sensors

MDPI

Article

Perception in the Dark; Development of a ToF Visual Inertial Odometry System

Shengyang Chen, Ching-Wei Chang and Chih-Yung Wen *

Deptartment of Mechanical Engineering and Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong; shengyang.chen@connect.polyu.hk (S.C.); chingwei.chang@connect.polyu.hk (C.-W.C.)

* Correspondence: cywen@polyu.edu.hk; Tel.: +852-34002522

Received: 11 January 2020; Accepted: 24 February 2020; Published: 26 February 2020

Abstract: Visual inertial odometry (VIO) is the front-end of visual simultaneous localization and mapping (vSLAM) methods and has been actively studied in recent years. In this context, a time-of-flight (ToF) camera, with its high accuracy of depth measurement and strong resilience to ambient light of variable intensity, draws our interest. Thus, in this paper, we present a realtime visual inertial system based on a low cost ToF camera. The iterative closest point (ICP) methodology is adopted, incorporating salient point-selection criteria and a robustness-weighting function. In addition, an error-state Kalman filter is used and fused with inertial measurement unit (IMU) data. To test its capability, the ToF–VIO system is mounted on an unmanned aerial vehicle (UAV) platform and operated in a variable light environment. The estimated flight trajectory is compared with the ground truth data captured by a motion capture system. Real flight experiments are also conducted in a dark indoor environment, demonstrating good agreement with estimated performance. The current system is thus shown to be accurate and efficient for use in UAV applications in dark and Global Navigation Satellite System (GNSS)-denied environments.

Keywords: VIO; ToF camera; real time; error-state Kalman Filter; data fusion; ICP

1. Introduction

The study is motivated and inspired by the need to design and build an efficient and low-cost robot-mount localization system for indoor industrial facility inspection, tunnel inspection, and search and rescue missions inside earthquake- or explosion-damaged buildings in poor lighting and GNSS-denied conditions. The system must be able to track the movement of the inspection vehicle in real time without the assistance of GNSS, GPS or other localization infrastructure. A vision based state estimator is particularly suitable for such a scenario.

Consequently, vision-based state estimation has become one of the most active research areas in the past few years and there have been many notable VO, VIO or VSLAM works, such as parallel tracking and mapping (PTAM) [1], SVO+MSF (semidirect visual odometry + multiple sensor fusion) [2–4], MSCKF (multi-state constrain Kalman filter) [5], OKVIS (open keyframe-based visual–inertial SLAM) [6], VINS (visual–inertial navigation system) [7,8], VI-ORB-SLAM (visual–inertial oriented FAST and rotated BRIEF-SLAM) [9], VI-DSO (visual–inertial direct sparse odometry) [10] and KinectFusion [11]. These works can be categorized by their pose estimation methods and the ways of fusing IMU data (Table 1). Currently, approaches to depth information based pose estimation can generally use one of two optimization methods: the direct optimization method and the iterative closest point (ICP) method.

Table 1. Different VIO frameworks.

Estimation Method and Input / IMU Fusion Method			Loose-Coupled	Tight-Coupled
Direct method		RGB/Grey	SVO+MSF [3,4]	VINS [7] [8] VI-DSO [10]
Image-based method	Feature based	RGB/Grey	MSCKF [5]	VI-ORB [12] OKVIS [6]
	ICP	Depth + RGB/Grey	KinectFusion [11] (no IMU support)	
		Depth + Grey	(current work)	n.a.
Others	NDT	RGB/Grey + Lidar	3D-NDT [13], Direct Depth SLAM [14]	

The optimization method models the estimation task as a minimization problem [15,16]. Generally, two kinds of optimization methods are widely used: Image-based optimization method and direct optimization method. In the traditional image-based optimization method, the objective function is modeled by the re-projection error of the features through feature matching; while in the direct optimization method, with the photometric camera model, the objective function is chosen as the intensity residual between frames. A comprehensive comparison of these two optimization methods can be found in Delmerico and Scaramuzza [17]. The target functions are solved by the gradient descent optimizer. As the image transformation in the traditional image-based method is represented by the Jacobian matrix in the objective function, which is the first order linearization of the real model, this method performs poorly in scenarios involving significant rotation, due to the implicitly high non-linearity in the real model.

Apart from the traditional image-based optimization method, the ICP method is considered as a relatively new type of its kind. The ICP method relies on the classic registration workflow [18], where the ICP algorithm begins with two sequential point clouds and an initial prediction of the transformation. The correspondence between two images (i.e., the source point cloud and the target point cloud) is found by searching the closest neighborhoods of the source points and estimating the progressive transformation to minimize the distances between the source points and target points. The iteration process is continued until transformation convergence is achieved. The value of the initial guess for the ICP is important, as a poor guess may lead to the iterative solution becoming trapped in a local minimum. Also, the ICP require high computation power especially when the number of points increase.

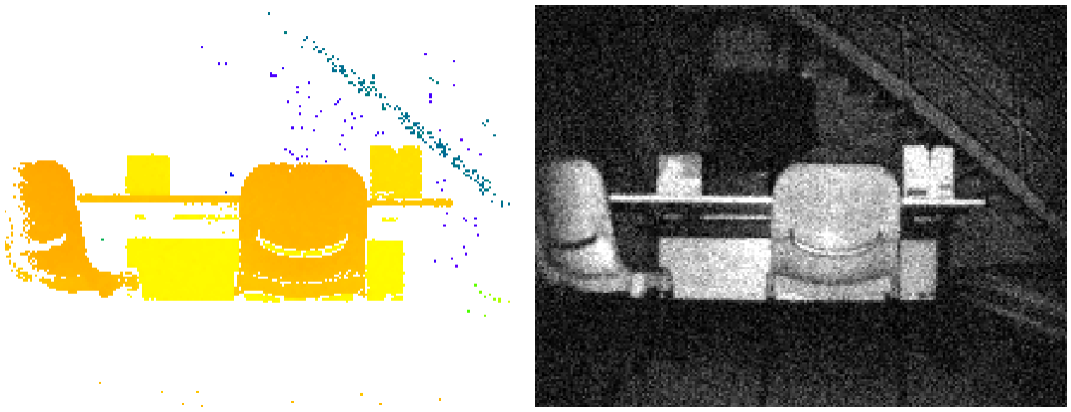
Additionally, there are some others, such as normal distribution transformation (NDT), in which the transform is calculated based on the probability density function inside the celled map. This kind of methods require huge amount of depth data and is normally used in the Lidar sensor applications. The notable works using the NDT method include: 3D-NDT [13,19] and Direct Depth SLAM [14].

Most of the methods described above use passive cameras and were developed for use under different experimental environments in daylight, meaning that camera exposure for good image quality is not a problem. However, many real application scenarios, such as indoor industrial facility inspection and search and rescue missions inside earthquake- or explosion-damaged buildings, are much more challenging than the benchmark case of the MH 05 difficult in EuRoC MAV Dataset [20], as these are in poor lighting conditions. In order to achieve perception in such dark or changing ambient light environments, we replace the conventional passive camera with an active ToF camera in this study. The pros and cons of using the conventional passive camera and the active ToF camera in motion estimation are listed in Table 2.

Table 2. Pros and cons of using passive camera and active ToF camera in motion estimation.

	Passive Camera	**Active ToF Camera**
pros	Less expensive in price Studied for years and with many CV algorithm Relatively high resolution	Strong resilience to ambient light Sense depth infomation without post-processing
cons	Fail in poor/changing lighting conditions Need initialization process to recover the depth	Limit sensing distance (less than 10 m) Current CV algorithm can not used on the NIR image Relatively low resolution

Notably, in a passive camera, the sensor is exposed to the light during the shutter opening period. The depth information can be triangulated through the motion of the camera or using two cameras with the calibrated extrinsic parameters. Contrarily, the ToF camera illuminates a modulated light and observes the reflected light. The phase shift between the illuminating light and the reflected light is measured and translated to distance [21]. Since such a ToF camera can measure the distance in "one shot" without any post processing, it enables good resilience to any change of environmental brightness [22]. The output of the ToF camera provides a depth image and an NIR intensity image simultaneously (Figure 1).



(**a**) (**b**)

Figure 1. PMD Flexx ToF camera output (224 x 171 pixels). (**a**) Depth image, white color mean depth information is not available. (**b**) NIR intensity image.

In this paper, we propose an optimized VIO framework using a combination of a pmd flexx ToF camera and Pixhawk v2 hardware as the IMU sensor. Salient point selection with a statistic based weight factor is applied in the basic ICP algorithm, and the estimation results are fused with the IMU data using an error state Kalman filter. The algorithm can achieve a realtime processing rate for UAV applications without needing to use a graphics-processing unit. The main contributions of this work are:

- Improved the conventional ICP workflow with the salient point selection criterias and the statistics based robust weighting factor.
- Development of an error state Kalman filter based framework to fuse global sensors, composed of a ToF camera and an IMU, and with local estimations, which achieves locally accurate localization and high computational efficiency.
- An evaluation of the proposed system in both the motion capture system and the real experiments. A ToF-IMU dataset with the ground truth is published.
- Open-source code of the error state Kalman filter based ToF–VIO for the research community.

2. ToF–VIO Platform

Our sensor platform consists of a pmd technologies Flexx ToF camera and an IMU sensor embedded in a Pixhawk v2 flight controller, as shown in Figure 2. The Flexx ToF camera has a resolution of 224 × 171 pixels and a depth detection range of 4 m with a frame rate set to be 15 Hz. The Pixhawk v2 IMU updates at a rate of 250 Hz. The times of the two sensors are synchronized by the software and the installation geometry of the camera and IMU is represented by (1).

$$\boldsymbol{T}_{IC} = \begin{bmatrix} 0 & 0 & 1 & 0.1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

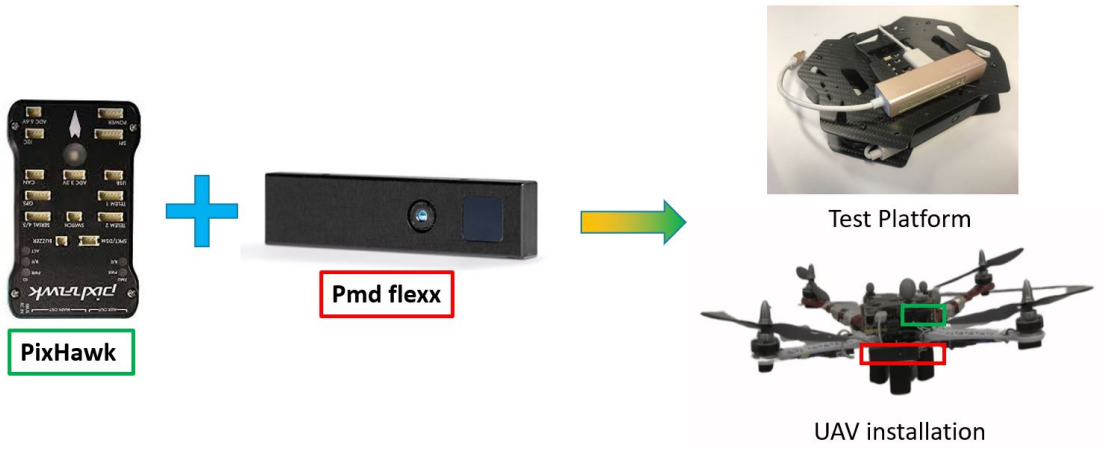


Figure 2. ToF–VIO testing platform and its implementation on a UAV.

