1 - 1201 is an open-road environment that covers an area of about $500\text{ m} \times 300\text{ m}$ while Site 2 - 1202 is a multi-level indoor environment that covers an area of $400\text{ m} \times 100\text{ m}$. The two study sites represent common environments where robot delivery systems may be employed. For this experiment, the ground truth was constructed in a similar manner to the Newer College Dataset ("The newer college dataset: Handheld lidar, inertial and vision with ground truth. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 4353–4360"). A centimeter-level terrestrial laser scanner (Leica MS60) was utilized to obtain the prior map at the study sites. To ensure the accuracy of the prior map, optical prisms were first evenly set as the control points in the environment. Then a standard geodetic network adjustment was applied for the terrestrial laser scanning registration. Each LiDAR scan from the wearable sensing system was then registered to the prior map to obtain the ground truth pose.

[0086] In Site 1, three data sequences (seq-01, seq-02, and seq-03) were collected by a user of the system under varying degrees of motion: walking (about 1.5 m/s), running (about 2.5 m/s), and a combination of walking and running, respectively. The trajectory length of each sequence is about 1500 m . When the user was running, the data sequence contained more vibration than the walking sequence. Visualization of the point clouds generated by the LJO with HCTO system in Site 1 is shown in Figure 13a. Based on this point cloud, the features for a random LiDAR frame is generated and it was found that only 500 high-quality features are retained from an initial set of 4,825 features. This reduction ensures that the system maintains real-time performance while preserving the most relevant information for accurate mapping and state estimation.

[0087] Similar to previous experiments, ATE was used as the metric to evaluate the accuracy of the different methods. The trajectories and localization errors from different

methods are plotted in Figure 14 and the final results are listed in Table 2 below. As the operator moved more gently in the first sequence, most methods achieved better results compared to the other two sequences. As the operator ran to collect the seq-02, all the methods suffered from the vibration. Table 2 also illustrated that the proposed LIO with HCTO system achieved the best performance for all three sequences in site 1.

Table 2

ATE of HCTO and other methods on in-house NTU-Campus Site (Unit [m]). The best results are in **BOLD**, second best results are underlined. X denotes diverge.

Dataset	Fast-Lio2	DLIO	SLICT	CLINS	HCTO (Ours)
site1-seq-01 (~1.5 km, W)	5.103	5.437	<u>4.383</u>	5.462	2.059
site1-seq-02 (~1.5 km, R)	<u>5.262</u>	5.503	5.547	6.337	3.446
site1-seq-03 (~1.5 km, W&R)	5.228	5.759	4.452	<u>3.406</u>	3.105
site2-seq-01 (~1.0 km, W)	2.528	2.548	x	<u>1.823</u>	0.551

W for walking; R for running; W&R for a combination of walking and running.

[0088] In Site 2, the user of the system walked (about 1.5 m/s) to collect data sequence (seq-01) in a very complicated indoor environment, i.e., an Auditorium in a university, which contains multiple levels and degenerated corridor scenes. Visualization of the point clouds generated by the LIO with HCTO system of Site 2 is shown in Figure 13b. The trajectories and localization errors from different methods are plotted in Figure 15. From the results, it can be seen that the proposed LIO with HCTO system was able to achieve trajectory accuracy, especially in the Z direction. The APE of the trajectory errors were also listed in Table 2 above. Due to the narrow corridors in the environment, SLICT diverged and did not get a valid result.

[0089] The quality of the results obtained by the different methods were also evaluated by visual inspection of the walls and ground constructed in the point map. The ground and the walls in the real scenes were confirmed to be level and coincident with the gravity direction respectively. The visual comparisons between point clouds generated by different methods are illustrated in Figure 16. From the visual inspection, only the point clouds generated by the proposed LIO with HCTO system, i.e., plot 1605, managed to achieve the leveling of the ground and straight-up walls, which is vital for a prior map used for a robot delivery system. Otherwise, the robot in the system would not be able to achieve the correct goal in a bent prior map. In one of the plots, i.e., plot 1604, the CVP factor and feature selection model in the HCTO module was disabled. This plot 1604 showed some attitude drift from the visual inspection. The main reason for the bent map constructed by the LIO methods 1601-1603 is

