Robotica: (2021) volume 39, pp. 1816–1833. © The Author(s), 2021. Published by Cambridge University Press.
doi:10.1017/S0263574720001526

Accurate 3D Localization Using RGB-TOF Camera and IMU for Industrial Mobile Robots

Majid Yekkehfallah†‡, Ming Yang†‡*, Zhiao Cai†‡, Liang Li† and Chuanxiang Wang†‡

†Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
E-mails: fallahmajid@sjtu.edu.cn, cza1019@sjtu.edu.cn, kdliliang@sjtu.edu.cn, wangcx@sjtu.edu.cn
‡Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

(Accepted December 22, 2020. First published online: February 22, 2021)

SUMMARY
Localization based on visual natural landmarks is one of the state-of-the-art localization methods for automated vehicles that is, however, limited in fast motion and low-texture environments, which can lead to failure. This paper proposes an approach to solve these limitations with an extended Kalman filter (EKF) based on a state estimation algorithm that fuses information from a low-cost MEMS Inertial Measurement Unit and a Time-of-Flight camera. We demonstrate our results in an indoor environment. We show that the proposed approach does not require any global reflective landmark for localization and is fast, accurate, and easy to use with mobile robots.

KEYWORDS: RGB-TOF camera; Industrial mobile robots; Localization; SLAM; Sensor fusion; Extended Kalman filter.

1. Introduction

Automatic industrial mobile robot systems are entirely automated carriage systems that use unmanned robots in indoor areas. Industrial mobile robots securely transport all kinds of goods without human involvement inside manufacturing, logistics, warehouse, and delivery environments. Most of the studies on indoor mobile robot localization have, so far, focused on sensing devices, such as laser range finders,[1] magnet spots,[2] and RFID tags.[3] However, recent developments in low-cost range sensors like RGB Time-of-Flight (TOF) cameras, which included two different data, one is RGB, and another one is depth image data, which uses the TOF technology,[4] make range sensors an attractive substitute for expensive laser scanners. In real-time localization in GPS-denied environments, localization based on environmental features is much more flexible and straightforward than using traditional methods like RFID tags and magnet tape. Using RGB-TOF cameras in real-time localization is very appealing because RGB-TOF cameras have a wide assortment of potential advantages, including the ability to take depth and color images instantaneously at a frequency of upward to 30 fps, which can take advantage of both light detection and ranging (Lidar) and RGB cameras. Compact and solid-state RGB filters afford cameras the many benefits of both laser (depth information) and sight (color image) modalities in an individual package (RGB-D). The RGB part of the camera helps to extract natural landmarks from the environment, which is a big advantage over other methods, such as Lidar-based localization.[1,5,6] Therefore, RGBD cameras have been extensively studied for a new range of applications, such as visual simultaneous localization and mapping

* Corresponding author. E-mail: MingYang@sjtu.edu.cn

(VSLAM).[7] VSLAM uses an environmental measurement model process, which is the estimation of the poses of the camera from its data stream (video and depth) in order to build a map of the whole environment while the camera is moving. The VSLAM system has been applied to solve the indoor data problem from the RGB-D data that use natural landmarks from the environment to build an online map in the indoor environment; however, RGB-D cameras, such as the RGB-TOF cameras, have a restricted field of view (FOV), resulting in lower efficiency in the data collection stage. Concurrently, visual odometry (VO) does not work appropriately in a no-texture area or areas with tedious textures, which are common in indoor environments.[8] Overall, the FOV of a depth camera is smaller than 70 degrees, and the acceptable distance is between 5 and 8 m, which can very easily cause match errors. Due to the limitations of the single RGB-TOF camera, we present the sensor fusion information of an IMU and a RGB-TOF camera based on an EKF state estimation algorithm,[9] which can obtain reliable maps in a real-time probabilistic framework. Localization with a single RGB-TOF camera in a low-texture environment can cause mismatching of the features and failure to estimate the poses. Usually, this problem happens in these situations:

1. The camera's FOV is not vast enough (such as a Lidar that is 70 degrees), making it easy to lose the tracking of pose during the camera's rotation.
2. This case can be worse when the camera turns to a featureless environment.

To achieve the desired frame rate, during the scanning, the camera starts to move quickly, resulting in a lack of features to match between the two frames; also, RGB images can be blurry during fast motion and thus decrease the number of proper matches. These two reasons can decrease the number of the iterations, N, which can lead to failures in the modeling of the parameters via the Random Sample Consensus (RANSAC) algorithm.[10] In Section 3, we will show how this problem influences the whole system of localization using a single RGB-TOF camera localization. This paper deals with a accurate localization using sensor fusion based on an extended Kalman filter (EKF) to overcome the limitation of a single RGB-TOF camera in localization and increase the accuracy of the localization. In this part, we fuse the visual odometry with IMU, which can aid the whole system in less features and fast-motion situations.