3. ICP Alignment

3.1. Visual Odometry by the ToF Camera Module

The ToF camera outputs a depth image $p_z = (u, v, z)$ and a near infrared (NIR) intensity image $p_i = (u, v, i)$ simultaneously, where z and i represent the depth and intensity information, respectively, of the pixel (u, v). Because these two images are captured by the same optical sensing module and aligned, they share the same camera parameters (f_x, c_x, f_y, c_y), where f and c are the focal length and the image center, respectively. For simplicity, we can use $p = (u, v, z, i)$ to represent the camera output, which is linked to its corresponding 3D point in the Camera Frame with the intensity, $P_c = (X_c, Y_c, Z_c, I)^T$, through the projection model of the camera as shown in Equation (2). Accordingly, $P_c = \pi(p)$ can be derived explicitly in Equation (3).

$$\begin{bmatrix} u \\ v \\ 1 \\ z \\ i \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 & 0 \\ 0 & f_y & c_y & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \\ Z_c \\ I \end{bmatrix} \tag{2}$$

$$P_c = \pi(p) = (\frac{u \cdot f_x}{z} + c_x, \frac{v \cdot f_y}{z} + c_y, z, i)^T \tag{3}$$

Assuming the world frame is fixed, the rigid motion of an object in front of a still camera can be described by a transformation matrix $\boldsymbol{T} = [\boldsymbol{R}|t]$, where $\boldsymbol{R}$ is the rotational matrix in $SO(3)$ and t

represents the translational vector. Equation (4) describes such a transformation from the world frame to the camera frame, where $P_w = (X_w, Y_w, Z_w, I)^T$ is the 3D point in the world Frame with its intensity.

$$P_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ I \end{bmatrix} = \boldsymbol{T} \cdot P_w = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ I \end{bmatrix} \tag{4}$$

As shown in Figure 3, a moving camera captures the point clouds $\boldsymbol{P}_{t1}$ and $\boldsymbol{P}_{t2}$ at two sequent time $t1$ and $t2$, respectively. Notably, $\boldsymbol{P}$ and P represent the point cloud and a single point, respectively. The captured point cloud, $\boldsymbol{P}_{t1}$ can be represented by Equation (5), while $\boldsymbol{P}_{t2}$ can be regard as the inter-frame transformation of $\boldsymbol{P}_{t1}$ due to the motion of the camera in the world frame, as seen in Equation (6). Here, $\boldsymbol{T}_{cam}$ represents the coordinate transformation of the camera in the world frame between $\boldsymbol{P}_{t1}$ and $\boldsymbol{P}_{t2}$.

$$\boldsymbol{P}_{t1} = \boldsymbol{T}_{t1} \cdot \boldsymbol{P}_w \tag{5}$$

$$\boldsymbol{P}_{t2} = \boldsymbol{T}_{cam}^{-1} \boldsymbol{P}_{t1} = \boldsymbol{T}_{cam}^{-1} \cdot \boldsymbol{T}_{t1} \cdot \boldsymbol{P}_w \tag{6}$$

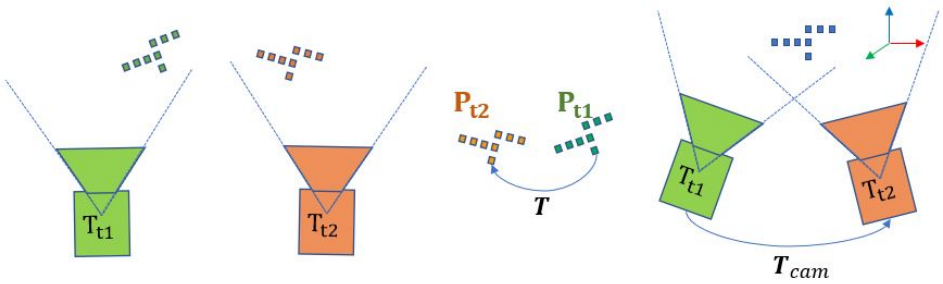


Figure 3. The camera pose in the world frame and the point cloud alignment procedure in the camera frame.

The frame to frame alignment can also be made through the transformation matrix $\boldsymbol{T}$, which aligns two point clouds as $\boldsymbol{P}_{t2} = \boldsymbol{T} \cdot \boldsymbol{P}_{t1}$. Combining it with Equation (5) and Equation (6), we could find the motion of the camera T_{cam} equal to the inverse of the point cloud transformation matrix $\boldsymbol{T}$ (Equation (7)). In another word, the camera motion can be derived from the point cloud alignment.

$$\boldsymbol{P}_{t2} = \boldsymbol{T}_{cam}^{-1} \cdot \boldsymbol{T}_{t1} \cdot \boldsymbol{P}_w = \boldsymbol{T} \cdot \boldsymbol{T}_{t1} \cdot \boldsymbol{P}_w \Rightarrow \boldsymbol{T}_{cam} = \boldsymbol{T}^{-1} \tag{7}$$

In this work, the ICP algorithm are used to derive the transformation $\boldsymbol{T}$. The workflow of the basic ICP algorithm can be summarized as:

- The point cloud alignment algorithm starts with the source point cloud $\boldsymbol{P}_s$ and the target point cloud $\boldsymbol{P}_t$ and an initial prediction of the transformation $\boldsymbol{T}_0$ between these two clouds.
- For every point in the target cloud ($P_i \in \boldsymbol{P}_t$), searching the corresponding point P_i' in the transformed source point cloud which has the closest distance:

$$P_i' = P_j \ , \ j = \underset{j}{\operatorname{argmin}} ||P_i - \boldsymbol{T} \cdot Pj|| \ , \ P_j \in \boldsymbol{P}_s \tag{8}$$

- Estimate the increment transformation from the point pairs which could minimize the error metric.

$$\Delta \boldsymbol{T} = \underset{\Delta \boldsymbol{T}}{\operatorname{argmin}} ||\boldsymbol{P} - \Delta \boldsymbol{T} \cdot \boldsymbol{P}'|| \tag{9}$$

- Applying the increment transformation.

$$\boldsymbol{T}^{n+1} \leftarrow \Delta \boldsymbol{T} \cdot \boldsymbol{T}^{n} \tag{10}$$

- By applying the above steps iteratively until the transformation of the two point clouds converge (Equation (11), where $\boldsymbol{T}_{TH}$ is the threshold of transformation) or meets a certain criterion, e.g reach maximum iterative number, the final transformation is obtained.

$$\Delta \boldsymbol{T} < \boldsymbol{T}_{TH} \tag{11}$$

Many variants of the basic ICP concept have been introduced to enhance the performance, i.e, to speed up the algorithm, improve robustness or increase accuracy. In the work of Rusinkiewicz and Levov [23], they classify these variants according to the effect on one of six stages of the algorithm: Selection, Matching, Weighting, Rejecting, Assigning Error Metric and Minimizing the error metric. Inspired by the work of Li and Lee [24], we develop a salient point selection criteria for acceleration of the ICP process. The statics based weighting function to improve the robustness of the ICP process is applied.

3.2. Selection of Sailent Points

The resolution of ToF camera in this study is 38K, i.e, every point cloud image contains 38K points. It is impossible to align such a huge amount of points on an embedded computer in real time. Therefore, several criteria to select the salient points from the raw image are applied in this work. As shown in Figure 4, these criteria can be divided into the rejection and acceptance groups.

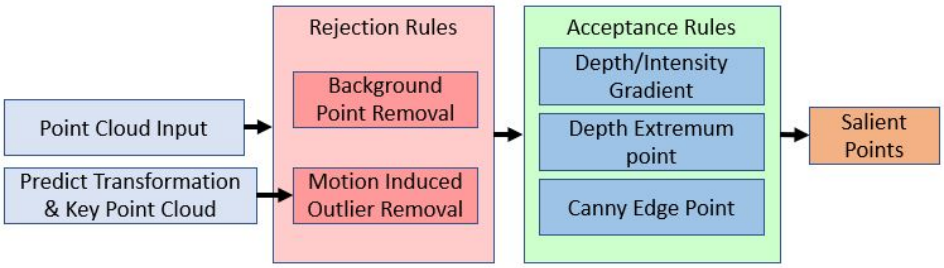


Figure 4. Workflow of sailent point selection.

3.2.1. Rejection Rules

Rejection Rules: Two kinds of rejection rules are applied: background points and motion induced outliers. A cloud point which satisfies either of these two criteria will be rejected.

Background Point

As shown in Figure 5, the background points might be blocked by the movement of the camera, compared to the object point (which are consistent in different frames). As these background points will introduce the error into the point cloud alignment, they must be removed.

The background points are directly identified from the depth image. Thus, if an abrupt decrease of depth is found to occur in the area nearby a single point, this point is inferred to be a background point. These criteria can be expressed by Equation (12), where $z(u, v)$ represents the depth value z value of the point $p = (u, v, z, i)^T$ at the (u, v) position and $\pi_{sh,bg}$ is a threshold. Notably, relevant background points tend to exist near the edges of an object; this means that comparing $z(u, v)$ with that of the point which is four pixels away is the most effect means of background point removal. In addition, $\pi_{sh,bg}$ is relevant to the detection range of the selected camera (see Section 5 Table 3).

$$\begin{aligned} z(u,v) - z(u+4,v) &> \pi_{sh,bg} * z(u,v) \quad \vee \\ z(u,v) - z(u-4,v) &> \pi_{sh,bg} * z(u,v) \quad \vee \\ z(u,v) - z(u,v+4) &> \pi_{sh,bg} * z(u,v) \quad \vee \\ z(u,v) - z(u,v-4) &> \pi_{sh,bg} * z(u,v) \end{aligned} \tag{12}$$

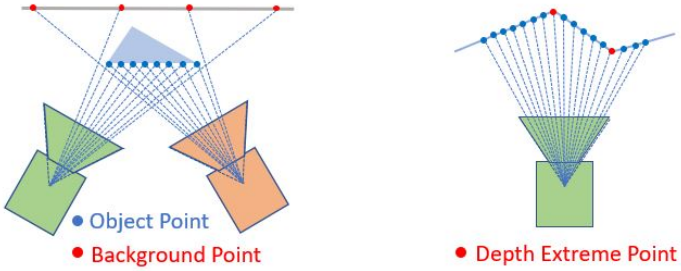


Figure 5. Background points and Depth extreme points.

Motion Induced Outlier

The initial coordinate transformation $\boldsymbol{T}^0$ can be predicted by the IMU integration process (which will be introduced in the next section) or using the transformation of previous frame. Together with the projection model of the camera Equations (2)–(4), the predicted positions of the points in the new frame can be calculated from the transformation of the source point cloud $\boldsymbol{P}_s$ by Equation (13). If this predicted point lays out of the FOV (Field-of-View) of the camera (Equation (14)), it will be removed before the ICP alignment. Note that this step is implemented at the step of making point pairs.

$$p = (u,v,z,i)^T = \pi^{-1}(\boldsymbol{T}^0 \cdot P_s) \tag{13}$$

$$\begin{aligned} u_{predict} < 0 \ \vee \ u_{predict} > u_{max} \ (i.e., width) \ \vee \\ v_{predict} < 0 \ \vee \ v_{predict} > v_{max} \ (i.e., height) \end{aligned} \tag{14}$$

3.2.2. Acception Rules

Three kinds of acceptance rules are further applied: gradient-based criteria, depth extreme points and canny features. Any point satisfies one of these three rules will be accepted. The relationship of the rejection rule and acceptance rule is "and,∧", which means that a salient point needs to satisfy the acceptance rules and it is not a rejected point.