that the unevenly distributed correspondences and degenerated scenes made the LIO systems converge to the local minimal solution and lose the observation of attitude, especially under the high vibration. However, the proposed LIO with HCTO system used the CVP factor to maintain the correct attitude and select good features to balance the unevenly distributed correspondences, thus achieving the best performance as shown in plot 1605. It also should be noted that due to the field of view limitation of the MiD360 on the helmet, the ground may not be observed in the LiDAR frame. Thus, the strategy of using ground points to restrict the drift can be difficult to be directly applied to the helmet system.

[0090] Apartments in hospitals and nursing homes are typical scenes that require robot delivery systems. These apartments usually contain a lot of degenerated scenes like staircases and long corridors, which pose challenges especially when the platform is moving. The LIO with HCTO system is validated in a multi-level apartment from outside to indoor as shown in Figure 17. To collect the data, the user of the system ran from the first floor to the top floor through the staircase as shown in illustration 1701. As for the inside of the apartment, the user walked through the long corridor as shown in illustration 1702. Most existing LIO systems diverged when the operator of such systems ran into the staircase or walked in the corridor. However, the proposed LIO with HCTO system was still able to achieve pretty good results. It should be noted that it took the system about 15 minutes to construct the prior map for the multi-level apartment using the wearable system with the proposed LIO with HCTO system, which is much more efficient and cost-effective than existing survey solutions.

[0091] The computational load of the LIO with HCTO system when the university campus Site-2 was generated was analyzed when the experiment was run on a computer with an Intel Core i9-12900 CPU. It was found that on average, it took 72.16 ms (Δt_{sum}) to complete one LiDAR frame procession, in which the pure IMU processing and LiDAR un-distortion took 9.17 ms ($\Delta t_{IMU+undist}$), the LiDAR feature association took 7.34 ms ($\Delta t_{matching}$), the feature selection and state estimation took 55.65 ms (Δt_{solve}), which guaranteed the real-time performance (10 Hz) of the proposed system. The memory usage and time performance for all the sequences were also recorded. To investigate the contribution of each design feature in the LIO with HCTO system's performance, different configurations of the HCTO were set and the results were compared on the helmet-based dataset with ground truth trajectories, and the results are listed in Table 3 below. It should be noted that the hybrid IMU factors were disabled

and the original IMU factors (same as the LFP) were used for all motion states, whose results were listed in the first column. The CVP factors are disabled and the results are listed in the second column. Lastly, the feature selection module was disabled and all the correspondences were retained for optimization without consideration of the time performance, and these results are listed in the third column.

Table 3

ATE in different helmet-based sequences with ground truth (Unit [m]). The results with increased ATE compared with HCTO are in RED. The results with decreased ATE compared with HCTO are in RED. x denotes diverge.

Dataset	Sequence	HCTO (w/o HB)	HCTO (w/o CVP)	HCTO (w/o FS)	HCTO
WHU-Helmet	seq-car-park (~0.6 km, W)	x (!=0)	0.901 (+0.277)	0.688 (+0.313)	0.504
	seq-subway-station (~0.8 km, W)	1.832 (+0.422)	1.644 (+0.234)	2.751 (+1.941)	1.410
NTU-Campus	site1-seq-01 (~1.5 km, W)	5.575 (+3.536)	2.819 (+0.760)	2.013 (-0.546)	2.059
	site1-seq-02 (~1.5 km, R)	x (!=0)	3.580 (+0.134)	3.312 (-0.134)	3.446
	site1-seq-03 (~1.5 km, W&R)	4.603 (+1.498)	3.315 (+0.210)	2.893 (-0.212)	3.105

W for walking; R for running; W&R for a combination of walking and running

HB for hybrid IMU factors; CVP for constant velocity factor; FS for feature selection.

