

Article

LiDAR-IMU Time Delay Calibration Based on Iterative Closest Point and Iterated Sigma Point Kalman Filter

Wanli Liu

School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China; lwl412101@cumt.edu.cn; Tel.: +86-516-8359-0706

Academic Editors: Cheng Wang, Julian Smit, Ayman F. Habib and Michael Ying Yang

Received: 3 January 2017; Accepted: 5 March 2017; Published: 8 March 2017

Abstract: The time delay calibration between Light Detection and Ranging (LiDAR) and Inertial Measurement Units (IMUs) is an essential prerequisite for its applications. However, the correspondences between LiDAR and IMU measurements are usually unknown, and thus cannot be computed directly for the time delay calibration. In order to solve the problem of LiDAR-IMU time delay calibration, this paper presents a fusion method based on iterative closest point (ICP) and iterated sigma point Kalman filter (ISP KF), which combines the advantages of ICP and ISP KF. The ICP algorithm can precisely determine the unknown transformation between LiDAR-IMU; and the ISP KF algorithm can optimally estimate the time delay calibration parameters. First of all, the coordinate transformation from the LiDAR frame to the IMU frame is realized. Second, the measurement model and time delay error model of LiDAR and IMU are established. Third, the methodology of the ICP and ISP KF procedure is presented for LiDAR-IMU time delay calibration. Experimental results are presented that validate the proposed method and demonstrate the time delay error can be accurately calibrated.

Keywords: LiDAR; inertial measurement unit; iterative closest point; iterated sigma point Kalman filter; time delay calibration

1. Introduction

In today's world, Light Detection and Ranging (LiDAR) devices and Inertial Measurement Units (IMUs) often found on vehicles, airplanes and robots are increasingly being used to perform localization or for navigation tasks. The LiDAR and IMU sensors, together, can supply accurate attitude estimation and are very suitable in many applications. However, in GPS-denied environments the IMUs are usually unreliable with respect to position for long periods of time due to time drift, which may cause large cumulative errors, the a LiDAR is a device which uses laser beams to determine the distance and azimuth from the sensor to an object, which can provide 3D localization information with high accuracy and efficiency to reduce or bound IMUs drift. Therefore, the integrated LiDAR-IMU system can provide highly accurate position and pose information over long periods of time in GPS-denied environments.

Usually combining information from multiple sensors offers several potential advantages, including enhanced accuracy and improved robustness, but these come at a cost: a fundamental requirement in any multiple sensor system is time delay calibration. To ensure optimal performance, the LiDAR and IMU sensors must be properly time delay calibrated, including estimates of the relative timing of each sensor measurement, coordinate transformation between the different sensors, and time delay calibration parameters estimation are required [1–4].

Several methods exist for LiDAR, IMU and camera time delay calibration. The work of Kelly et al. [5] presented a time delay iterative closest point method for determining the time delay

between inertial and visual sensor measurements. Aghili et al. [6] presented a robust 6-Degree of Freedom (DOF) relative navigation system by combining the iterative closest point (ICP) registration algorithm and a noise Adaptive Kalman Filter (AKF) in a closed loop configuration together with measurements from a LiDAR and an IMU. Pothou et al. [7] investigated the determination of the misalignment between the IMU body frame and the LiDAR frame, using a Quality Assurance/Quality Control (QA/QC) technique to evaluate the LiDAR-IMU bore sight misalignment. Deymier et al. [8] proposed the self-calibration of a vehicle's acquisition system with cameras, IMU and 3D LiDAR. Jian et al. [9] integrated two complementary technologies—Inertial Navigation System (INS) and LiDAR Simultaneous Localization and Mapping (SLAM)—into one navigation frame with a loosely coupled Extended Kalman Filter (EKF) to use the advantages and overcome the drawbacks of each system to establish a stable long-term navigation process. Mirzaei et al. [10] presented an EKF for precisely determining the unknown transformation between a camera and an IMU. Yun et al. [11] developed the IMU/Vision/LiDAR integrated navigation system which can provide accurate relative navigation information in GNSS-denied environments; and construct an overall integrated navigation filter based on the EKF approach. Li et al. proposed [12] a method of integrating the measurements from a LiDAR and a Micro-Electro-Mechanical System (MEMS) IMU, and using the Kalman Filter (KF) to estimate the error of IMU and LiDAR sensors.

As mentioned above, the AKF, EKF, KF and ICP algorithms are used in LiDAR, IMU and camera sensors to calibrate the time delay. During the course of LiDAR-IMU calibration, neither the EKF algorithm nor AKF algorithm can avoid time delay calibration bias and filtering divergences. Initially, the correspondences between LiDAR and IMU measurements are usually unknown; as a result the relative time delay information between the LiDAR-IMU data flows cannot be computed directly [13–16]. The question is posed differently as follows: first, the measurement rate information of LiDAR and IMU cannot be directly compared. Second, in general, the rigid body transformation between the LiDAR and the IMU is unknown. Third, both signals are discrete and have different temporal resolutions, the LiDAR typically scanning information is available at 5–20 Hz or less, however the IMU produces data is normally available at 100 Hz or more [17–20].

In order to solve the problem of LiDAR-IMU time delay calibration, we present a fusion method based on ICP and an iterated sigma point Kalman filter (ISPKF), which combines the advantages of ICP and ISPKF [21–23]. The total least squares cost function is used by the ICP algorithm for registration, which allows us to merge the IMU orientation measurement uncertainty in a principled way, and to reduce the longer time intervals due to the accumulated noise effects by integrated LiDAR orientation measurement. The ISPKF algorithm measurement update step is iterated until the change in the posterior state estimate drops below some small threshold, and can optimally estimate the time delay calibration parameters. First of all, the coordinate transformation from the LiDAR frame to the IMU frame is realized. Second, the measurement model and time delay error model of LiDAR and IMU sensors are established. Third, the methodology of the ICP and ISPKF procedure is presented for LiDAR-IMU time delay calibration. Experimental results are presented that validate the proposed method and demonstrate the time delay can be accurately calibrated.

This paper is organized as follows: the coordinate transformation between LiDAR and IMU are described in Section 2. In Section 3, we establish the LiDAR and IMU measurement models and the time delay error model in LiDAR-IMU. Section 4 given the implementation procedure of ICP-ISPKF for LiDAR-IMU time delay calibration. Section 5 experimental studies and analysis the results using LiDAR, IMU and optical measuring system based our proposed method are described, followed by the conclusions in Section 6.

2. Coordinate Transformation between LiDAR and IMU

2.1. Coordinate Frame

In general, there are basically four coordinate frames in the LiDAR and IMU, The relationship between these frames is shown in Figure 1:

- (1) LiDAR frame, $\{L\}$, is represented in this frame of reference, in which the axes are defined as right, forward and up.
- (2) IMU frame, $\{I\}$, is defined by the IMU, in which angular rotation rates and linear accelerations are measured, with its origin at a point on the IMU body.
- (3) Object frame, $\{O\}$, is the coordinate of moving object, the axes in the object frame are forward, right and down.
- (4) World frame, $\{W\}$, is considered to be the fundamental coordinate frame and serves as an absolute reference for both the $\{I\}$ and the $\{L\}$.

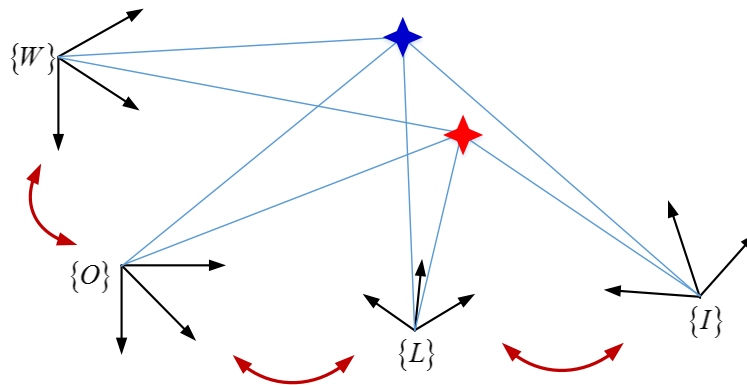


Figure 1. Relationship between the $\{L\}$, $\{I\}$, $\{O\}$ and $\{W\}$.

2.2. Transformation from LiDAR Frame to IMU Frame

2.2.1. Transformation from LiDAR Frame to the World Frame

Suppose a point ${}^W P$ in the $\{W\}$ frame is located at ${}^L P$ in the $\{L\}$, the transformation from the $\{W\}$ coordinate to the $\{L\}$ coordinate system can be expressed as [24]:

$${}^L P = R_W^L {}^W P + {}^L T_{LW} \quad (1)$$

where R_W^L is a 3×3 orthonormal matrix representing the rotation from the $\{W\}$ frame to the $\{L\}$ frame, and ${}^L T_{LW}$ is the translation vector. The subscript is the origin of the $\{L\}$ coordinate, and is the origin of the tangent frame. Our goal is to calculate R_W^L and ${}^L T_{LW}$.