Gradient-Based Criteria

The position where the Depth or Intensity changed dramatically are believed to carry the important position information or features. Thus, the criteria based on the intensity gradient (Equation (15)) and the depth gradient (Equation (16)) gradient are developed.

$$\begin{aligned} |i(u+2,v) - i(u-2,v)| &> \pi_{sh,i} \quad \vee \\ |i(u,v+2) - i(u,v-2)| &> \pi_{sh,i} \end{aligned} \tag{15}$$

$$\begin{aligned} |z(u+2,v) - z(u-2,v)| &> \pi_{sh,z} * z(u,v) \quad \vee \\ |z(u,v+2) - z(u,v-2)| &> \pi_{sh,z} * z(u,v) \end{aligned} \tag{16}$$

Depth Extreme Point

As shown in Figure 5, the depth extreme points are considered to contain 3D features and thus need to be select as salient points. The extreme point of the depth (the local maximum of minimum

of the depth) on a continuous plane can be found by a zero depth gradient $g_u(u,v) = \partial(u,v)/\partial u = 0$ or $g_v(u,v) = \partial(u,v)/\partial v = 0$. As the cloud points are discrete, the extreme points are detected by calculating monotonically the gradient in an interrogation window of 5×5 pixels. Thus, the points which satisfy Equations (17) and (18) are considered to be the extreme points in the u direction. Similar procedures are conducted to extract the extreme points in the v direction.

$$g_u(u,v) = z(u+1,v) - z(u,v) \tag{17}$$

$$\begin{aligned} &(g_u(u-2,v) < 0 \ \wedge \ g_u(u-1,v) < 0 \ \wedge \\ &\ g_u(u,v) > 0 \quad \wedge \ g_u(u+1,v) > 0) \quad \vee \\ &(g_u(u-2,v) > 0 \ \wedge \ g_u(u-1,v) > 0 \ \wedge \\ &\ g_u(u,v) < 0 \quad \wedge \ g_u(u+1,v) < 0) \end{aligned} \tag{18}$$

Canny Features

The Canny edge detector is believed to be one of the most popular edge detection methods ever since it was developed [25]. It has been shown that the introduction of the canny feature will increase the alignment accuracy. Therefore, this detector is applied to the NIR image to extract the edge points as a supplement of salient point detector. Once the canny edge is detected, the corresponding points in the point cloud will be accepted. The Canny detector form OpenCV with the parameter: $threshold1 = 150$, $threshold2 = 300$ and $apertureSize = 3$ are used in this work.

3.3. Weighting of the Point Pairs

In Kerl's work [15], intensity residuals are found to follow the t-distribution approximately. We measured the position error distribution of the point pairs from two images. The results of three different cases are shown in Figure 6. In first case, the camera is fixed and the point cloud is captured twice. The second and third cases show the position error distributions of the point clouds by a moving camera with and without running four ICP loops, respectively. As shown in Figure 5, the t-distribution approximates the error distribution better than the normal distribution in every case.

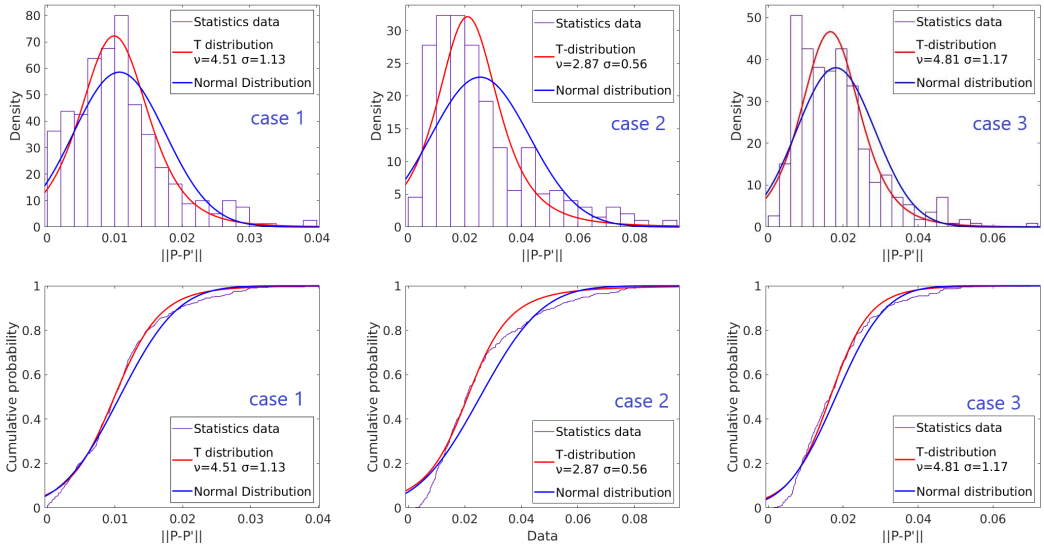


Figure 6. Distribution and approximate distribution curve of the $||\boldsymbol{P}_i - \boldsymbol{P}_i'||$ in different cases. Uper figures plot the probability density function(PDF) and lower figures plot the cumulative distribution function (CDF).

Base on our observation of distribution, we modify the error matrix (Equation (9)) with the robust weighting factor:

$$\Delta \boldsymbol{T} = \underset{\Delta \boldsymbol{T}}{\operatorname{argmin}} \sum_{i}^{size(\boldsymbol{P})} w(i)||P_i - \Delta \boldsymbol{T} \cdot P_i'|| \tag{19}$$

$$w(i) = \frac{\nu + 1}{\nu + (\frac{||P_i - \boldsymbol{T} \cdot P_i'|| - u}{\sigma})^2} \tag{20}$$

In Equation (19), ν is the degree of freedom. As indicated in Figure 6, ν is close to 2 when a large motion occurs and the alignment of two frames is moderate; while ν increases when the alignment is improved. In the following experiments, $\nu = 4$ is set for the weight factor calculation. Assuming the point cloud are well aligned, the mean of the error matrix u is set to zero. The deviation can then be calculated by the following equation recursively:

$$\sigma = \frac{1}{n} \sum_{i=1}^{n} ||P_i - \boldsymbol{T} \cdot P_i'|| \frac{\nu + 1}{\nu + (\frac{||P_i - \boldsymbol{T} \cdot P_i'||}{\sigma})^2} \tag{21}$$

4. Data Fusion

4.1. Modelling Equations of IMU Sensors

The MEMS IMU sensor consists of a 3-axis gyroscope and a 3-axis accelerometer. The sensor measurements of the angular velocities by the gyroscope $\boldsymbol{\omega}_m = (\omega_x, \omega_y, \omega_z)^T$ and the accelerations by the accelerometer $\boldsymbol{a}_m = (a_x, a_y, a_z)^T$ can be described by the following model:

$$\begin{aligned} &\boldsymbol{\omega}_m = \boldsymbol{\omega}_{real} + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n \\ &\boldsymbol{a}_m = \boldsymbol{R}(\boldsymbol{q})(\boldsymbol{a}_{real} - \boldsymbol{g}) + \boldsymbol{a}_b + \boldsymbol{a}_n \\ &\boldsymbol{\omega}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_{\omega}^2) \qquad \dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_{bn} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_{\omega_b}^2) \\ &\boldsymbol{a}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_{a}^2) \qquad \dot{\boldsymbol{a}}_b = \boldsymbol{a}_{bn} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_{a_b}^2) \end{aligned} \tag{22}$$

where $\boldsymbol{\omega}_{real}$ and $\boldsymbol{a}_{real}$ represent the true angular velocity and acceleration, respectively; $\boldsymbol{\omega}_n$ and $\boldsymbol{a}_n$ refer to the additive noises of the sensor which follow the Gaussian distributions in nature; and the bias part $\boldsymbol{\omega}_b$ and $\boldsymbol{a}_b$ can be described as random walk processes. The derivatives of the gyroscope bias $\boldsymbol{\omega}_{bn} = \dot{\boldsymbol{\omega}}_b$ and the accelerometer bias $\boldsymbol{a}_{bn} = \dot{\boldsymbol{a}}_b$ also follow the Gaussian distributions. The error variance $\boldsymbol{\sigma}_{\omega}^2$ and $\boldsymbol{\sigma}_{a}^2$ can be found in the datasheet of the IMU. Some high precision IMUs also provide the $\boldsymbol{\sigma}_{\omega_b}^2$ and $\boldsymbol{\sigma}_{a_b}^2$. If not, this two values can be derived from the IMU calibration process or one can set them as the squares of additive noises. $\boldsymbol{q} = (q_w, q_x, q_y, q_z)^T$ is the quaternion representation of rotation from the inertial frame to the IMU frame. $\boldsymbol{R}(\boldsymbol{q})$ is the rotation matrix corresponding to the quaternion vector.

4.2. Error State Kalman Filter

The Error State Kalman filter (ESKF) is adopted to estimate the errors in every state by using the differences between the IMU data and the Frame to Frame Alignment results. Figure 7 shows the workflow of the proposed estimator. When an IMU data are fed in, the integration process will integrate the nominal state to provide a prediction pose for the frame to frame alignment module. At the same time, the error state will also be updated according to the ESKF processing model. After finishing the alignment process, the output of the alignment module will serve as the measurement to correct the error state. Finally, the new state will be calculated by the composition of the error state and the nominal state.

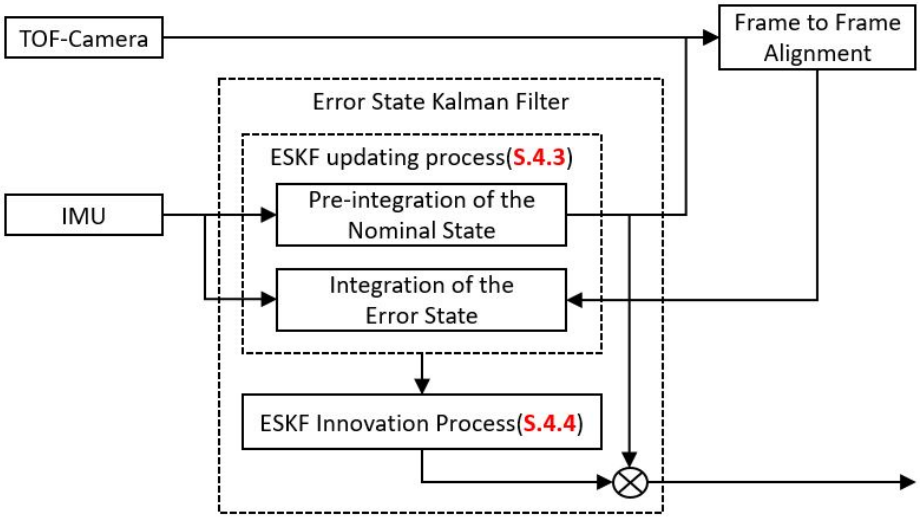


Figure 7. Workflow of the Error State Kalman Filter estimator by using the ToF camera alignment results and the IMU data.