[0092] The first column of Table 3 illustrates that through the use of the hybrid IMU factors, the vibration effect could be suppressed. The high-frequency part cannot be fitted by the B-Spline trajectory, which may cause a large drift or even divergence. The second column of Table 3 illustrates that the CVP factor could limit the drift. From the results of university-campus dataset, the CVP factor managed to achieve a higher improvement in its performance when the operator was walking and this could be due to the fact that the CVP period may not exist when the operator was running. The third column of Table 3 illustrated that the feature selection module was able to achieve a performance improvement in degenerated scenes in the WHU-Helmet dataset. It should be noted that HCTO (w/o FS) could not perform in real-time as it retained all the correspondences.

[0093] In summary, based on the experimental results, it can be concluded that the hybrid IMU factors are effective in ensuring robustness when high vibration is present. The CVP factor can noticeably reduce the drift, and the feature selection scheme can be very effective in degenerative cases. In cases where feature selection reduces the accuracy, it is only a minor decrease that can be justified by the significant improvement in real-time performance.

[0094] Numerous other changes, substitutions, variations, and modifications may be ascertained by the skilled in the art and it is intended that the present application encompass all such changes, substitutions, variations, and modifications as falling within the scope of the appended claims.

CLAIMS:

1. A module for generating a set of optimized estimated states in real time comprising:
 - a processing unit; and
 - a non-transitory media readable by the processing unit, the media storing instructions that when executed by the processing unit causes the processing unit to:
 - segment sequential data received from an inertial measurement unit (IMU) into a low-frequency part, a high-frequency part and a constant velocity part;
 - compute hybrid IMU constraints based on data contained within the low-frequency part, the high-frequency part and the constant velocity part;
 - compute an initial guess of system states for an active LiDAR frame by performing an IMU optimization process based on the hybrid IMU constraints;
 - obtain an optimal set of LiDAR correspondences based on measurements of the active LiDAR frame and the initial guess of system states; and
 - compute the set of optimized estimated states for the active LiDAR frame by performing a maximum-a-priori (MAP) optimization process using the optimal set of LiDAR correspondences, the hybrid IMU constraints, and prior constraints.
2. The module according to claim 1, wherein the instructions that cause the processing unit to segment the sequential data into the low-frequency part, the high-frequency part and the constant velocity part further comprises instructions for directing the processing unit to:
 - segment the sequential data into individual segments according to a B-Spline time interval Δt ;
 - integrate and normalize acceleration data within each individual segment, whereby an individual segment having normalized integrated acceleration data equal to a zero value is classified as the constant velocity part;
 - fit acceleration measurements contained within each unclassified individual segment to a linear model, and compute a root mean square error (RMSE) for each of the individual segments based on linear fitting residuals;
 - classify an individual segment as the low-frequency part when it is determined that the RMSE computed for the individual segment is equal to or less than three times an accelerometer measurement error of the IMU; and

classify an individual segment as the high-frequency part when it is determined that the RMSE computed for the individual segment is more than three times the accelerometer measurement error.

3. The module according to claim 2, wherein the hybrid IMU constraints comprise constant velocity part (CVP) factors for the constant velocity part, high frequency part (HFP) factors for the high-frequency part and low frequency part (LFP) factors for the low-frequency part.
4. The module according to claim 3, wherein the instructions that cause the processing unit to compute the hybrid IMU constraints further comprises instructions for directing the processing unit to:
 - compute the CVP factors based on CVP parameters derived during a time interval of the CVP, the CVP parameters comprising roll and pitch correction factors, CVP rotation matrices defining an orientation of a body frame in a world frame, CVP acceleration parameters in the body frame, CVP IMU accelerometer biases, and a gravity vector in the world frame.
5. The module according to claim 4, wherein the instructions that cause the processing unit to compute the hybrid IMU constraints further comprises instructions for directing the processing unit to:
 - compute the HFP factors based on HFP parameters derived during a time interval of the HFP, the HFP parameters comprising IMU pre-integration parameters, HFP rotation matrices defining an orientation of the body frame in the world frame, positions and velocities of the body frame in the world frame, the gravity vector in the world frame and the B-Spline time interval Δt .
6. The module according to claim 5, wherein the IMU pre-integration parameters comprise pre-integrated position parameters, pre-integrated velocity parameters and pre-integrated rotation parameters.
7. The module according to claim 5 or 6, wherein the instructions that cause the processing unit to compute the hybrid IMU constraints further comprises instructions for directing the processing unit to:

compute the LFP factors based on LFP parameters derived during a time interval of the LFP, the LFP parameters comprising accelerations and angular velocities of the body frame in the world frame, LFP rotation matrices defining an orientation of the body frame in the world frame, LFP acceleration parameters in the body frame, LFP IMU accelerometer biases, the gravity vector in the world frame and gyroscope parameters in the body frame.