The transformation is implemented through the observation of LiDAR scanning to a calibration plane. A geometric constraint can be obtained between LiDAR scanning points and the calibration plane. Since LiDAR scanning points lie on the calibration plane, and ${}^W r$ is the normal vector to the plane, we have:

$${}^W r \cdot {}^W P = d \quad (2)$$

where ${}^W P$ is the coordinate of LiDAR scanning point in $\{W\}$ frame; d is a scalar representing the distance from the origin of the $\{W\}$ frame to the calibration plane, which is calculated from the position and orientation of the calibration plane. From Equation (1) we know that:

$${}^W P = \left(R_W^L\right)^{-1} \left({}^L P - {}^L T_{LW}\right) \quad (3)$$

By substituting Equation (3) into Equation (2), we have:

$${}^W r \cdot \left(R_W^L \right)^{-1} \left({}^L P - {}^L T_{LW} \right) = d \quad (4)$$

For any given LiDAR scanning point and calibration plane position, Equation (4) gives a constraint on R_W^L and ${}^L T_{LW}$. It will be solved in two consecutive steps: a linear solution, followed by a non-linear optimization [25,26]:

(1) Linear solution. The LiDAR scanning plane in the $\{L\}$ is ${}^L Z = 0$. Each LiDAR scanning point can be represented as ${}^L P = [{}^L P_x \ {}^L P_y \ {}^L P_z]^T$. Then Equation (4) is rewritten as:

$${}^W r \cdot \left(R_W^L \right)^{-1} \left(\begin{bmatrix} {}^L P_x \\ {}^L P_y \\ {}^L P_z \end{bmatrix} - {}^L T_{LW} \right) = {}^W r \cdot \left(R_W^L \right)^{-1} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - {}^L T_{LW} \right) \begin{bmatrix} {}^L P_x \\ {}^L P_y \\ {}^L P_z \end{bmatrix} = d \quad (5)$$

Let's define $Z = \left(R_W^L \right)^{-1} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - {}^L T_{LW} \right)$, Equation (5) is rewritten as:

$${}^W r \cdot Z \cdot {}^L P = d \quad (6)$$

where Z is the parameter to be solved, which is the integration of two unknown parameters R_W^L and ${}^L T_{LW}$.

Equation (6) can be solved using the standard linear least squares algorithm. In order to obtain a better result, multiple calibration planes should be used in the transformation. Suppose in the transformation we use a total of N calibration planes, with M_i ($i = 1, \dots, N$) LiDAR scanning points

on the i -th plane. Let $Z = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{bmatrix}$, the normal vector for the i -th plane ${}^W r_i = [r_{i,1} \ r_{i,2} \ r_{i,3}]$,

the distance from origin of the $\{W\}$ frame and the i -th calibration plane is d_i and the j -th LiDAR scanning point on the i -th calibration plane is ${}^L P_{ij} = [{}^L P_{ij,x} \ {}^L P_{ij,y} \ {}^L P_{ij,z}]^T$, Equation (6) is rewritten as:

$$(r_{i,1}z_{11} + r_{i,2}z_{21} + r_{i,3}z_{31}) \cdot {}^L P_{ij,x} + (r_{i,1}z_{12} + r_{i,2}z_{22} + r_{i,3}z_{32}) \cdot {}^L P_{ij,y} + (r_{i,1}z_{13} + r_{i,2}z_{23} + r_{i,3}z_{33}) \cdot {}^L P_{ij,z} = d_i \quad (7)$$

where $i = 1, 2, \dots, N, j = 1, 2, \dots, M$.

Therefore for each LiDAR scanning point we have a linear equation which is a row in Equation (7) can be calculated using the standard linear least square algorithm. Then R_W^L and ${}^L T_{LW}$ will be obtained from Z .

(2) Nonlinear solution. The linear solution is obtained by the least squares method, which aims to minimize an algebraic distance. A nonlinear minimization method is used to minimize the differences between the measured Euclidean distances as well as the calculated distance from the LiDAR scanning points to the calibration plane, which is physically meaningful.

Equation (4) gives two types of distances: d is the distance from the calibration plane to the origin of the $\{W\}$ frame obtained by a field survey, and ${}^W r \cdot \left(R_W^L \right)^{-1} \left({}^L P - {}^L T_{LW} \right)$ is the calculated distance. The difference between these two distances is defined as the distance error for one calibration plane pose. The nonlinear solution aims to minimize the sum function of the distance errors for all the plane positions. The sum function is defined as:

$$sum = \sum_{i=1}^N \sum_{j=1}^{M_i} \left({}^W r_i \cdot \left(R_W^L \right)^{-1} \left({}^L P_{ij} - {}^L T_{LW} \right) - d_i \right)^2 \quad (8)$$

where r_i defines the i -th calibration plane, d_i is the distance from the i -th plane to the center of the $\{W\}$ frame, and P_{ij} is the j -th LiDAR scanning point with the i -th calibration plane; the pair of R_W^L and ${}^L T_{LW}$ that minimize Equation (8) are considered to be the rotation and translation matrix to be calculated. Equation (8) can be minimized as a nonlinear optimization problem by getting the translation and rotation matrix between the $\{W\}$ frame and LiDAR scanning points in the $\{W\}$ frame can be converted to a point in the $\{L\}$ frame.

2.2.2. Transformation from LiDAR Frame to IMU Frame

The transformation of the LiDAR frame to the IMU one is to obtain the geometric relationship between the $\{L\}$ frame and $\{O\}$ frame. The IMU consists of three gyros and three accelerometers. The gyros provide change of Euler angles, while the accelerometers give the specific force. By integrating the output of the gyros and the accelerometers, we can obtain the translation and rotation matrix between the $\{W\}$ frame and the $\{O\}$ frame. Let the rotation matrix be R_O^W , and translation vector be ${}^O T_{WO}$, then a point in the $\{O\}$ frame can be converted to a point in the $\{W\}$ frame by:

$${}^W P = R_O^W P + {}^O T_{WO} \quad (9)$$

where ${}^W P$ is the point in the $\{W\}$ frame, and ${}^O P$ is the point in the $\{O\}$ frame.

Finally, by substituting Equation (1) into Equation (9) we have:

$${}^L P = R_W^L \left(R_O^W P + {}^O T_{WO} \right) + {}^L T_{LW} \quad (10)$$

Equation (10) is the transformation from the $\{O\}$ frame to the $\{L\}$ frame.

3. LiDAR and IMU Measurement Model

3.1. LiDAR Measurement Model

The LiDAR scans the laser beam through 360° . Therefore, a post which will be modeled as a vertical line in the tangent frame will appear as a point in the LiDAR frame. Similarly, a wall which will be modeled as a vertical plane in the tangent frame will appear as a line in the LiDAR frame of references. The LiDAR position and point feature detection are shown in Figure 2.

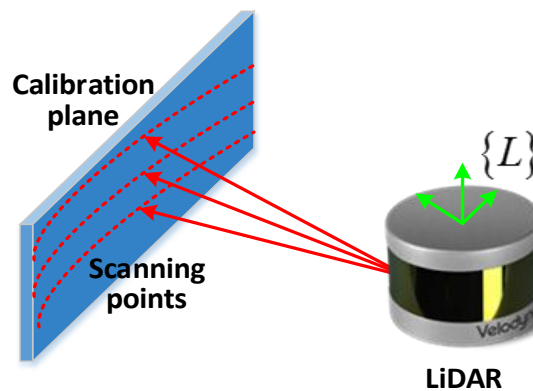


Figure 2. The LiDAR position and point feature detection.

Consider a mapped vertical post at P_0 . The post will appear as a point in the LiDAR return. The vector from T to P_0 in world frame is ${}^W T_{LP_0} = [N_0 \ E_0 \ 0]^T$, which is known from the survey. The line is denoted as:

$${}^W T_{WP}(m) = {}^W T_{WP_0} + m e_3 \quad (11)$$

where ${}^W T_{WP}(m)$ is the vector in world frame from T to a point P at height m on the post, m is a scalar with $m \in (0, +\infty)$. The vector e_3 is parallel to the post. In our application, all the posts and planes are modeled as vertical. For a vertical post $e_3 = [0 \ 0 \ 1]^T$ in the world frame.

For any feature point P , the vector from the LiDAR to the point is:

$$T_{LP} = T_{WP} - (T_{WO} + T_{OL}) \quad (12)$$

As is shown in Figure 3. Equation (12) is valid for all the reference frames.

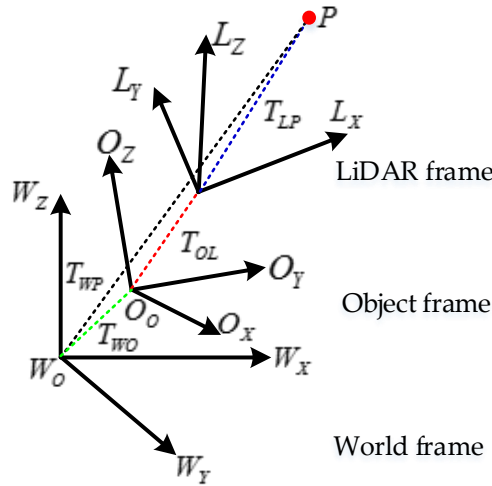


Figure 3. The LiDAR, Object and World frames as well as the feature point P .