In the current work, the quaternion (rotation) $\boldsymbol{q}$, position $\boldsymbol{p}$, velocity $\boldsymbol{v}$, together with the biases of the gyroscope $\boldsymbol{\omega}_b$ and accelerometer $\boldsymbol{a}_b$ are used to describe the system. Following the notation style of Santamaria-Navarro et al. [26], The total system state is described by the true state $\boldsymbol{x}_t = (\boldsymbol{q}_t, \boldsymbol{p}_t, \boldsymbol{v}_t, \boldsymbol{\omega}_{bt}, \boldsymbol{a}_{bt})$, nominal state $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p}, \boldsymbol{v}, \boldsymbol{\omega}_b, \boldsymbol{a}_b)$ and error state $\delta\boldsymbol{x} = (\delta\boldsymbol{\theta}, \delta\boldsymbol{p}, \delta\boldsymbol{v}, \delta\boldsymbol{\omega}_b, \delta\boldsymbol{a}_b)$. In the error state, the Euler angles are selected to represent the rotation so that the dimension of the rotation component is 3. The composition of the state follows:

$$\begin{aligned} \boldsymbol{q}_t &= \delta\boldsymbol{q} \otimes \boldsymbol{q}, & \delta\boldsymbol{q} &= \boldsymbol{q}(\delta\boldsymbol{\theta}) = e^{\delta\boldsymbol{\theta}/2} \\ \boldsymbol{p}_t &= \boldsymbol{p} + \delta\boldsymbol{p}, & \boldsymbol{v}_t &= \boldsymbol{v} + \delta\boldsymbol{v} \\ \boldsymbol{\omega}_{bt} &= \boldsymbol{\omega}_b + \delta\boldsymbol{\omega}_b, & \boldsymbol{a}_{bt} &= \boldsymbol{a}_b + \delta\boldsymbol{a}_b \end{aligned} \tag{23}$$

where $\otimes$ indicates quaternion multiplication. In the ESKF fusion, the inputs are divided into 3 parts: $\boldsymbol{u}$,$\boldsymbol{n}$ and $\boldsymbol{z}$. The measurement input $\boldsymbol{u}$ contains the readout from the gyroscope and the accelerometer $\boldsymbol{u} = (\boldsymbol{\omega}_m, \boldsymbol{a}_m)$. The noise impulse input $\boldsymbol{n}$ is the assumed IMU noise $\boldsymbol{n} = (\boldsymbol{\omega}_n, \boldsymbol{a}_n, \boldsymbol{\omega}_{bn}, \boldsymbol{a}_{bn})$ and the observation $\boldsymbol{z}$ includes the rotation and translation from the ICP alignment process $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{t})$. According to the IMU measurement module, every component in the noise impulse input follows the Gaussian distribution with a zero mean value. The purpose of doing so is intended to separate the error part from other parts and let the error remain in the error state. The storage in the nominal state is always the best fusion result. The ESKF workflow can be divided into two steps: (1) the ESKF updating process driven by the IMU and (2) the ESKF innovation step from the IMU (see Figure 7), detailed as follows:

1. Update the nominal state through the nominal state processing model $\boldsymbol{x}_n = \boldsymbol{f}_{ns}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$ and update the error state and the covariance matrix through the error state processing model $\delta\boldsymbol{x}_n = \boldsymbol{f}_{es}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{n}_t)$
2. Innovate the Kalman Gain K, the error state and the covariance matrix, and then inject the error state to the nominal state

4.3. ESKF Updating Process

4.3.1. Kinematics of True and Nominal State

The system kinematics of the IMU can be describe by the equation:

$$\begin{aligned}
\dot{\boldsymbol{q}}_t &= \frac{1}{2}\boldsymbol{q}_t \otimes \boldsymbol{\omega}_t = \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) \\
\dot{\boldsymbol{p}}_t &= \boldsymbol{v}_t, \qquad \dot{\boldsymbol{v}}_t = \boldsymbol{R}_t{}^{-1}(\boldsymbol{a}_m - \boldsymbol{a}_{bt} - \boldsymbol{a}_n) + \boldsymbol{g} \\
\dot{\boldsymbol{\omega}}_{bt} &= \boldsymbol{\omega}_{bn}, \quad \dot{\boldsymbol{a}}_{bt} = \boldsymbol{a}_{bn}
\end{aligned} \tag{24}$$

where $\boldsymbol{q} \otimes \boldsymbol{\omega}$ represent $\boldsymbol{q} \otimes [0, \boldsymbol{\omega}]^T$ and the multiplication operation of the quaternion and the angular velocity can be calculated by $\boldsymbol{q}_t \otimes \boldsymbol{\omega}_t = \Omega(\boldsymbol{\omega}_t)\boldsymbol{q}_t$ where $\Omega(\boldsymbol{\omega})$ is the quaternion integration matrix:

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{25}$$

The nominal state kinematics can be updated following the system module. The noise of the IMU is not considered in this nominal state module, i.e., the nominal state is only updated by the measurement input $\boldsymbol{u}_t = (\boldsymbol{\omega}_m, \boldsymbol{a}_m)$. The kinematics in continuous time is shown as follows:

$$\begin{aligned}
\dot{\boldsymbol{q}} &= \frac{1}{2}\boldsymbol{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b) = \frac{1}{2}\Omega(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\boldsymbol{q} \\
\dot{\boldsymbol{p}} &= \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = \boldsymbol{R}(\boldsymbol{q})^{-1}(\boldsymbol{a}_m - \boldsymbol{b}_a) + \boldsymbol{g} \\
\dot{\boldsymbol{\omega}}_b &= 0, \qquad \dot{\boldsymbol{a}}_b = 0
\end{aligned} \tag{26}$$

In a discrete time, the nominal state can be calculated by:

$$\begin{aligned}
\boldsymbol{q} &\leftarrow \boldsymbol{q} \otimes \boldsymbol{q}\{(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\Delta t\} \\
\boldsymbol{p} &\leftarrow \boldsymbol{p} + \boldsymbol{v}\Delta t + \frac{1}{2}(\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) + \boldsymbol{g})\Delta t^2 \\
\boldsymbol{v} &\leftarrow \boldsymbol{v} + (\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) + \boldsymbol{g})\Delta t \\
\boldsymbol{\omega}_b &\leftarrow \boldsymbol{\omega}_b, \qquad \boldsymbol{a}_b \leftarrow \boldsymbol{a}_b
\end{aligned} \tag{27}$$

4.3.2. Updating Process of Error State and Covariance Matrix

The error state kinematics can be described by the following equations.

$$\begin{aligned}
\delta\dot{\boldsymbol{\theta}} &= -\boldsymbol{R}\delta\boldsymbol{\omega}_b - \boldsymbol{R}\boldsymbol{\omega}_n \qquad & \delta\dot{\boldsymbol{p}} &= \delta\boldsymbol{v} \\
\delta\dot{\boldsymbol{v}} &= -[\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b)]_\times\delta\boldsymbol{\theta} - \boldsymbol{R}\delta\boldsymbol{a}_b - \boldsymbol{R}\boldsymbol{a}_n & & \\
\delta\dot{\boldsymbol{\omega}}_b &= \boldsymbol{\omega}_{bn} & \delta\dot{\boldsymbol{a}}_b &= \boldsymbol{a}_{bn}
\end{aligned} \tag{28}$$

The equations of $\delta\dot{\boldsymbol{p}}$, $\delta\dot{\boldsymbol{\omega}}_b$ and $\delta\dot{\boldsymbol{a}}_b$ can be derived directly by the definition of the error state. The derivations of the error velocity $\delta\dot{\boldsymbol{v}}$ and the error orientation $\delta\dot{\boldsymbol{\theta}}$ can be found out in the Appendix A. The error state in the discrete time domain follows:

$$\begin{aligned}
\delta\boldsymbol{\theta} &\leftarrow \delta\boldsymbol{\theta} - \boldsymbol{R}\delta\boldsymbol{\omega}_b\Delta t - \boldsymbol{R}\boldsymbol{\omega}_n \qquad & \delta\boldsymbol{p} &\leftarrow \delta\boldsymbol{p} + \delta\boldsymbol{v}\Delta t \\
\delta\boldsymbol{v} &\leftarrow \delta\boldsymbol{v} + (-[\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{b}_a)]_\times\delta\boldsymbol{\theta} - \boldsymbol{R}\delta\boldsymbol{a}_b)\Delta t + \boldsymbol{R}\boldsymbol{a}_n & & \\
\delta\boldsymbol{\omega}_b &\leftarrow \delta\boldsymbol{\omega}_b + \boldsymbol{\omega}_{bn}, & \delta\boldsymbol{a}_b &\leftarrow \delta\boldsymbol{a}_b + \boldsymbol{a}_{bn}
\end{aligned} \tag{29}$$

By applying $\boldsymbol{n} = [\boldsymbol{\omega}_n, \boldsymbol{a}_n, \boldsymbol{\omega}_{bn}, \boldsymbol{a}_{bn}]^T$ as the input, which drives the system forward and induces the system transition matrix $\boldsymbol{F}$ and input matrix $\boldsymbol{F}_i$, The kinetics of the error state and covariance of the error state can be represented by:

$$
\begin{aligned}
&\delta \boldsymbol{x}_t = \boldsymbol{F}\delta \boldsymbol{x}_{t-1} \\
&\boldsymbol{\Sigma} = \boldsymbol{F}\boldsymbol{\Sigma}\boldsymbol{F}^T + \boldsymbol{F}_i \boldsymbol{Q}_{imu} \boldsymbol{F}_i^T \\
&\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{F}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{F}_3 & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{F}_1 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{F}_i = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \\
&\boldsymbol{F}_1 = \begin{bmatrix} -\boldsymbol{R}\Delta t \end{bmatrix} \qquad \boldsymbol{F}_2 = \begin{bmatrix} \boldsymbol{I}\Delta t \end{bmatrix} \\
&\boldsymbol{F}_3 = \begin{bmatrix} -[\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b)]_\times \Delta t \end{bmatrix}
\end{aligned} \tag{30}
$$

Here $[\bullet]_\times$ is the skew operator which produces the skew-symmetric matrix. According to the IMU measurement model, the covariance matrix $\boldsymbol{Q}_{imu}$ of the the perturbation input $\boldsymbol{n} = [\boldsymbol{\omega}_n, \boldsymbol{a}_n, \boldsymbol{\omega}_{bn}, \boldsymbol{a}_{bn}]^T$ is:

$$
\boldsymbol{Q}_{imu} = diag(\sigma_\omega^2 \Delta t^2, \sigma_a^2 \Delta t^2, \sigma_{\omega_b}^2 \Delta t, \sigma_{a_b}^2 \Delta t) \tag{31}
$$

4.4. ESKF Innovation Process

In Figure 6, the output of ICP alignment is the orientation $\boldsymbol{q}$ and the position $\boldsymbol{p}$ of the camera in the world frame. It is necessary to transfer the pose to the IMU link through Equation (32):

$$
\boldsymbol{T}_{WI} = \boldsymbol{T}_{WC} \cdot \boldsymbol{T}_{IC}^{-1} \tag{32}
$$

where $\boldsymbol{T}_{WC}$ is the ICP result representing the transformation of Camera in the world frame, $\boldsymbol{T}_{IC} = [\boldsymbol{R}_{IC}|\boldsymbol{t}_{IC}]$ is the installation geometry of camera in the IMU link (see Figure 1) and $\boldsymbol{T}_{WI} = [\boldsymbol{R}_{WI}|\boldsymbol{t}_{WI}]$ is the IMU pose in the world frame. $\boldsymbol{T}_{IC}$ in the current setup is represented by Equation (1). $\boldsymbol{T}_{WI}$ is then used as the input of the filter $\boldsymbol{y} = [\boldsymbol{\theta}, \boldsymbol{p}]^T$. Since the estimate of the error state is zero ($\bar{\delta \boldsymbol{x}} = \boldsymbol{0}$), the innovation $\boldsymbol{z}$ and the covariance matrix $\boldsymbol{\Sigma}$ are :

$$
\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{h}(\boldsymbol{x}) \tag{33}
$$

$$
\boldsymbol{\Sigma} = \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \boldsymbol{Q}_{icp} \tag{34}
$$

where $\boldsymbol{h}(\boldsymbol{x})$ is the measurement model of the system state, $\boldsymbol{H}$ is the jacobian matrix of the measurement model and $\boldsymbol{Q}_{icp}$ is the covariance of ICP. As the orientation and the position are directly measured by ICP, $\boldsymbol{H}$ is the identity matrix accordingly. Therefore, the Kalman gain, the estimate of the error state and the covariance matrix can be innovated.