The experiment was conducted in two parts in an indoor environment; the first one is using a single RGB-TOF camera. The ground truth data used here is from a centimeter-level accuracy method with SICK Lidar to compare the accuracy of this work second one using a public dataset. In Section 2, the related work is investigated, and the localization using a single RGB-TOF camera is explained in Section 3. Section 4 describes the accurate localization approach. In Section 5, we demonstrate the experimental results of the accurate localization approach. Section 6 focuses on the conclusion of this paper.

2. Related Work

Previous works on indoor mobile robot localization can be divided into two main groups: Lidar-based localization and camera-based localization, but there are some other traditional simple methods like these three methods,[2,3,11] which uses a magnet or RFID as a primary sensor. Basically, mobile industrial robot applications need one of the state-of-the-art methods that can precisely use the indoor environment that is proper for that mobile robot's application. The method[2] uses a magnet as the main sensor for the localization and navigation of the vehicle in the indoor environment; in order to get accurate results, the method uses hall-effect sensors, encoders, and counters as complementary sensors. A fuzzy controller is used as a solution to decrease the wheel-skidding error, which improves the accuracy of localization but makes it necessary to install sensors in the environment for every localization route. Localization based on RFID has been used for a long time in indoor localization for mobile robot applications. The method[11] uses a RFID tag and a laser to regenerate maps with a mobile robot and proposed an algorithm that is used as a variant FastSLAM to recognize the geometric structure of the environment using laser data. It then estimated the pose of the RFID tags based on the trajectory, which was computed by the FastSLAM. The results of that work show that the combination of a laser scanner and RFID technology can help to decrease the computational demand for localization; however, this requires the installation of a huge amount of RFID tags in the

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

environment. Using magnet sensors and magnet tape is another early development that has long been used for localization in mobile robot applications. The method[12] is one of the state-of-the-art methods for the localization and navigation of an unmanned mobile robot. The main contribution of that work is a magnetic nail based on a Kalman filter (KF) and a method to improve localization accuracy by considering the kinematics of the mobile robots on the state equation, angle, and position estimated by a gyroscope and an encoder. The experimental results were compared with two common methods on mobile robots (specifically strip magnets and lasers). The method is cheaper than lasers, but once the work environment changes, the magnet strip route must be changed. Therefore the method is difficult to use in a dynamic application that requires frequent environmental changes. Another article[7] describes a vision-based approach that focused on a natural landmark that was detected by different descriptors. The primary sensor in that work is the RGB-TOF camera, which uses structured light in the infrared spectrum to collect depth information. The highlighted topics in that work are feature matching, feature extraction, and comparing these different methods in terms of accuracy and speed. The method uses natural landmark detectors, which have recently become popular in the field because there is no demand for any artificial landmarks, and they use both RGB and depth information to make a 3D color map. The result of that work is the creation of an accurate map that is like the real environment, but there are some problems with that work, such as the low-texture environment and fast motion, which decrease the accuracy of that method. Another main problem with that kind of vision-based localization is the FOV of the camera, which has limitations and is not wide enough. One way to solve the low-texture and fast-motion scenarios is to use multiple RGB-TOF cameras. The multiple RGB-TOF cameras approach increases the camera FOV and helps to extract more features during the scanning of the environment. One paper[13] uses three RGB-TOF cameras in order to overcome the limitations of using only one RGB-TOF camera. Specifically, the three RGB-TOF cameras are mounted on the rig in different directions to access a larger FOV. The paper considers a new calibration method for multiple cameras, which shows improved mapping efficiency and camera FOV. The method can solve the fast-motion problem, but does not solve the low-texture issue. Another group generated an IMU-aided VO system[14] on aerial vehicles (AVs) that considers the dynamic model of AVs to deal with drift-free velocity and altitude estimations. The IMU and VO were not based on mapping and localization, but almost all such work in the AV (a quad-rotor robot, in that case) focused on the weight of the system and, as a result, the velocity estimation. Using selected algorithms for sensor fusion has a low computational cost and, therefore, should be used on an onboard computer, which is vital in AV applications. A recent Lidar-based paper[15] used 2D or 3D Lidar to provide real-time SLAM and loop closure at 5-cm accuracy via various platforms and sensor configurations. In order to achieve the best real-time performance, that research work used the branch-and-bound method of calculating scan-to-submap matches. In another work,[1] an industrial robot was equipped with a single 2D laser scanner to measure the range of bearing related to shining reflectors used as landmarks, which is a common practice in industrial applications. One of the state-of-the-art methods[16] used 3D Lidar to provide precise localization with a hybrid filtering framework; this method employs Lidar, GPS, and IMU to produce exact localization in an automatically guided vehicle application. It contains two main parts: a hybrid filtering-based method used for localization and a map manager that was proposed for large-scale environments. Another state-of-the-art method,[17] is based on nonlinear optimization; this method is highly accurate and mostly focused on the accurate monocular visual odometry method, which can deal with unknown states. The paper[18] is another robust stereo VIO odometry in the application of autonomous flight; this works uses the multistate constraint Kalman filter to reach the computational efficiency and robustness in this application. This work due to the limitation in processing in the autonomous flight application, they introduced an optimized and fast algorithm, which is most important in this application, which has shown in the experimental result part of this paper. This VIO approach,[19] which uses iterated extended Kalman filter (IEKF) is efficient in reducing the errors that are related to nonlinearities. One of the innovatives in this paper is employed image batches as a landmark, which can help to detect the non-corner landmark as a line; this advantage can aid the feature tracking in fewer features environments that have better performance than classic VIO.[20] However, the experimental results of this work don't prove this method is robust in every environment. This paper[21] presents a tightly integrate VIO in key frame-based visual SLAM. The approach integrated the IMU error and re-projection error to get optimized nonlinear cost function. This work develops both algorithms and hardware to