In Equation (12) and Figure 3, T_{WP} is a constant in the world frame known from the feature survey. T_{WO} will be calculated in the world frame. T_{OL} in the object frame is known, and can be determined by the pre-calibration.

R_O^L is the rotation matrix from the object frame to the LiDAR frame, so the vector from the LiDAR to a feature point in LiDAR coordinates is denoted as:

$${}^L T_{LP} = R_O^L \left(R_W^O ({}^W T_{WP} - {}^W T_{WO}) - {}^O T_{OL} \right) \quad (13)$$

By substituting Equation (11) into Equation (13) we have:

$${}^L T_{LP}(m) = R_O^L \left(R_W^O ({}^W T_{WP} + m e_3 - {}^W T_{WO}) - {}^O T_{OL} \right) \quad (14)$$

which is the vector in the LiDAR coordinates from the origin of the LiDAR sensor to a point with height m on the post. By expanding Equation (14), the coordinate of ${}^L T_{LP}(m)$ in the $-{}^L Z$ direction is:

$$Z(m) = e_3^W \cdot {}^L T_{LP}(m) = m \cdot e_3^W \cdot R_O^L \cdot R_W^O \cdot e_3 + e_3^W \cdot R_O^L \left(R_W^O ({}^W T_{WP} - {}^W T_{WO}) - {}^O T_{OL} \right) \quad (15)$$

Equation (15) is the theoretical equation. Note that for a single-planar LiDAR, the scan plane in LiDAR coordinate is ${}^L Z = 0$. Therefore, for all the LiDAR points we have ${}^L Z = 0$, and we can use this fact to calculate m as:

$$m = - \frac{e_3^W \cdot R_O^L \left(R_W^O ({}^W T_{WP} - {}^W T_{WO}) - {}^O T_{OL} \right)}{e_3^W \cdot R_O^L \cdot R_W^O \cdot e_3} \quad (16)$$

By substituting Equation (16) into Equation (14), the detected point is:

$${}^L T_{LP}(m) = R_O^L \left(R_W^O ({}^W T_{WP} - \frac{e_3^W \cdot R_O^L \left(R_W^O ({}^W T_{WP} - {}^W T_{WO}) - {}^O T_{OL} \right)}{e_3^W \cdot R_O^L \cdot R_W^O \cdot e_3} e_3 - {}^W T_{WO}) - {}^O T_{OL} \right) \quad (17)$$

Equation (17) is the LiDAR measurement model.

3.2. IMU Measurement Model

In general, the IMU measures the moving object's rotational motion in three orthogonal axes. Usually, the output rates of IMU measuring angles are not zero, even when the IMU is stationary, and during the IMU measurements the angular products are easily be corrupted by noise. At time t , the IMU measuring angular velocity can be written as [5,27]:

$$\omega_m(t) = \omega^I(t) + b_g + n_g(t) \quad (18)$$

where $n_g(t)$ is the noise vector, b_g and $\omega^I(t)$ are the bias vector and the true angular velocity of the IMU, respectively.

In fact, all IMUs are subject to time drift or their measurement bias values slow change with time. Usually, in the very short period of time of the calibration interval for the IMU, the measurement bias and drift may be negligible, however, in order to obtain an accurate estimate of the gyroscope biases, at the start of calibration, we average several seconds of IMU data.

At a time $t > t_0$, in order to obtain the IMU orientation estimate, according to the modified Rodrigues parameters kinematic differential equation, we integrate each IMU measurement forward as:

$$\dot{\rho}(t) = \frac{1}{4} \left((1 - \|\rho(t)\|^2) I_3 + 2[\rho(t)]_{\times} + 2\rho(t) \right) \omega^I(t) \quad (19)$$

where $\omega^I(t) = \omega_m(t) - b_g - n_g(t)$.

As integration continues, the noise and drift will cause inaccuracies, therefore, relative to its initial orientation, the IMU true orientation will become less certain. By estimating the IMU propagation orientation covariance, the matrices F and G can be defined as:

$$\begin{cases} F = \frac{\partial \dot{\rho}}{\partial \omega} = \frac{1}{2} \left((\rho^T \cdot \omega) I_3 - [\omega]_{\times} + \rho \omega^T - \omega \cdot \rho^T \right) \\ G = \frac{\partial \dot{\rho}}{\partial n_g} = -\frac{1}{4} \left((1 - \|\rho\|^2) I_3 + 2[\rho]_{\times} + 2\rho \rho^T \right) \end{cases} \quad (20)$$

Using the continuous time equation the propagation orientation covariance can be written as:

$$\dot{P}_I = FP_I + P_I F^T + GQ_g G^T \quad (21)$$

According to Equation (21), any feature point in the IMU measurement curve can be selected as a zero uncertainty reference. We choose one of the earliest IMU measurements as the uncertainty computation reference, mainly because our computed deviation estimate is relative to the earliest measurement.

3.3. Time Delay Error Model

We will identify a specific instantaneous local LiDAR frame as $\{L_k\}$ with timestamp t_{L_k} according the receiver clock, for $k = 1, \dots, n$ LiDAR poses. Similarly, for $j = 1, \dots, m$ IMU poses, the $\{I_j\}$ will be identified as a specific instantaneous local IMU frame with timestamp t_{I_j} . In general, the IMU data are available at a substantially higher rate than the LiDAR data, and $m > n$.

To compare LiDAR and IMU measurement data, a common representation is required; we use the orientation measurements to match IMU and LiDAR. The IMU orientation is measured with respect to the initial sensor pose. The LiDAR orientation is measured relative to a series of calibration planes. Our algorithm employs the modified Rodrigues parameters as a minimal and convenient orientation representation. The use of the modified Rodrigues parameters vector representation allows us to propagate the IMU orientation uncertainty forward continuously in time [28].

We have the following relationship at any time t_{I_j} :

$${}^W_I \rho(t) = {}^W_L \rho(t + \tau) \cdot {}^L_I \rho \quad (22)$$

where ${}^L_I \rho$ is the modified Rodrigues parameter vector with the orientation from the $\{I\}$ frame to the $\{L\}$ frame. ${}^W_L \rho(t + \tau)$ is the modified Rodrigues parameter vector with the orientation from the $\{L\}$ frame to the $\{W\}$, τ is the time delay. ${}^W_L \rho(t)$ is the modified Rodrigues parameter vector with the orientation from the $\{I\}$ frame to the $\{W\}$ frame.

Although the orientation of the rate measurement information supplied by the IMU cannot be measured directly, by integrating the IMU gyroscope data, we can obtain the IMU measurement orientation change over a period of time. Similarly, we can conveniently compute the LiDAR measured orientation relative to the $\{W\}$ frame. The measurement relationship between the $\{I\}$, $\{L\}$ and $\{W\}$ frame can be computed as:

$${}^{I_0}_I \rho(t) = {}^{I_0}_W \rho \cdot {}^W_L \rho(t + \tau) \cdot {}^L_I \rho \quad (23)$$

where ${}^{I_0}_W \rho$ is the modified Rodrigues parameter vector with the orientation from the $\{I_0\}$ frame to the $\{W\}$ frame.

(1) Position error analysis: the error in position can be calculated as follows:

$$\delta({}^W T_{WO}) = R_O^W \cdot {}^O V - \hat{R}_O^W \cdot {}^O \hat{V} = \hat{R}_O^W \cdot \delta({}^O V) - {}^W \hat{V} \cdot \rho \quad (24)$$

where $R_O^W = \hat{R}_O^W(I + \rho)$, $I = (R_W^L)^T \cdot R_W^L$, ρ is the calibration plane tilt error, and ${}^O V = {}^O \hat{V} + \delta({}^O V)$.

(2) Velocity error analysis: the error in velocity is modeled as follows:

$$\begin{aligned} \delta({}^O \dot{V}) &= {}^O \dot{V} - {}^O \hat{V} = {}^O a_O - (\Omega_{IL}^O + \Omega_{IO}^O) \cdot {}^O V - {}^O \hat{a}_O + (\hat{\Omega}_{IL}^O + \hat{\Omega}_{IO}^O) \cdot {}^O \hat{V} \\ &= \hat{R}_O^W \left(\left(\frac{\partial(g^W)}{\partial({}^W T_{WO})} + {}^W \hat{V} \cdot \frac{\partial({}^W \omega_{IL})}{\partial({}^W T_{WO})} \right) \delta({}^W T_{WO}) + (g^W + {}^W \omega_{IL} ({}^W \hat{V})^T - ({}^W \omega_{IL})^T \cdot {}^W \hat{V} \cdot I) \rho \right) \\ &\quad - (\hat{\Omega}_{IL}^O + \hat{\Omega}_{IO}^O) \delta({}^O V) - \delta b_a - {}^O \hat{V} \cdot \delta b_g - \gamma_a - {}^O \hat{V} \cdot \gamma_g \end{aligned} \quad (25)$$

where g^W is the local gravity vector at the moving object location represented in the $\{W\}$ frame, ${}^W \omega_{IL}$ is the angular rate of the $\{W\}$ frame origin to the $\{I\}$ frame represented in the $\{L\}$ frame. γ_g is the white Gaussian measurement noise, b_g is the gyro bias, which is modeled as a random constant plus random noise.