$$
\boldsymbol{K} \leftarrow \boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{\Sigma}^{-1} \tag{35}
$$

$$
\bar{\delta \boldsymbol{x}} \leftarrow \boldsymbol{K}\boldsymbol{z} \tag{36}
$$

$$
\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma} - \boldsymbol{K}\boldsymbol{Z}\boldsymbol{K}^T \tag{37}
$$

After the innovation process is done, the error state is then injected into the nominal state following the general composition rules defined in Equation (23). The injection process make sure that the nominal state is always updated. Afterward the error state will be reset to zero ($\delta \boldsymbol{x} = 0$) and the orientation part of the covariance matrix need to be updated according to the newest nominal state, as follow:

$$
\boldsymbol{\Sigma}_{\delta\theta} \leftarrow \boldsymbol{G}\boldsymbol{\Sigma}_{\delta\theta}\boldsymbol{G}^T \tag{38}
$$

where $\boldsymbol{G}$ is the Jacobian matrix of the nominal state toward the error state and

$$\boldsymbol{G} = \boldsymbol{I} + [\frac{1}{2}\delta\boldsymbol{\theta}]_{\times} \tag{39}$$

4.5. Integration and Re-Integration

In the VIO system, the IMU data are updated at a high rate (i.e., >200 Hz for a typical MEMS IMU) while the camera capture rate is relatively slow (i.e., <30 Hz). In addition, the ICP processing time is dependent on the numerical convergence rate and is not constant. Therefore, we need to maintain the state queue and update it properly using the integration/re-integration method.

As shown in Figure 8, the above procedure results in a fixed-length state queue being maintained. The integration of the state is driven by the IMU measurement, i.e., whenever an IMU measurement is inputted, the state will be updated and stored in the queue. When the ICP and filter innovation processes are concluded, the state will be updated according to the capture timestamp. Then, the re-integration process integrates the state from the innovation state to the most recent state. As a result, the system state is always updated and the VIO output rate can keep pace with the IMU update rate.

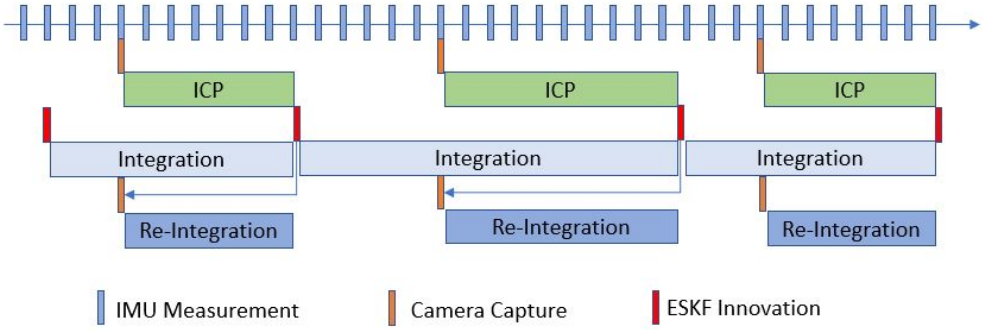


Figure 8. Time synchronization of IMU and ICP.

5. Results and Discussion

Some threshold parameters are relevant to the camera resolution and its detection range (Table 3). Also, the sampling rates of different sensors and camera calibration parameters are listed in Table 4.

Table 3. Device Related Parameter Selection.

Symbol	Value	Description
$\pi_{sh,bg}$	0.01	Background outlier threshold
$\pi_{sh,i}$	100	Intensity gradient threshold
$\pi_{sh,z}$	0.07	Depth gradient threshold

Table 4. Detail of Sensors Information.

Topic Name	Content	Frequency
/image_depth	Depth image	15
/image_nir	8-Bit NIR intensisty image	15
/points	Organized point cloud	15
/camera_info	Camera Information (f_x, f_y, c_x, c_y)	15
/imu	IMU data	250
/gt	Ground truth captured by Vicion	50

Due to there is no existing public ToF-IMU dataset, we first conducted the handheld test and UAV test and compared the results with those of the motion capture system, which serve as the ground truth benchmarks. The accuracy of the odometry is presented by the root mean square error (RMSE) of translational drift. Both the absolute trajectory error (ATE) and the one-step relative pose error (RPE) cases are considered [27]. The definitions of ATE and RPE are shown below:

$$\boldsymbol{E}_{ATE,i} = \boldsymbol{T}_{gt,i}^{-1}\boldsymbol{S}\boldsymbol{T}_{est,i} \tag{40}$$

$$\boldsymbol{E}_{RPE,i} = (\boldsymbol{T}_{gt,i}^{-1}\boldsymbol{T}_{gt,i+1})^{-1}(\boldsymbol{T}_{est,i}^{-1}\boldsymbol{T}_{est,i+1}) \tag{41}$$

$$RMSE(\boldsymbol{E}_{1:n}) := (\frac{1}{n}\sum_{i=1}^{n}\|trans(\boldsymbol{E}_i)\|^2)^{\frac{1}{2}} \tag{42}$$

where $\mathbf{T}_{gt,i}$ is the transformation of ground truth of the frame i, $\mathbf{T}_{est,i}$ the estimate transformation of the frame i and $\mathbf{S}$ the least-squares solution that maps the estimated trajectory $\mathbf{T}_{est,1:n}$ onto the ground truth trajectory $\mathbf{T}_{gt,1:n}$. We then carried out a UAV field test and an exploration test on a ground moving platform. The results were compared with those obtained by RealSense T265 VIO sensors. All data is provided in rosbag and are compatible with the TUM dataset [28].

5.1. Handheld Test

In the handheld test, we held the VIO platform and successively moved in the x, y, and z directions. This generated a good agreement between estimated trajectory and ground truth, as can be observed in Figures 9 and 10. The length of the trajectory is 12.86 m and ATE and RPE of the estimated trajectory is 0.047 m and 0.017 m/s, respectively.

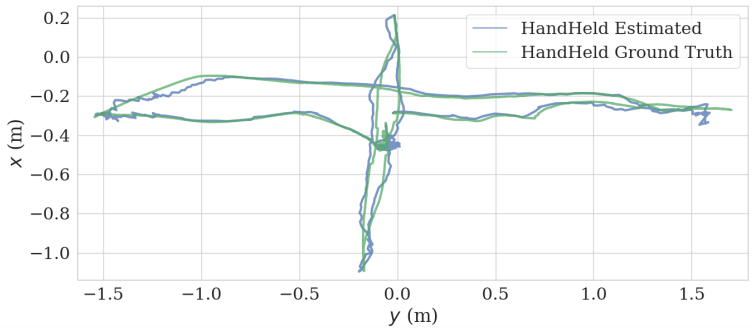


Figure 9. Estimate trajectory and ground truth from x–y plane of hand held test.

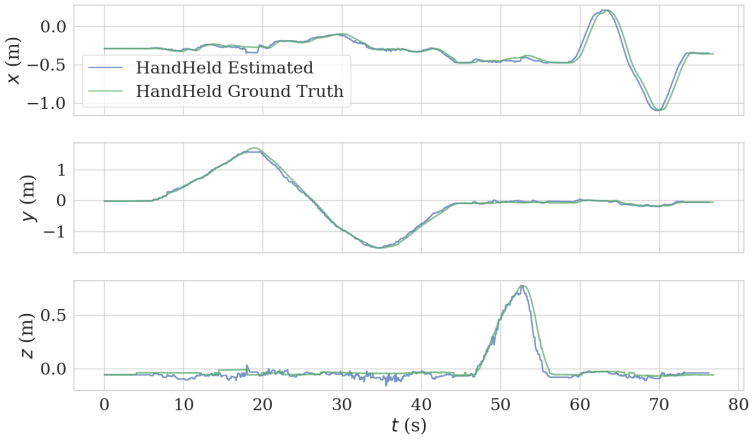


Figure 10. Comparison of estimated trajectory and ground truth of hand held test.

Based on this dataset, Figure 11 shows the comparison of ground truth trajectory with those using algorithms of the conventional ICP (**icp**), the ICP with the salient point selection criteria (**icp+s**), the ICP with the salient point selection criteria and robust weighting factor (**icp+s+w**), the random sub-sampled based ICP (**icp+rd**) and the ICP with the RANSAC pipeline (**icp+r**) [29]. The corresponding accuracy and processing time of each algorithm are also evaluated and listed in Table 5. It can be seen that the conventional ICP workflow can provide the accurate pose estimation but the processing time is unfavorable to the real-time applications. Both the random sub-sampling based ICP and the ICP with our salient point selection method can dramatically reduce processing time. Nevertheless, the former

induced a large ATE error. The ICP with the RANSAC method achieves best in the RPE error but with the largest ATE error and a second largest average processing time. Notably, the conventional ICP, **icp+s**, **icp+r**, and **icp+rd** are all with weights for points set to equal. Together with the robust weighting factor (the weights for points set to the t distribution), the proposed method (**icp+s+w**) in this work reaches almost the same ATE accuracy and 26% improvement in RPE accuracy when compared with the conventional ICP. In the meanwhile, the processing time of icp+s+w is only 25% that of the conventional ICP. The superiority of icp+s+w over other methods is clearly illustrated.

Furthermore, we visualize the salient points in Figure 12a. Our selection algorithm selects the feature and the edge of the object from the raw input point cloud. The number of salient points is consistent (Figure 12b) even if the size of input cloud varies. Together with the good accuracy of icp+s and icp+s+w methods, it is believed that the salient point selection criteria is efficient and robust.

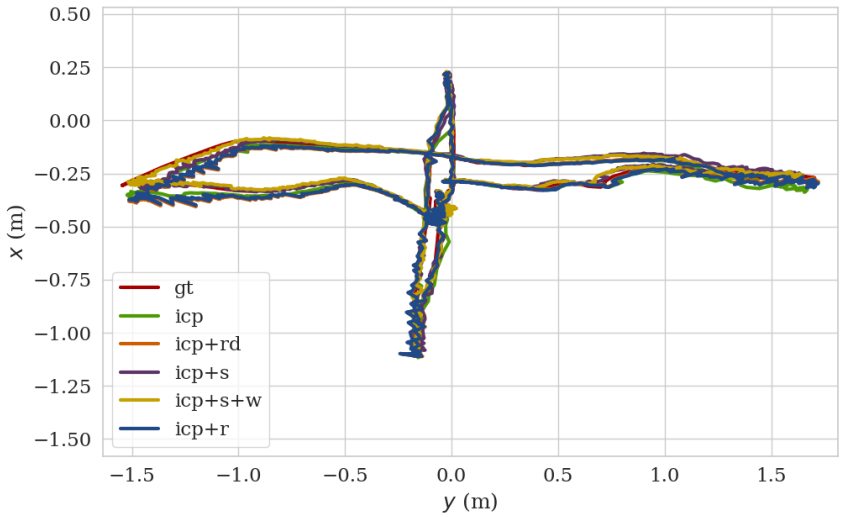
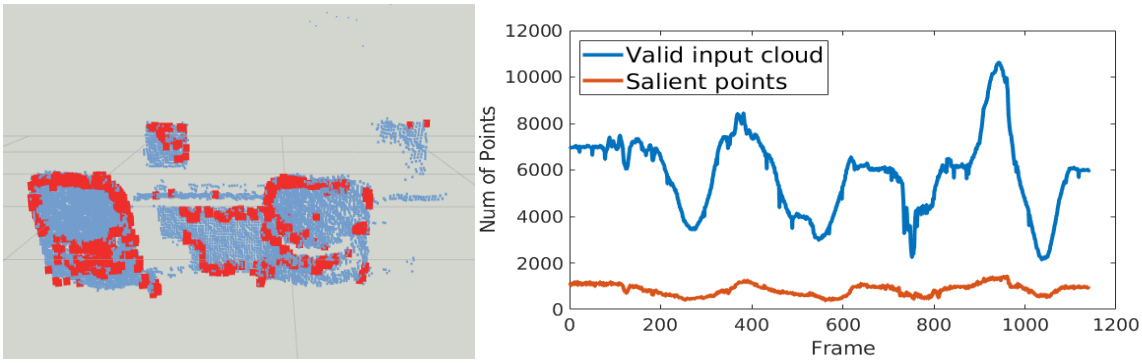


Figure 11. Trajectories of different ICP methods.