8. The module according to claim 1, wherein the optimal set of LiDAR correspondences for the active LiDAR frame comprises an optimal set selected from a set of LiDAR point-to-plane factors, whereby each LiDAR point-to-plane factor is computed based on a difference in value between a LiDAR measurement in the active LiDAR frame and a reference surface on an existing LiDAR map.
9. The module according to claim 8, wherein the difference in value between the LiDAR measurement in the active LiDAR frame and the reference surface on the existing LiDAR map is determined based on the LiDAR measurement, an average position of points on the existing LiDAR map that are closest to the LiDAR measurement, a normal vector that is perpendicular to the reference surface formed by the points on the existing LiDAR map, a point-to-plane rotation matrix associated with the LiDAR measurement, and a position of the body frame in the world frame as associated with the new LiDAR measurement.
10. The module according to claim 8, wherein the instructions that cause the processing unit to obtain the optimal set of LiDAR correspondences based on the measurements received from the LiDAR module and the initial guess of system states comprises instructions for directing the processing unit to:
 - compute mean values and variance values for each dimension of Jacobian matrices, wherein each Jacobian matrix is computed based on the LiDAR measurements received from the LiDAR module that have been linearized with regard to the initial guess of system states;
 - normalize the dimensions of the Jacobian matrices by subtracting a corresponding mean value from each dimension and dividing a result of the subtraction with a corresponding variance value;
 - group the normalized Jacobian matrices into sets of groups based on a high-dimensional voxelization with a predetermined voxel size; and

perform one or more rounds of stochastic-greedy selection to obtain the optimal set of LiDAR correspondences, the stochastic-greedy selection being performed based on the set of LiDAR point-to-plane factors, the Jacobian matrices, the sets of groups, and a minimum eigenvalue until a size of the optimal set of LiDAR correspondences is equal to a predetermined number of LiDAR correspondences.

11. The module according to claim 1, further comprising instructions for directing the processing unit to:

select a new keyframe from a keyframe database when it is determined that the set of optimized estimated states exceeds predetermined thresholds; and

generate the motion-corrected LiDAR point cloud map based on the set of optimized estimated states and the new keyframe.

12. A method for generating a set of optimized estimated states in real time using a computing module, the method comprising:

segmenting sequential data received from an inertial measurement unit (IMU) into a low-frequency part, a high-frequency part and a constant velocity part;

computing hybrid IMU constraints based on data contained within the low-frequency part, the high-frequency part and the constant velocity part;

computing an initial guess of system states for an active LiDAR frame by performing an IMU optimization process based on the hybrid IMU constraints;

obtaining an optimal set of LiDAR correspondences based on measurements of the active LiDAR frame and the initial guess of system states; and

computing the set of optimized estimated states for the active LiDAR frame by performing a maximum-a-priori (MAP) optimization process using the optimal set of LiDAR correspondences, the hybrid IMU constraints, and prior constraints.

13. The method according to claim 12, wherein the segmenting of the sequential data into the low-frequency part, the high-frequency part and the constant velocity part further comprises the steps of:

segmenting the sequential data into individual segments according to a B-Spline time interval Δt ;

integrating and normalizing acceleration data within each individual segment, whereby an individual segment having normalized integrated acceleration data equal to a zero value is classified as the constant velocity part;

fitting acceleration measurements contained within each unclassified individual segment to a linear model, and computing a root mean square error (RMSE) for each of the individual segments based on linear fitting residuals;

classifying an individual segment as the low-frequency part when it is determined that the RMSE computed for the individual segment is equal to or less than three times an accelerometer measurement error of the IMU; and

classifying an individual segment as the high-frequency part when it is determined that the RMSE computed for the individual segment is more than three times the accelerometer measurement error.