(3) Attitude error analysis: the attitude error is given by:

$$\delta(\hat{\Theta}) = \hat{\Theta} - \dot{\Theta} = \Omega_E^{-1} \cdot {}^W \hat{\omega}_{IW} - \delta({}^W \omega_{IW}) - \hat{R}_I^W (\delta b_g + \gamma_g) \quad (26)$$

where $\delta({}^W \omega_{IW})$ can be calculated as:

$$\delta({}^W \omega_{IW}) = -\omega_{IL} \cdot \begin{bmatrix} \sin \alpha \\ 0 \\ \cos \alpha \end{bmatrix} \cdot \frac{\partial \Theta}{\partial ({}^W R_{IW})} \cdot \delta({}^W R_{IW}) \quad (27)$$

(4) Time delay calibration parameters error model: the measurement noise vector, the time delay parameter vector and process noise vector of LiDAR-IMU can be expressed as:

$$\begin{cases} x = \begin{bmatrix} \delta({}^W T_{WO}) & \delta({}^O V) & \delta \Theta & \delta \omega_{ba} & \delta \omega_{bg} \end{bmatrix}^T \\ b = \begin{bmatrix} b_a & b_g \end{bmatrix}^T \\ \gamma = \begin{bmatrix} \gamma_a & \gamma_g \end{bmatrix}^T \end{cases} \quad (28)$$

where the x , b and γ are the time delay of the error state vector, process noise vector and measurement noise vector, respectively.

According to the Equations (25)–(28), the time delay calibration parameters error model of LiDAR-IMU can be computed as:

$$\begin{bmatrix} \delta({}^W T_{WO}) \\ \delta({}^O \dot{V}) \\ \delta\Theta \\ \delta\omega_{ba} \\ \delta\omega_{bg} \end{bmatrix} = \begin{bmatrix} \hat{R}_O^W \frac{\partial(g^W)}{\partial({}^W T_{WO})} & \hat{R}_O^W & -{}^W \hat{V} & 0 & 0 \\ 0 & -(\hat{\Omega}_{IL}^O + \hat{\Omega}_{IO}^O) & -\hat{R}_O^W g^W & -I & -{}^O \hat{V} \\ 0 & 0 & \hat{\Omega}_E^{-1} & 0 & -\hat{R}_O^W \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta({}^W T_{WO}) \\ \delta({}^O V) \\ \delta\Theta \\ \delta\omega_{ba} \\ \delta\omega_{bg} \end{bmatrix} + \begin{bmatrix} -I & 0 & 0 & 0 \\ 0 & -{}^O \hat{V} & 0 & 0 \\ 0 & -\hat{R}_O^W & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} \gamma_a \\ \gamma_g \\ b_a \\ b_g \end{bmatrix} \quad (29)$$

4. Time Delay Calibration Using the ICP-ISP KF Integration Method

The ICP algorithm can be used to register the spatial data from IMU and LiDAR measurements, which operates by aligning two curves in a 3-D orientation space generated from integrated IMU gyroscope data and from LiDAR scanner data. Each point on the respective curve has a corresponding timestamp, identifying the time at which the measurement arrived at the receiver. By registering the orientation curves, we are able to use the timestamp values to estimate the relative delay between IMU and LiDAR data streams. ICP incorporates in a principled way the uncertainty in the LiDAR orientation measurements and accounts for the fact that the integrated IMU orientation becomes less accurate over longer intervals due to the incorporation of noise. We specifically avoid this using the ISP KF because of The ISP KF algorithm measurement update step is iterated until the change in the posterior state estimate drops below some small threshold, and can optimally estimate the time delay calibration parameters. The ISP KF is shown to achieve superior performance in terms of efficiency and accuracy compared with the KF, EKF and UKF, also the Gabe Sibley et al. [29] have compared the ISP KF with KF, EKF, UKF and Gauss-Newton filter, and demonstrate the ISP KF's capabilities in avoiding IMU measuring noise. Therefore, in order to solve the problem of LiDAR-IMU time delay calibration, we use the fusion method based on ICP and ISP KF algorithm.

As shown in Figure 4, the LiDAR-IMU time delay calibration is integrated with ICP and ISP KF method. Firstly, once the ICP succeeds matching the LiDAR measurement points, the LiDAR-IMU state estimations are updated with the precise registration value. Secondly, at the time delay filter estimate step, the obtained predicted pose is required to find the corresponding points between LiDAR and IMU measurements. Thirdly, at each time step, the recursive algorithm is used to update the related covariance estimates and precise registration value for LiDAR-IMU. The ICP convergence information and the fault-detection logic are used to adaptively adjust ISP KF reliable estimate of motion state and a set of related parameters. In closed-loop ICP-ISP KF architecture, through ICP initial guess and fault detection, the LiDAR-IMU robust pose tracking and automatic fault recovery are established. Finally, the pose prediction information used to align the time delay error for ICP [6,30].

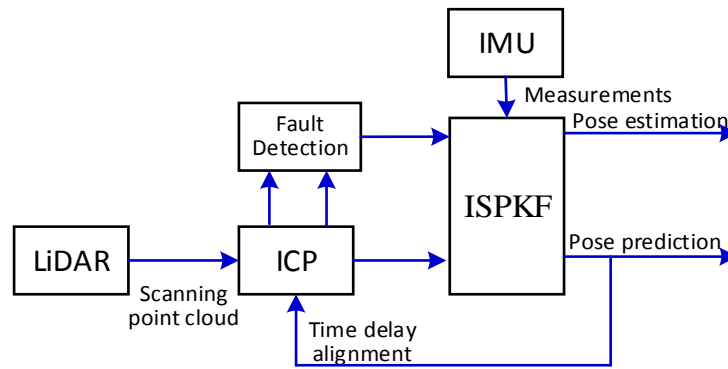


Figure 4. The Architecture of ICP and ISPKF Integration method.

4.1. The ICP Algorithm for Estimation the Time Delay and Relative Orientation of LiDAR-IMU

The ICP algorithm is utilized to estimation the LiDAR-IMU time delay and relative orientation. At the beginning, the transforms between the LiDAR-IMU orientation curves are computed through iteratively selecting n correspondences point using ICP algorithm. We will adjust the search the corresponding time scale with the orientation curves converge. The ICP algorithm can be describe in two steps as follows [5,31]:

Step I: Registration Rules.

The ICP algorithm operates by iteratively selecting the closest point between the IMU orientation curve and the LiDAR measurement point, and the concept of ICP proximity requires a suitable distance measurement.

For the LiDAR measurement point $P_{L_i}^W$, the distance to the nearby IMU point $P_{I_j}^W$ is determined by applying the current spatial transformation model and calculating the incremental modified Rodrigues parameter vector. Taking from $P_{L_i}^W$ to the orientation $P_{I_j}^W$, we can get the following equation:

$$\sigma_{ij} = -(\hat{\rho}_W^{I_0} \cdot \rho_{L_i}^W \cdot \hat{\rho}_I^L) \cdot \rho_{I_j}^{I_0} \quad (30)$$

According to the Equation (30), it can be known that the closest point is the IMU measurement point, and it produces the minimum value of distance function that can be computed as:

$$d_{ij} = 4\arctan\sqrt{\sigma_{ij}^T \sigma_{ij}} \quad (31)$$

where d_{ij} is the incremental rotation arc length taking from $\rho_{L_i}^W$ to $\rho_{I_j}^{I_0}$ in radians orientation.

The search ranges of each LiDAR scanning point in the IMU measurement curve are constrained by the maximum expected time delay, and can be definitely expressed as $[t_{L_i} - \hat{t} - \delta t, t_{L_i} - \hat{t} + \delta t]$, where $\pm\delta t$ is the maximum value. When the algorithm converges, the δt can be reduced on each iteration.

Step II: Nonlinear Iterative Registration.