Table 5. Accuracy and processing time comparison of different methods.

Method	Translation Error (RMSE)				Average Processing Time (ms)	Relevant Processing Time $\frac{t(x)-t(\text{icp})}{t(\text{icp})}$
	ATE (m)	Improvement $\frac{ATE(x)-ATE(\text{icp})}{ATE(\text{icp})}$	RPE (m/s)	Improvement $\frac{ATE(x)-ATE(\text{icp})}{ATE(\text{icp})}$		
icp	0.048	0	0.023	0	148	100%
icp+s	0.051	–6%	0.019	17%	28.0	18%
icp+s+w	0.047	2%	0.017	26%	36.2	25%
icp+rd	0.063	–31%	0.021	9%	20.4	14%
icp+r	0.070	–45%	0.015	28%	83.8	56%



(**a**) salient points(red) (**b**) number of sailient points and valid input points

Figure 12. Visualization of salient points.

Figure 13 demonstrates that the error of mean $\| p - p^{'} \|$ in each frame (different lines) decreases with the increase of ICP loops. In general, three (no motion) to 15 ICP loops were performed and the motion prediction of IMU provided the initial guess of the ICP algorithm. After about ten loops, the mean $\| p - p^{'} \|$ almost reach the minimum value. The refinement process continues until the corresponding transformation change ($\Delta \boldsymbol{T}$) is relatively small or the maximum loops (15 in our experiment) are reached.

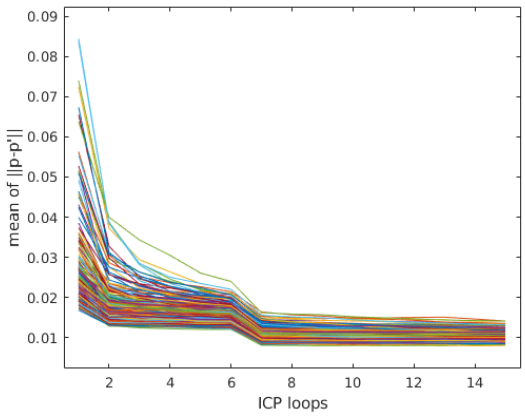


Figure 13. The mean value of $||p - p'||$ in ICP loops.

5.2. UAV Test in LAB Environment

In the UAV test, we mounted the VIO system on our UAV platform and flew the UAV in a circular trajectory (13.019 m) in a laboratory, generating the data in Figures 14 and 15, with an ATE 0.04 m and RPE of 0.021 m/s. Note that we turned off the ambient light during the test, which induced the failure of the mono camera; however, the ToF camera based VIO system continued to function in this dark environment (Figure 16). The full video of this test can be found in Supplementary Materials.

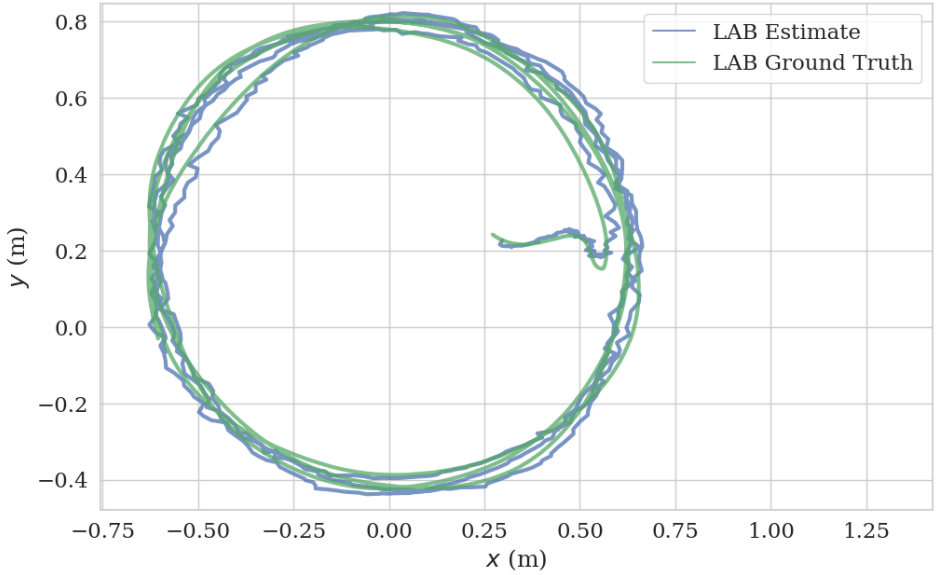


Figure 14. Estimated trajectory and ground truth from an x–y plane of lab environment test.

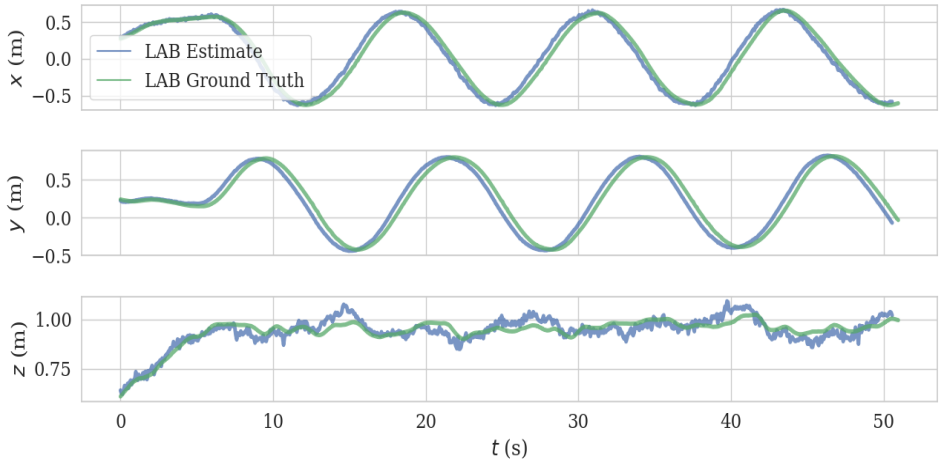


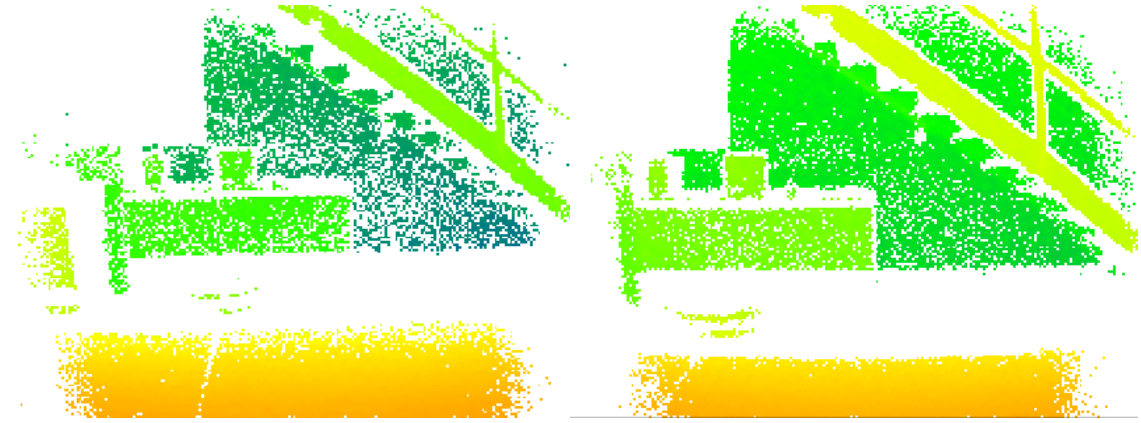
Figure 15. Comparison of estimated trajectory and ground truth of in lab environment test.

(**a**) Lab environment (**b**) Lab environment

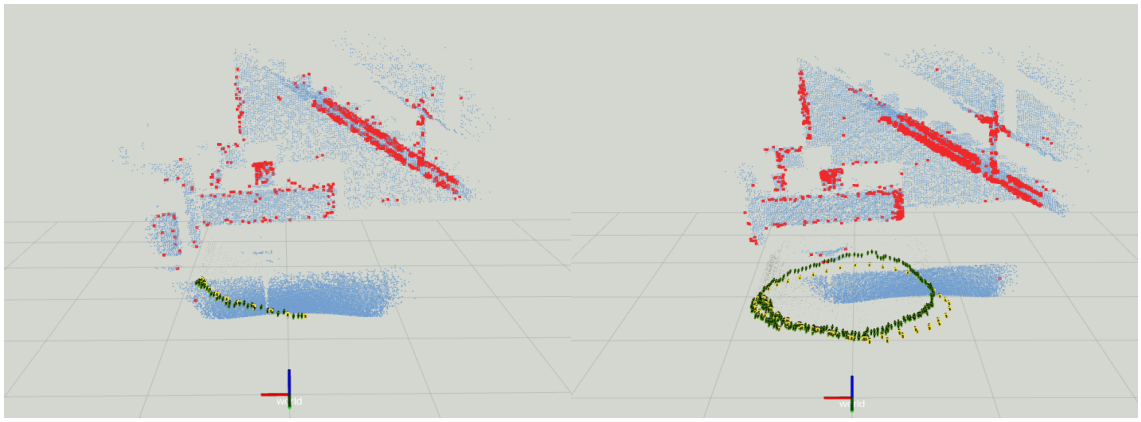
(**c**) Monochrome camera view (**d**) Monochrome camera view

Figure 16. *Cont.*

(**e**) ToF camera intensity image (**f**) ToF camera intensity image



(**g**) ToF camera depth image (**h**) ToF camera depth image



(**i**) Estimated trajectory (**j**) Estimated trajectory

Figure 16. The VIO results obtained in an environment of varying ambient light intensity. (**a**, **c**,**e**,**g**,**i**) are recorded when light is on, (**b**,**d**,**f**,**h**,**i**) are recorded when light is turned off.

5.3. Field Test in Corridor

We mounted the ToF–VIO sensor and a Realsense T265 sensor on the UAV platform and conducted a field test in a corridor. The UAV took off, flew straight along the corridor, and landed 5 m in front of the takeoff position. The trajectory length including takeoff and landing is 8.23 m. Figure 17 presents the comparison between the estimated trajectory generated by the current ToF–VIO sensor and that by the Realsense T265 sensor. In general, the agreement is good, especially in the early period. The ATE

and RPE between ToF–VIO and T265 is 0.12 m and 0.029 m/s, respectively. As the UAV took off and landed at the same ground level, our ToF–VIO has better estimation in the z-direction.