14. The method according to claim 13, wherein the hybrid IMU constraints comprise constant velocity part (CVP) factors for the constant velocity part, high frequency part (HFP) factors for the high-frequency part and low frequency part (LFP) factors for the low-frequency part.

15. The method according to claim 14, wherein the computing of the hybrid IMU constraints further comprises the steps of:

computing the CVP factors based on CVP parameters derived during a time interval of the CVP, the CVP parameters comprising roll and pitch correction factors, CVP rotation matrices defining an orientation of a body frame in a world frame, CVP acceleration parameters in the body frame, CVP IMU accelerometer biases, and a gravity vector in the world frame.

16. The method according to claim 15, wherein the computing of the hybrid IMU constraints further comprises the steps of:

computing the HFP factors based on HFP parameters derived during a time interval of the HFP, the HFP parameters comprising IMU pre-integration parameters, HFP rotation matrices defining an orientation of the body frame in the world frame, positions and velocities of the body frame in the world frame, the gravity vector in the world frame and the B-Spline time interval Δt .

17. The method according to claim 16, wherein the IMU pre-integration parameters comprise pre-integrated position parameters, pre-integrated velocity parameters and pre-integrated rotation parameters.
18. The method according to claim 16 or 17, wherein the computing of the hybrid IMU constraints further comprises the steps of:
 - computing the LFP factors based on LFP parameters derived during a time interval of the LFP, the LFP parameters comprising accelerations and angular velocities of the body frame in the world frame, LFP rotation matrices defining an orientation of the body frame in the world frame, LFP acceleration parameters in the body frame, LFP IMU accelerometer biases, the gravity vector in the world frame and gyroscope parameters in the body frame.
19. The method according to claim 12, wherein the optimal set of LiDAR correspondences for the active LiDAR frame comprises an optimal set selected from a set of LiDAR point-to-plane factors, whereby each LiDAR point-to-plane factor is computed based on a difference in value between a LiDAR measurement in the active LiDAR frame and a reference surface on an existing LiDAR map.
20. The method according to claim 19, wherein the difference in value between the LiDAR measurement in the active LiDAR frame and the reference surface on the existing LiDAR map is determined based on the LiDAR measurement, an average position of points on the existing LiDAR map that are closest to the LiDAR measurement, a normal vector that is perpendicular to the reference surface formed by the points on the existing LiDAR map, a point-to-plane rotation matrix associated with the LiDAR measurement, and a position of the body frame in the world frame as associated with the new LiDAR measurement.
21. The method according to claim 19, wherein the obtaining of the optimal set of LiDAR correspondences based on the measurements received from the LiDAR module and the initial guess of system states $\hat{\mathbf{X}}(t)$ comprises the steps of:
 - computing mean values and variance values for each dimension of Jacobian matrices, wherein each Jacobian matrix is computed based on the LiDAR measurements received from the LiDAR module that have been linearized with regard to the initial guess of system states;

normalizing the dimensions of the Jacobian matrices by subtracting a corresponding mean value from each dimension and dividing a result of the subtraction with a corresponding variance value;

grouping the normalized Jacobian matrices into sets of groups based on a high-dimensional voxelization with a predetermined voxel size; and

performing one or more rounds of stochastic-greedy selection to obtain the optimal set of LiDAR correspondences, the stochastic-greedy selection being performed based on the set of LiDAR point-to-plane factors r_t^L , the Jacobian matrices, the sets of groups, and a minimum eigenvalue until a size of the optimal set of LiDAR correspondences is equal to a predetermined number of LiDAR correspondences.

22. The method according to claim 12, further comprising the steps of:

selecting a new keyframe from a keyframe database when it is determined that the set of optimized estimated states exceeds predetermined thresholds; and

generating the motion-corrected LiDAR point cloud map based on the set of optimized estimated states and the new keyframe.