We can get n correspondence relationships between the IMU and LiDAR orientation measurement curves by selecting the closest IMU point for each LiDAR point. The generalized nonlinear total least squares method is used to align the IMU and LiDAR orientation curves, where we minimize the cost function as follows:

$$U\left(\begin{matrix} I_0 \\ W \end{matrix} \rho, \begin{matrix} L \\ I \end{matrix} \rho\right) = \sum_{k=1}^n s_k P_{L_k}^{-1} s_k^T + \sum_{k=1}^n t_k P_{I_{f(k)}}^{-1} t_k^T \quad (32)$$

where P_{L_k} and $P_{I_{f(k)}}$ are the associated covariance matrices, s and t are the stacked residuals vectors, the $\begin{matrix} I_0 \\ W \end{matrix} \rho$ and $\begin{matrix} L \\ I \end{matrix} \rho$ are transform parameters can be computed as:

$$\begin{cases} s_k = \frac{W}{L_k} \rho - \frac{W}{L_k} \hat{\rho} \\ t_k = \frac{I_0}{I_{f(k)}} \rho - \frac{I_0}{I_{f(k)}} \hat{\rho} \end{cases} \quad (33)$$

We remark the $\frac{W}{L_i} \rho$ and $\frac{I_0}{I_{f(k)}} \rho$ are actually estimated quantities, but that we treat them as observations here. Considering the uncertainty in LiDAR and IMU measurements, we take that the residuals subjecting to n model constraints can be expressed as:

$$\frac{I_0}{I_{f(k)}} \hat{\rho} = \frac{W}{I_0} \hat{\rho} \cdot \frac{W}{L_i} \hat{\rho} \cdot \frac{L}{I} \hat{\rho} \quad (34)$$

By using Lagrange multipliers and incorporating constraints, differentiating and rearranging, we subjected the Equation (33) to the n constraints and derived from Equation (34), the following equation can be obtained:

$$\begin{bmatrix} \Delta_{\frac{W}{L_i} \rho}^{I_0} \\ \Delta_{\frac{L}{I} \rho} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^n J_k M_k^{-1} J_k^T \end{bmatrix}^{-1} \begin{bmatrix} \sum_{k=1}^n \Delta y_k M_k^{-1} J_k^T \end{bmatrix} \quad (35)$$

where $\Delta_{\frac{W}{L_i} \rho}^{I_0}$ and $\Delta_{\frac{L}{I} \rho}$ are the incremental updates to $\frac{W}{L_i} \hat{\rho}$ and $\frac{L}{I} \hat{\rho}$, respectively, for the present iteration, and Δy is a stacked vector, $\Delta y_i = \frac{I_0}{I_{f(k)}} \rho - (\frac{I_0}{I_{f(k)}} \hat{\rho} \cdot \frac{W}{L_k} \rho \cdot \frac{L}{I} \hat{\rho})$. The matrix M_k is the combined observation covariance matrix, $M = P_{I_{f(k)}} + H_k P_k H_k^T$, where H_k represents the Jacobian matrix with respect to observation $\frac{W}{L_k} \rho$:

$$H(\frac{I_0}{W} \rho, \frac{W}{L_k} \rho, \frac{L}{I} \rho) = \begin{bmatrix} \frac{\partial(\frac{I_0}{W} \rho \cdot \frac{W}{L_k} \rho \cdot \frac{L}{I} \rho)}{\partial(\frac{I_0}{W} \rho)}, \frac{\partial(\frac{I_0}{W} \rho \cdot \frac{W}{L_k} \rho \cdot \frac{L}{I} \rho)}{\partial(\frac{L}{I} \rho)} \end{bmatrix} \quad (36)$$

According to Equation (36), the ICP algorithm requires an initial estimate between the LiDAR and IMU data and we will use simple manual measurements to estimate the relative orientation between the LiDAR and IMU.

4.2. Iterated Sigma Point Kalman Filter (ISPKF) Algorithm for Compensation Calibration Parameters

After the relative orientation and optimal time delay estimation for the LiDAR-IMU, the ISPKF algorithm is used for compensation of the time delay calibration parameters between the LiDAR and IMU measurements.

The ISPKF algorithm uses the process noise component to implement the incremental state vector and the state covariance matrix, as expressed in:

$$\hat{X}_a(t_k) = \begin{bmatrix} \hat{X}(t_k) \\ n(t_k) \end{bmatrix} \quad (37)$$

where $n(t_k)$ is the noise vector in measurement process, $\hat{X}_a(t_k)$ is the augmented state vector. At time t_{k-1} , shortly after the LiDAR and IMU measurement update, the enhanced state covariance matrix $P_a^+(t_{k-1})$ and the enhanced state mean $\hat{X}_a^+(t_{k-1})$ can be formed as:

$$P_a^+(t_{k-1}) = \begin{bmatrix} P^+(t_{k-1}) & 0_{m \times 12} \\ 0_{12 \times m} & Q_c(t_{k-1}) \end{bmatrix} \quad (38)$$

$$\hat{X}_a^+(t_{k-1}) = \begin{bmatrix} \hat{X}^+(t_{k-1}) \\ 0_{12 \times 1} \end{bmatrix} \quad (39)$$

where the state vector $\hat{X}_a^+(t_{k-1})$ has size $m = 26 + 3n$ and $Q_c(t_{k-1})$ is the continuous-time covariance matrix for the noise vector $n(t_{k-1})$, when n calibration points are included, for the Cartesian and

inverse calibration points parameterization, respectively. The scaled form is employed for unscented transform, which requires a scaling term:

$$\lambda = \alpha^2(N + \beta) - N \quad (40)$$

The α parameter is used to control the sigma points spread with respect to the mean of the state; we usually set a small positive value for parameter α . In the a posteriori state distribution Taylor series expansion, the β parameter is used to correct the higher order term. We set $\beta = 2$ and minimize the joint Gaussian distribution fourth-order error.

We use the augmented state vector $\hat{X}_a^+(t_{k-1})$ to generate a set of sigma points according to the following equation:

$$\begin{cases} \chi_a^{(0)}(t_{k-1}) = \hat{X}_a^+(t_{k-1}) \\ \chi_a^{(l)}(t_{k-1}) = \hat{X}_a^+(t_{k-1}) + (S(t_{k-1}))_j, j = l = 1, \dots, N \\ \chi_a^{(l)}(t_{k-1}) = \hat{X}_a^+(t_{k-1}) - (S(t_{k-1}))_j, j = 1, \dots, N, l = N + 1, \dots, 2N \\ S(t_{k-1}) = \sqrt{(\lambda + N)P_a^+(t_{k-1})} \end{cases} \quad (41)$$

where $P_a^+(t_{k-1})$ is the augmented state covariance matrix, $(S(\cdot))_j$ expresses the j -th column of the matrix S . The weight values of the associated sigma point can be described as:

$$\begin{cases} W_m^{(0)} = \lambda / (\lambda + N) \\ W_c^{(0)} = \lambda / (\lambda + N) + (1 - \alpha^2 + \beta) \\ W_m^{(j)} = W_m^{(j)} = \frac{1}{2(\lambda + N)}, j = 1, \dots, 2N \end{cases} \quad (42)$$

A single σ -point can be propagated by the enhanced nonlinear process model function f_a . Over the time interval $t \in [t_{k-1}, t_k]$, the a priori state estimate and covariance t_k can be computed as:

$$\begin{cases} \chi_k^{(p)} = f_a(\chi_a^{(p)}(t_{k-1})), p = 0, \dots, 2N \\ \hat{X}_k^- = \sum_{p=0}^{2N} W_m^{(p)} \chi_k^{(p)} \\ P_k^- = \sum_{p=0}^{2N} W_c^{(p)} (\chi_k^{(p)} - \hat{X}_k^-) (\chi_k^{(p)} - \hat{X}_k^-)^T \end{cases} \quad (43)$$

Through propagating each σ point through the nonlinear measurement model function h , we can determine the predicted measurement vector as follows:

$$\begin{cases} \mathbb{Z}_k^{(p)} = h(\chi_k^{(p)}), p = 0, \dots, 2N \\ \hat{Z}_k = \sum_{p=0}^{2N} W_m^{(p)} \mathbb{Z}_k^{(p)} \end{cases} \quad (44)$$

Through computing the a posteriori state vector, the state covariance matrix, and the Kalman gain matrix, we can perform the state update as follows:

$$\begin{cases} P_{\hat{Z}_k \hat{Z}_k} = \sum_{p=0}^{2N} W_c^{(p)} (\mathbb{Z}_k^{(p)} - \hat{Z}_k) (\mathbb{Z}_k^{(p)} - \hat{Z}_k)^T \\ K_k = P_{\hat{X}_k \hat{Z}_k} (P_{\hat{Z}_k \hat{Z}_k} + R_k)^{-1} \\ \hat{X}_k^+ = \hat{X}_k^- + K_k (\mathbb{Z}_k - \hat{Z}_k) \\ P_k^+ = P_k^- - K_k P_{\hat{Z}_k \hat{Z}_k} K_k^T \end{cases} \quad (45)$$

where R_k is the measurement covariance matrix for Z_k , while $P_{\hat{Z}_k \hat{Z}_k}$ and $P_{\hat{X}_k \hat{Z}_k}$ are the predicted measurement covariance matrix and the state-measurement cross-covariance matrix, respectively.

5. Experiments and Discussion

To verify the effectiveness of the proposed algorithm, a field experiment was performed on the fourth floor of the School of Mechanical and Electrical Engineering building on the campus of China University of Mining and Technology. The experimental layout is shown in Figure 5. The experiments were conducted with a VLP-16 LiDAR (Velodyne, Morgan Hill, CA, USA) and Spatial FOG IMU (Advanced Navigation Company, Sydney, Australia). The specifications of the IMU and LiDAR are listed in Table 1.