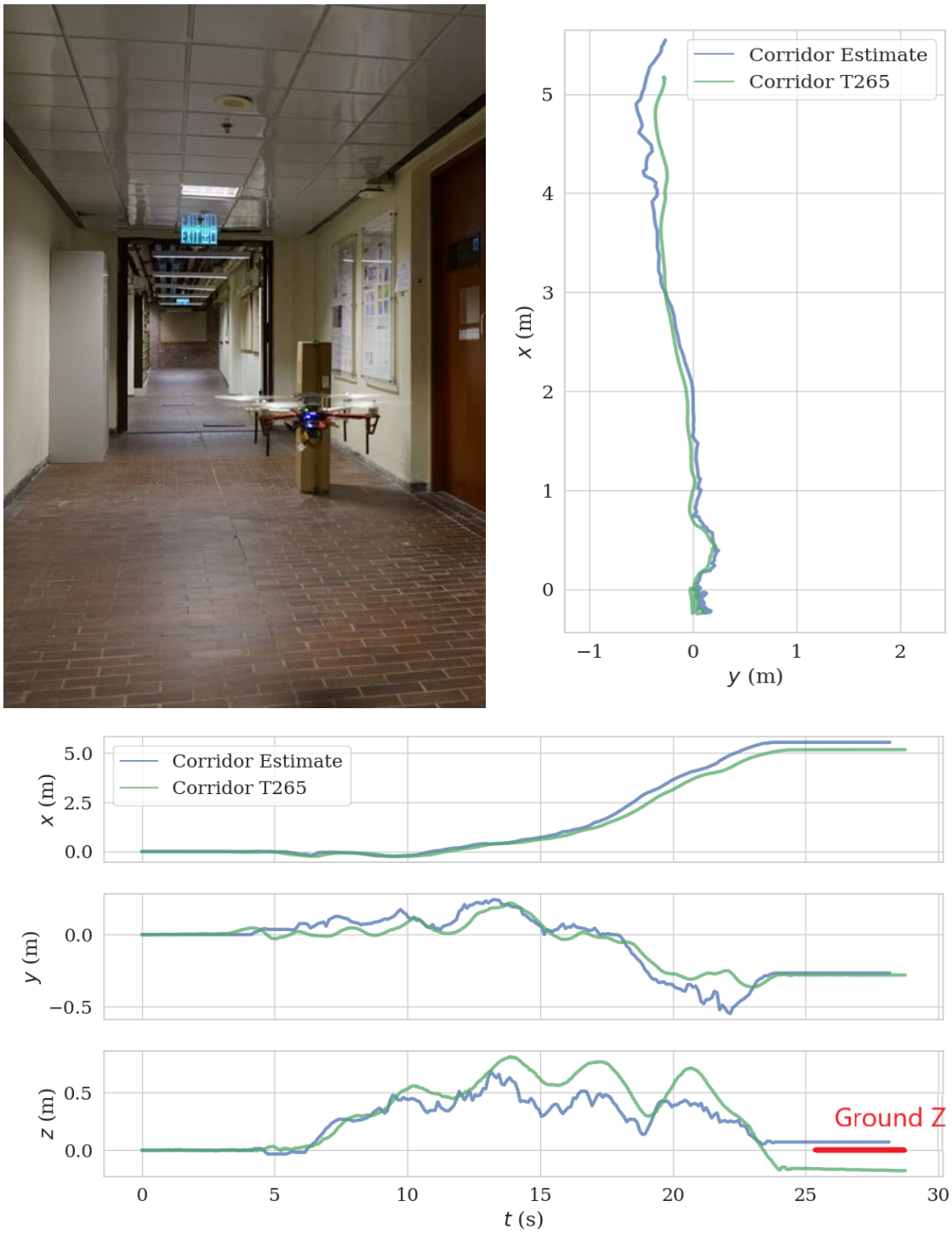


Figure 17. Comparison of the estimated trajectory by the current ToF–VIO sensor with that by the Realsense T265 sensor in the corridor environment.

5.4. Exploration Test Using a Ground Moving Platform

To further demonstrate the performance of our system in a longer range, we mounted the ToF–VIO sensor and a Realsense T265 sensor on a ground moving platform to explore in the indoor Lab environment. The Realsense T265 sensor were used as the benchmark. The trajectory length for

this experiment sequence is 25.675 m (captured by T265 sensor). Figure 18 shows the reconstructed images of the lab and the corresponding trajectories by ToF–VIO and T265. The colors of these points represent the z values (height). As seen, the map shows many detailed features, including a flat ground and straight walls. Figure 19 presents the comparison between the estimated trajectory generated by these two sensors. Good agreement between these two sensors can also be observed. The ATE and RPE between our ToF–VIO and T265 is 0.78 m and 0.025 m/s respectively. Compared with the UAV field test in Section 5.3, the ATE over the trajectory length increase from 1.4% to 3.0%. As known, in an unknown environment without any reference map, the ATE will be accumulated as the drift is not compensated. However, the PRE of these two tests remain at the same level (<0.03 m/s), even when the range increases. The pose estimations from these two sensors agree with each other. The full video of this test can be found in Supplementary Materials.

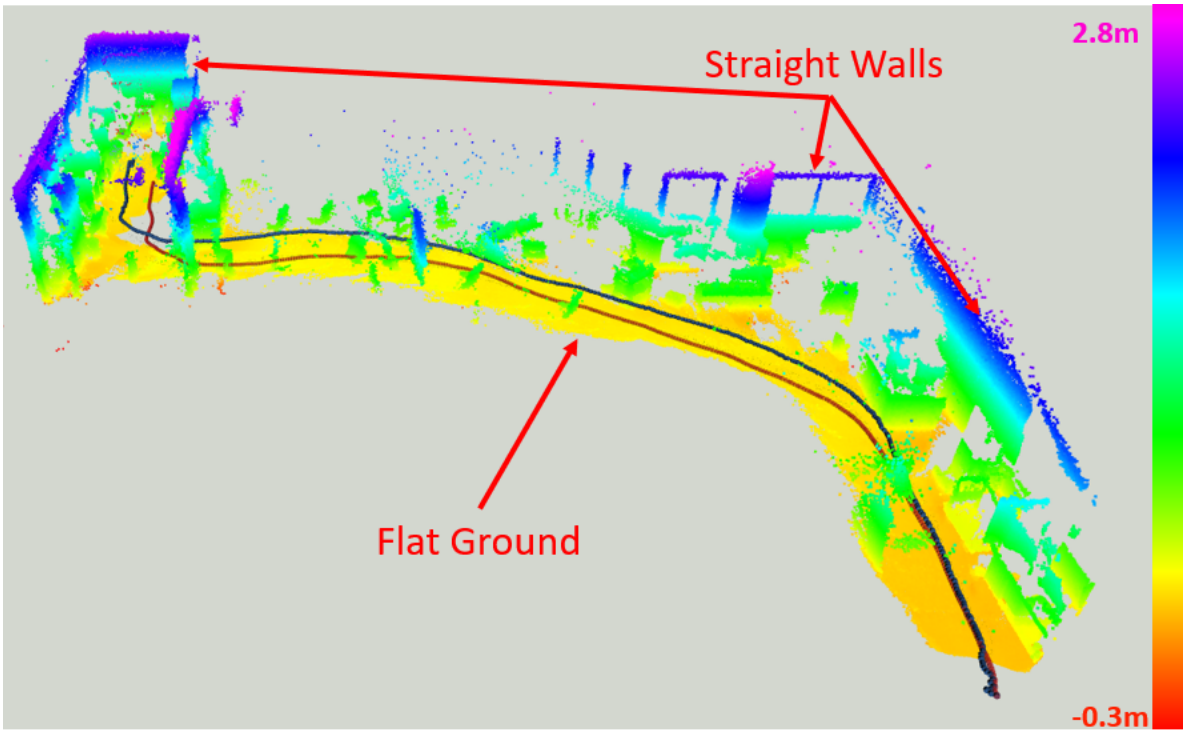


Figure 18. Exploration and restruction of the indoor environment using ToF–VIO, blue path (ToF–VIO), red path (T265).

5.5. Analysis

As shown in Figure 20, the number of salient points is only 5% of the number in the original point clouds. Aside from small variations, the number of salient points remains consistent in different frames. We can therefore efficiently obtain a pose estimation by alignment of these points.

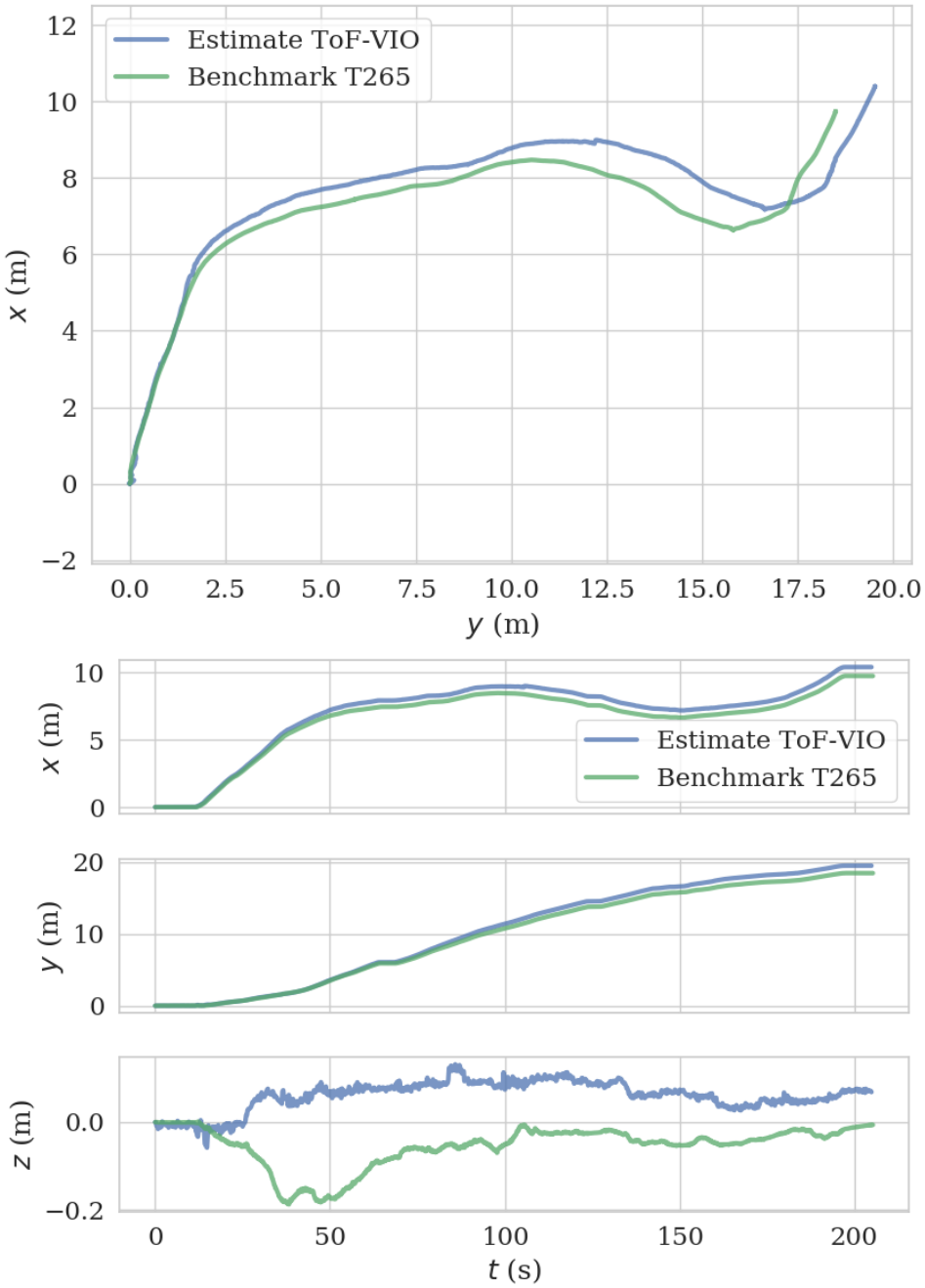


Figure 19. Comparison of the estimated trajectory by the current ToF–VIO sensor with that by the Realsense T265 sensor in the exploration test.