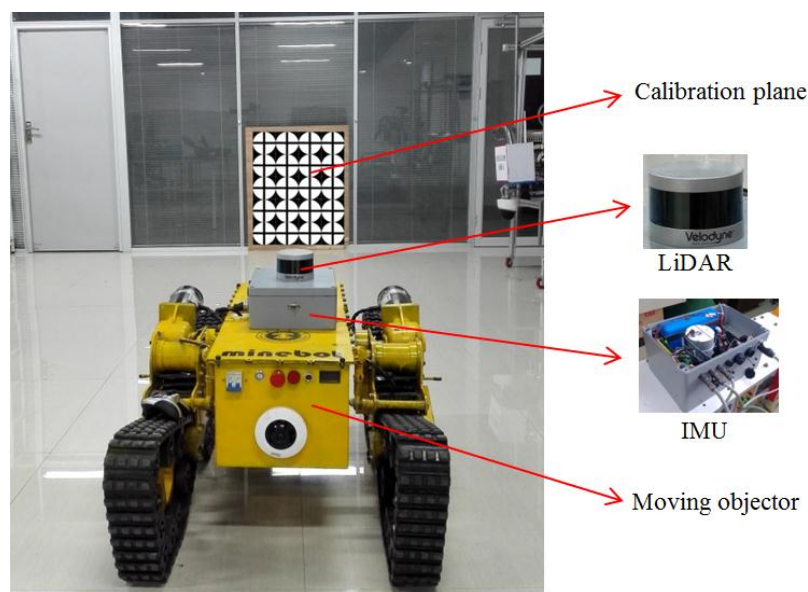


Figure 5. Experiment layouts.

Table 1. The specifications of the IMU and LiDAR.

IMU		LiDAR			
Navigation	Sensors Accelerometers Gyroscopes				
Horizontal Position Accuracy: 0.5 m	Range	10 g	490°/s	Channels	16
Vertical Position Accuracy: 0.8 m	Bias Instability	15 μ g	0.05°/h	Range	100 m
Velocity Accuracy: 0.007 m/s	Initial Bias	<1 mg	<1°/h	Accuracy	\pm 3 cm
Roll & Pitch Accuracy: 0.01°	Scaling Error	<0.03%	<0.01%	Vertical FOV	30°
Heading Accuracy: 0.05°	Scale Stability	<0.04%	<0.02%	Horizontal FOV	360°
Output Data Rate: Up to 1000 Hz	Non-linearity	<0.03%	<0.005%	Output Data Rate	300,000 pts/s

The LiDAR consists of 16 laser scanners that collectively span a 27° of vertical field of view. Firstly, in each experiment, we initialize the IMU biases and keeping the IMU stationary for 3 s. We begin recording data from the LiDAR and the IMU, the moving object carrying the LiDAR-IMU platform automatically moves following different configurations in front of the calibration plane. The IMU update data is recorded at 100 Hz, while the LiDAR scanning data rate is 100,000 points/s. We chose a set of three different trajectories for the LiDAR-IMU, and for each trajectory, we experimented with a different time shift and a different initial orientation estimate. In all trajectories, the initial true relative orientation of the IMU with respect to the LiDAR was the same. In the experiments, the moving object ran three different trials, with the LiDAR-IMU platform translating and rotating in front of the calibration plane for approximately 5 s, and the LiDAR-IMU platform then ran in three different orientation and time delay configurations. For configuration 1, we set the IMU-camera relative

orientation to a nominal value (a roll of 90° , pitch of 0° , and yaw of 90°), and fixed the estimated time delay value at zero. For configuration 2, we used the mean LiDAR-IMU relative orientation computed by averaging the roll, pitch, and yaw values, while again fixing the time delay at zero. Finally, for configuration 3, we used the averaged LiDAR-IMU relative orientation and the mean time delay value from configuration 2, and we determined the performance of each configuration by computing the RMS error. Once the LiDAR and IMU measurements for each configuration of the calibration plane were available, we used the method described in Section 4 to accurately estimate the LiDAR-IMU time delay calibration parameters and the LiDAR-IMU transformation parameters.

The experimental results for LiDAR-IMU time delay calibration as shown in Figure 6. Figure 6a,d,g, shows the initial alignment between IMU and LiDAR in East, North and Up orientation respectively. Figure 6b,e,h, shows one time alignment using ICP-ISP KF between IMU and LiDAR in East, North and Up orientation, respectively. Figure 6c,f,i, shows ten times alignment using ICP-ISP KF between IMU and LiDAR in East, North and Up orientation respectively. We examined the error in the LiDAR-IMU time delay calibration using ICP-ISP KF for ten times, as shown in Table 2.

Table 2. Time delay calibration times using ICP-ISP KF for LiDAR-IMU.

Time Delay Calibration Times Using ICP-ISP KF	Time Delay Error (ms)	Alignment Error in East (m)	Alignment Error in North (m)	Alignment Error in Up (m)
0	9.58	0.093	0.168	0.089
1	4.67	0.067	0.097	0.063
2	2.45	0.043	0.068	0.041
3	1.66	0.037	0.047	0.035
4	1.17	0.031	0.039	0.029
5	0.87	0.026	0.032	0.025
6	0.63	0.023	0.027	0.021
7	0.57	0.020	0.024	0.019
8	0.53	0.019	0.021	0.018
9	0.51	0.018	0.020	0.018
10	0.50	0.018	0.019	0.017

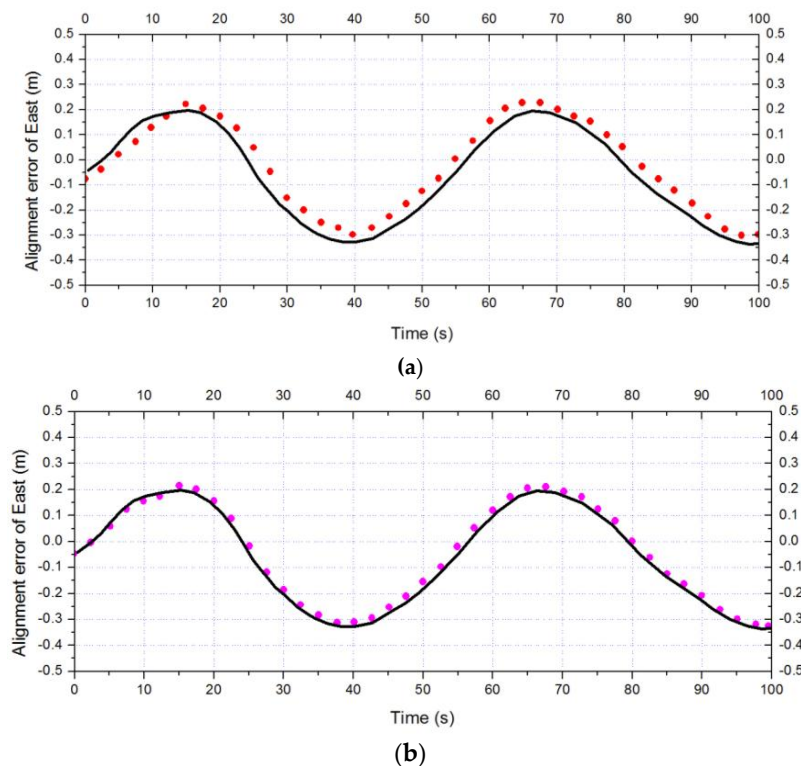


Figure 6. Cont.