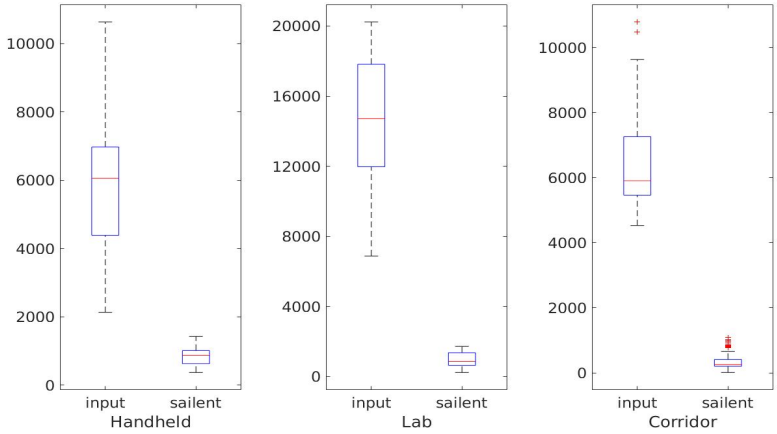


Figure 20. Box plots of the numbers of input points and salient points in three tests.

We tested our algorithm on a TX2-embedded computer and an Intel i5 PC, and the resulting calculation times are listed in Table 6. The results show that our algorithm can estimate the pose at a rate of at least 15 Hz, which means it can be used in realtime applications.

Table 6. Calculation times of different processes of our algorithms on a Intel-i5 PC and TX2 embedded computers.

Process	Time (TX2)	Time (i5-PC)
Salient Point Selection	6 ms	4 ms
ICP alignment	33 ms	30 ms
ESKF	1 ms	1 ms

6. Conclusions

In this paper, we have described the development of a ToF camera-based VIO system. This system was demonstrably superior to the conventional ICP-based workflow, as the computational time was reduced by salient-point selection criteria and the robustness of frame-to-frame alignment was ensured by the statistic weight function. The IMU data were loosely coupled in the proposed system with an ESKF to provide the ego-motion pose estimation. We then assembled our experimental platform and conducted a field test. The results showed that our proposed approach achieved similar accuracy to the state-of-the-art VIO system. Experimental data also showed that our system exhibited excellent performance in an environment of varying ambient-light intensity and in a totally dark environment. The limitation of this study is the range of the ToF camera. The depth detection range of the current ToF camera is 4 m, which is short and will limit the vehicle speed and operation time for the mission. With the development of ToF sensor technology, this limitation will be relieved and the current algorithm can still find good applications.

Supplementary Materials: The following are available online at https://www.youtube.com/watch?v=IqfIqArsWXA, Video: ToF–VIO demonstration. https://www.youtube.com/watch?v=lsl-8b6PmMI, Video: ToF–VIO in exploration. https://github.com/HKPolyU-UAV/ToF--VIO, Souce code and project page.

Author Contributions: Conceptualization, C.-Y.W. and S.C.; methodology, S.C.; software, S.C. and C.-W.C.; validation, C.-W.C. and S.C.; resources, S.C.; writing–original draft preparation, S.C.; writing–review and editing, C.-Y.W. and S.C.; visualization, C.-W.C.; supervision, C.-Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research is support by EMSD HongKong under Grant. DTD/M&V/W0084/S0016/0523—Indoor and Outdoor Inspection of E&M Installations using Unmanned Aerial Vehicles.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of Error State Integration Model

Appendix A.1. Linear Velocity Error

Substituting the error state of the linear velocity into its true state yields

$$\dot{\boldsymbol{v}}_t = \boldsymbol{R}_t(\boldsymbol{a}_m - \boldsymbol{a}_b - \delta\boldsymbol{a}_b - \boldsymbol{a}_n) + \boldsymbol{g} \tag{A1}$$

As the error state is relatively small, the rotation matrix $\boldsymbol{R}_t$ can be approximated by $\boldsymbol{R}_t = (\boldsymbol{I} + [\delta\boldsymbol{\theta}]_\times)\boldsymbol{R}$. According to the error state definition $\dot{\boldsymbol{v}}_t = \dot{\boldsymbol{v}} + \delta\dot{\boldsymbol{v}}$; we have

$$\dot{\boldsymbol{v}} + \delta\dot{\boldsymbol{v}} = (\boldsymbol{I} + [\delta\boldsymbol{\theta}]_\times)\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b - \delta\boldsymbol{a}_b - \boldsymbol{a}_n) + \boldsymbol{g} \tag{A2}$$

Then deducting the nominal state Equation (26) from Equation (A2) results in

$$\delta\dot{\boldsymbol{v}} = \boldsymbol{R}(-\delta\boldsymbol{a}_b - \boldsymbol{a}_n) + [\delta\boldsymbol{\theta}]_\times\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) \tag{A3}$$

Assuming the acceleration noise is a white and isotropic noise ($\boldsymbol{R}(\boldsymbol{a}_n) = \boldsymbol{a}_n$), the velocity error can be solved as

$$\delta\dot{\boldsymbol{v}} = -[\boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b)]_\times\delta\boldsymbol{\theta} - \boldsymbol{R}\delta\boldsymbol{a}_b - \boldsymbol{R}\delta\boldsymbol{a}_n \tag{A4}$$

Appendix A.2. Orientation Error

The derivation of the orientation can be given by the derivation of the general composition form (Equation (23)) and the system kinematics (Equation (24)):

$$\dot{\boldsymbol{q}}_t = \delta\dot{\boldsymbol{q}} \otimes \boldsymbol{q} + \delta\boldsymbol{q} \otimes \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) \tag{A5}$$

Matching the nominal state kinematics (Equation (26)), we have:

$$\delta\dot{\boldsymbol{q}} \otimes \boldsymbol{q} = \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) - \frac{1}{2}\delta\boldsymbol{q} \otimes \boldsymbol{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b) \tag{A6}$$

Consider the composition rule of $\boldsymbol{q}_t$ and $\boldsymbol{\omega}_{bt}$ in Equation (23), Equation (A6) is simplified as follow:

$$\delta\dot{\boldsymbol{q}} \otimes \boldsymbol{q} = \frac{1}{2}\delta\boldsymbol{q} \otimes \boldsymbol{q} \otimes (-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n) \tag{A7}$$

By miltiplying the conjugate $\boldsymbol{q}^*$ on the both sides, we can get

$$\delta\dot{\boldsymbol{q}} = \frac{1}{2}\delta\boldsymbol{q} \otimes (\boldsymbol{R}(-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n)) \tag{A8}$$

with the vector rotation rules $\boldsymbol{q} \otimes \boldsymbol{\omega} \otimes \boldsymbol{q}^* = \boldsymbol{R}\boldsymbol{\omega}$. Further expand Equation (A8) and consider $\delta\dot{\boldsymbol{q}} = \frac{1}{2}\delta\boldsymbol{\theta}$ is solved as:

$$\delta\dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 & -(\boldsymbol{R}(-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n))^T \\ (\boldsymbol{R}(-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n)) & -[(\boldsymbol{R}(-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n))]_\times \end{bmatrix} \begin{bmatrix} 1 \\ \frac{\delta\boldsymbol{\theta}}{2} \end{bmatrix} \tag{A9}$$

Considering the second row, because the error term $((-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n))$ is small, the $\frac{1}{2}(-\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n)\delta\boldsymbol{\theta}$ term can be neglected. Finally, the orientation error is obtained:

$$\delta\dot{\boldsymbol{\theta}} = \boldsymbol{R}(-\delta\boldsymbol{\omega}_{bn} - \boldsymbol{\omega}_n) \tag{A10}$$

References

1. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
2. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE international conference on robotics and automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
3. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [CrossRef]
4. Lynen, S.; Achtelik, M.W.; Weiss, S.; Chli, M.; Siegwart, R. A robust and modular multi-sensor fusion approach applied to mav navigation. In Proceedings of the 2013 IEEE/RSJ international conference on intelligent robots and systems, Tokyo, Japan, 3–7 November 2013; pp. 3923–3929.
5. Mourikis, A.I.; Roumeliotis, S.I. A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
6. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-based visual–inertial slam using nonlinear optimization. In Proceedings of the Robotis Science and Systems (RSS), Berlin, Germany, 24–28 June 2013. [CrossRef]
7. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual–inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
8. Shan, Z.; Li, R.; Schwertfeger, S. RGBD-Inertial Trajectory Estimation and Mapping for Ground Robots. *Sensors* **2019**, *19*, 2251. [CrossRef]
9. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
10. Von Stumberg, L.; Usenko, V.; Cremers, D. Direct sparse visual–inertial odometry using dynamic marginalization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2510–2517.
11. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A.W. Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR* **2011**, *11*, 127–136.
12. Mur-Artal, R.; Tardós, J.D. Visual–inertial monocular SLAM with map reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [CrossRef]
13. Magnusson, M.; Lilienthal, A.; Duckett, T. Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* **2007**, *24*, 803–827. [CrossRef]
14. Zhao, S.; Fang, Z. Direct depth SLAM: Sparse geometric feature enhanced direct depth SLAM system for low-texture environments. *Sensors* **2018**, *18*, 3339. [CrossRef]
15. Kerl, C.; Sturm, J.; Cremers, D. Robust odometry estimation for RGB-D cameras. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3748–3754.
16. Meilland, M.; Comport, A.I.; Rives, P. A spherical robot-centered representation for urban navigation. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 5196–5201.
17. Delmerico, J.; Scaramuzza, D. A benchmark comparison of monocular visual–inertial odometry algorithms for flying robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2502–2509.
18. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
19. Stoyanov, T.; Magnusson, M.; Andreasson, H.; Lilienthal, A.J. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *Int. J. Robot. Res.* **2012**, *31*, 1377–1393. [CrossRef]
20. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]

21. Li, L. *Time-of-Flight Camera–An Introduction*; Technical White Paper; Texas Instruments: Dallas, TX, USA, 2014.
22. Sarbolandi, H.; Lefloch, D.; Kolb, A. Kinect range sensing: Structured-light versus Time-of-Flight Kinect. *Comput. Vis. Image Underst.* **2015**, *139*, 1–20. [CrossRef]
23. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
24. Li, S.; Lee, D. Fast visual odometry using intensity-assisted iterative closest point. *IEEE Robot. Autom. Lett.* **2016**, *1*, 992–999. [CrossRef]
25. Tomono, M. Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 4306–4311.
26. Santamaria-Navarro, A.; Solà, J.; Andrade-Cetto, J. *Visual Guidance of Unmanned Aerial Manipulators*; Springer: New York, NY, USA, 2019.
27. Grupp, M. evo: Python package for the evaluation of odometry and SLAM. Available online: https://github.com/MichaelGrupp/evo (accessed on 25 February 2020).
28. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the International Conference on Intelligent Robot Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.
29. Holz, D.; Ichim, A.E.; Tombari, F.; Rusu, R.B.; Behnke, S. Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robot. Autom. Mag.* **2015**, *22*, 110–124. [CrossRef]