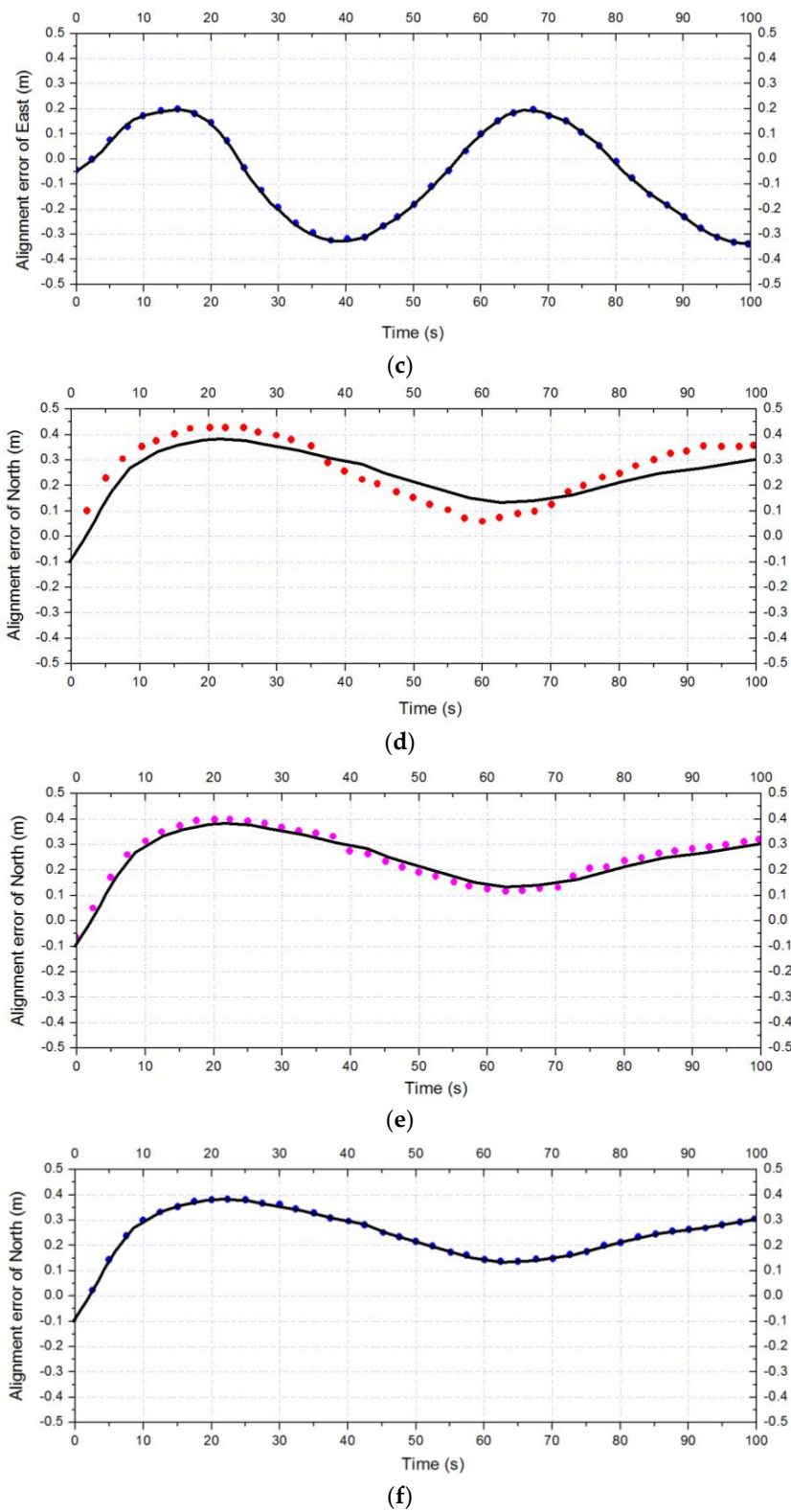


Figure 6. Cont.

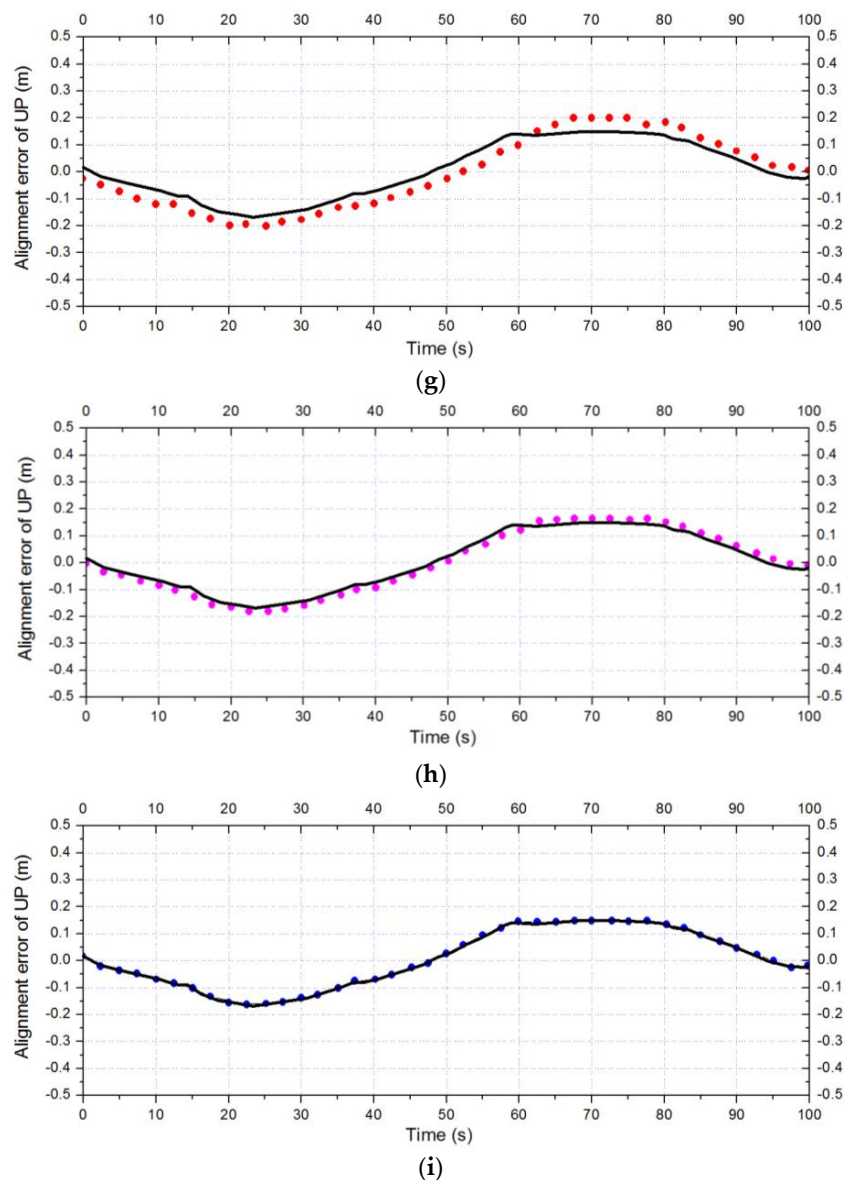


Figure 6. Time delay calibration using ICP-ISP KF for LiDAR-IMU. **(a)** Initial alignment between LiDAR (red dots) and IMU (black line) in East orientation; **(b)** time delay calibration one time using ICP-ISP KF between LiDAR (magenta dots) and IMU (black line) in East orientation; **(c)** time delay calibration ten times using ICP-ISP KF between LiDAR (blue dots) and IMU (black line) in East orientation; **(d)** initial alignment between LiDAR (red dots) and IMU (black line) in North orientation; **(e)** time delay calibration one time using ICP-ISP KF between LiDAR (magenta dots) and IMU (black line) in North orientation; **(f)** time delay calibration ten times using ICP-ISP KF between LiDAR (blue dots) and IMU (black line) in North orientation; **(g)** initial alignment between LiDAR (red dots) and IMU (black line) in Up orientation; **(h)** time delay calibration one time using ICP-ISP KF between LiDAR (magenta dots) and IMU (black line) in Up orientation; **(i)** time delay calibration ten times using ICP-ISP KF between LiDAR (blue dots) and IMU (black line) in Up orientation.

According to the Table 2 and Figure 6, the initial time delay alignment error is 9.58 ms, the LiDAR-IMU alignment errors in the East, North and Up orientation are 0.093 m, 0.168 m, 0.089 m respectively. After one time calibration using ICP-ISP KF converged, the time delay alignment errors are reduced to 4.67 ms, and the alignment errors in the East, North and Up orientation are reduced to 0.067 m, 0.097 m, 0.063 m, respectively. For clarity, after ten times calibration using ICP-ISP KF

converged, the final time delay alignment error was reduced to 0.50 ms, and the final alignment errors in the East, North and Up orientation were reduced to 0.018 m, 0.019 m, 0.017 m, respectively.

In order to prove the efficiency and accuracy of the ISPKF method, we used the ISPKF, KF and EKF methods to estimate the LIDAR-IMU time delay calibration errors. Since obtaining the initial truth time delay value is very difficult, we supposed the initial time delay alignment errors between LiDAR and IMU are zero, when the LiDAR and IMU measurements for each configuration of the calibration plane were available, the ICP-KF, ICP-EKF and ICP-ISPKF method are used to perform the time delay calibration for LiDAR-IMU.

The experimental results are shown in Figure 7, where the red line is the time delay alignment error result using the ICP-KF method, the blue line is the time delay alignment error result using the ICP-EKF method, the green line is the time delay alignment error result using the ICP-ISPKF method, and the one time calibration mean alignment errors using the ICP-KF, ICP-EKF and ICP-ISPKF methods are 0.233 m, 0.151 m, 0.067 m, respectively.

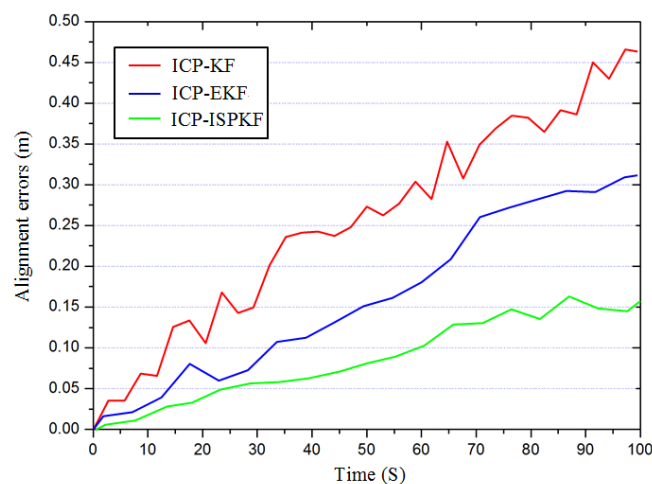


Figure 7. Time delay calibration using ICP-KF, ICP-EKF and ICP-ISPKF for LiDAR-IMU.

6. Conclusions

In order to solve the problem of LiDAR-IMU time delay calibration, we have presented a fusion method based on an iterative closest point algorithm and an iterated sigma point Kalman filter, which combines the advantages of ICP and ISPKE. The ICP algorithm can precisely determine the unknown transformation between LiDAR-IMU; and the ISPKE algorithm can optimally estimate the time delay calibration parameters. We realized the coordinate transformation from the LiDAR frame to the IMU frame, and established the measurement model and time delay error model of LiDAR and IMU. We also presented the methodology of the ICP and ISPKE procedure for the LiDAR-IMU time delay calibration. Intensive experimental studies were conducted to check the validity of the theoretical results. The results show that after ten times calibration using ICP-ISPKE converged, the time delay alignment error was reduced from 9.58 ms to 0.50 ms, and the alignment errors in the East, North and Up orientation were reduced from 0.093 m, 0.168 m, 0.089 m to 0.018 m, 0.019 m, 0.017 m, respectively. In the future work, the ICP-ISPKE algorithm will be used to calibrate the time delay error in other multi-sensor systems. Within the batch execution framework, different methods will be examined for incorporating varying time delays. We are convince that our proposed methods will be useful for other types of time delay calibration for multiple sensors.

Acknowledgments: This project is supported by National Natural Science Foundation of China (51304190, U1510116), and the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions. The authors would like to express their sincere thanks to them.

Author Contributions: Wanli Liu and Shibo Wang conceived and designed the experiments; Wanli Liu wrote the paper.

Conflicts of Interest: The author declares no conflict of interest.

References

- Gong, X.; Lin, Y.; Liu, J. 3D LIDAR-camera extrinsic calibration using an arbitrary trihedron. *Sensors* **2013**, *13*, 1902–1918. [[CrossRef](#)] [[PubMed](#)]
- Sim, S.; Sock, J.; Kwak, K. Indirect correspondence-based robust extrinsic calibration of LiDAR and camera. *Sensors* **2016**, *16*, 933. [[CrossRef](#)] [[PubMed](#)]
- Zhang, X.; Shi, H.; Pan, J.; Zhang, C. Integrated navigation method based on inertial navigation system and LiDAR. *Opt. Eng.* **2016**, *55*, 044102. [[CrossRef](#)]
- Yun, S.; Lee, Y.J.; Kim, C.J.; Sung, S. Integrated navigation design using a gimbaled vision/LiDAR system with an approximate ground description model. *Int. J. Aeronaut. Space Sci.* **2014**, *14*, 369–378. [[CrossRef](#)]
- Kelly, J.; Roy, N.; Sukhatme, G.S. Determining the time delay between inertial and visual sensor measurements. *IEEE Trans. Robot.* **2014**, *30*, 1514–1523. [[CrossRef](#)]
- Aghili, F.; Su, C.-Y. Robust Relative Navigation by Integration of ICP and Adaptive Kalman Filter Using Laser Scanner and IMU. *IEEE/ASME Trans. Mech.* **2016**, *21*, 2015–2026. [[CrossRef](#)]
- Anna, P.; Charles, T.; Spyros, K. On using QA/QC techniques for LiDAR/IMU boresight misalignment. In Proceedings of the 5th International Symposium on Mobile Mapping Technology, Padua, Italy, 19–23 May 2007.
- Deymier, C.; Teuliere, C.; Chateau, T. Self-calibration of a vehicle's acquisition system with cameras, IMU and 3D LiDAR. *Traitement Du Signal* **2015**, *32*, 121–145. [[CrossRef](#)]
- Tang, J.; Chen, Y.; Niu, X.; Wang, L.; Chen, L.; Liu, J.; Shi, C.; Hyyppä, J. LiDAR scan matching aided inertial navigation system in GNSS-denied environments. *Sensors* **2015**, *15*, 16710–16728. [[CrossRef](#)] [[PubMed](#)]
- Mirzaei, F.M.; Kottas, D.G.; Roulletiotis, S.I. 3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *Int. J. Robot. Res.* **2012**, *31*, 452–467. [[CrossRef](#)]
- Yun, S.; Lee, Y.J.; Sung, S. IMU/Vision/LiDAR integrated navigation system in GNSS denied environments. In Proceedings of the 2013 IEEE Aerospace Conference Proceedings, Big Sky, MT, USA, 2–9 March 2013.
- Li, R.; Liu, J.; Zhang, L.; Hang, Y. LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments. In Proceedings of the Inertial Sensors & Systems Symposium, Laguna Beach, CA, USA, 25–26 February 2014; pp. 1–15.
- Zhang, X.; Lin, Z. Online calibration method for IMU based on the usage of single-beam LiDAR. *Infrared Laser Eng.* **2013**, *42*, 466–471.
- Pham, D.D.; Suh, Y.S. Pedestrian navigation using foot-mounted inertial sensor and LIDAR. *Sensors* **2016**, *16*, 120. [[CrossRef](#)] [[PubMed](#)]
- Zhang, X.; Yin, J.; Lin, Z.; Zhang, C. A positioning and orientation method based on the usage of INS and single-beam LiDAR. *Optik* **2015**, *126*, 3376–3381. [[CrossRef](#)]
- Nguyen, H.-N.; Zhou, J.; Kang, H.-J. A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network. *Neurocomputing* **2015**, *151*, 996–1005. [[CrossRef](#)]
- Xian, Z.; Hu, X.; Lian, J. Fusing Stereo Camera and Low-Cost Inertial Measurement Unit for Autonomous Navigation in a Tightly-Coupled Approach. *J. Navig.* **2014**, *68*, 434–452. [[CrossRef](#)]
- Bing, A.; Sentis, L.; Paine, N.; Han, S.; Mok, A.; Fok, C.-L. Stability and Performance Analysis of Time-Delayed Actuator Control Systems. *J. Dyn. Syst. Meas. Control* **2016**, *138*, 1–20.
- Wu, Q.; Jia, Q.; Shan, J.; Meng, X. Angular velocity estimation based on adaptive simplified spherical simplex unscented Kalman filter in GFSINS. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2013**, *228*, 1375–1388. [[CrossRef](#)]
- Jwo, D.-J. Navigation Integration Using the Fuzzy Strong Tracking Unscented Kalman Filter. *J. Navig.* **2009**, *62*, 303–322. [[CrossRef](#)]
- Gaurav, P.; McBride, J.R.; Savarese, S.; Eustice, R.M. Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information. *J. Field Robot.* **2014**, *32*, 696–722.

22. García-Moreno, A.-I.; González-Barbosa, J.-J.; Ramírez Pedraza, A.; Hurtado Ramos, J.B.; Ornelas Rodriguez, F.J. Accurate evaluation of sensitivity for calibration between a LiDAR and a panoramic camera used for remote sensing. *J. Appl. Remote Sens.* **2016**, *10*, 024002. [[CrossRef](#)]
23. Klimkovich, B.V.; Tolochko, A.M. Determination of time delays in measurement channels during SINS calibration in inertial mode. *Gyroscopy Navig.* **2016**, *7*, 137–144. [[CrossRef](#)]
24. Huang, L.; Barth, M. A novel multi-planar LIDAR and computer vision calibration procedure using 2D patterns for automated navigation. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 117–122.
25. Zhou, L. A new minimal solution for the extrinsic calibration of a 2D LIDAR and a camera using three plane-line correspondences. *IEEE Sens. J.* **2014**, *14*, 442–454. [[CrossRef](#)]
26. Park, Y.; Yun, S.; Won, C.S.; Cho, K.; Um, K.; Sim, S. Calibration between color camera and 3D LIDAR instruments with a polygonal planar board. *Sensors* **2014**, *14*, 5333–5353. [[CrossRef](#)] [[PubMed](#)]
27. Kelly, J.; Sukhatme, G.S. Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *Int. J. Robot. Res.* **2011**, *30*, 56–79. [[CrossRef](#)]
28. Kelly, J.; Sukhatme, G.S. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In Proceedings of the 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Daejeon, Korea, 15–18 December 2009; Volume 30, pp. 360–368.
29. Sibley, G.; Sukhatme, G.; Matthies, L. The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo. *Robot. Sci. Syst.* **2006**, *8*, 235–244.
30. Farhad, A. 3D simultaneous localization and mapping using IMU and its observability analysis. *Robotica* **2011**, *29*, 805–814.
31. Kelly, J.; Sukhatme, G.S. A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors. In *Experimental Robotics*; Springer: Berlin, Germany, 2012; Volume 79, pp. 195–209.



© 2017 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).