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

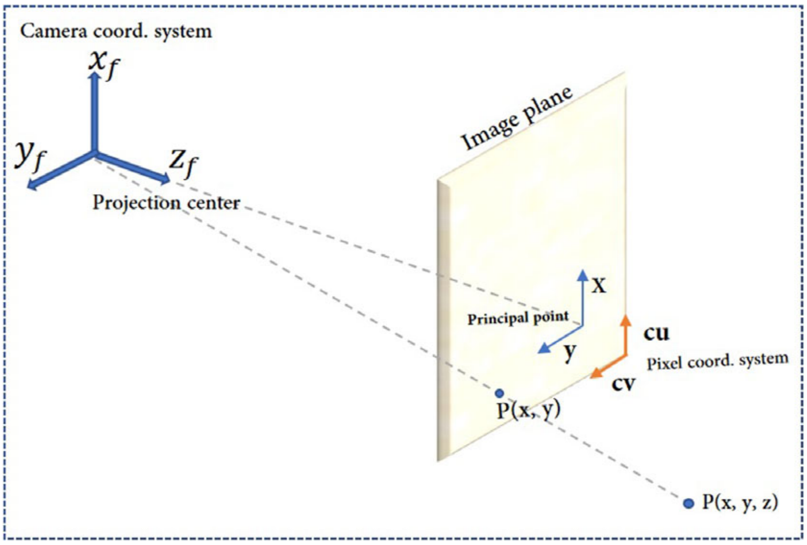

Fig. 1. Two coordinate systems from the RGB-TOF camera and the real world.

achieve accurate real-time VSLAM with robust keypoint matching. But this work is not easy to run on any framework because it also needs to develop the hardware to get the best performance.

3. Localization Based on Natural Landmark Detection

In this section, we explained the principle of localization using the RBG-TOF camera in the indoor environment and relation and projection between the RGB-TOF camera and the real world.

3.1. Localization using RGB-TOF camera

Localization and pose estimation together comprise one of the essential topics for robotics applications like navigation and mapping. In this paper, we propose a localization system that uses a RGB-TOF camera. It gets the advantages of both image and depth data in natural landmark detection from the environment; initially, this localization framework starts to build a sparse feature map of a small area through feature detection. Then, the detected features are tracked and matched, and the system uses images from the RGB-TOF camera to track the pose (position and orientation) of the camera, which is related to the created map. Matching frames from the RGB-TOF video stream with the original map provides a set of geometric constraints that allows us to estimate the camera's position and orientation. The front-end part processes the RGB-TOF camera data and extraction and the matching of BRIEF/GFTT. Refs. [22, 23], which are faster than SURF/SIFT.[24,25] This work chooses fast feature extraction(BRIEF/GFTT) because speed is critical in this application, and this method is faster than other methods.[24,25] The problem with visual descriptors is that they are sensitive to light conditions and less-textured environments, such as white walls, which make it difficult to extract the feature from the images and match two frames. Once a feature is extracted, the system starts to compute a similar transformation with RANSAC.

3.2. 3D projection and 3D transformation

In this part, we begin with the structure of the 3D projection from the world to the camera, and then show how a featureless environment affects this system. In the front-end part, primarily from the RGB-TOF camera's input sequence images, the feature points are detected by BRIEF/GFTT, and then the detected points whose corresponding depth is unreachable are removed. On the other side, those detected points that corresponding depth are reachable then, pixel position of features and its respective raw depth, d are converted into 3D feature points (see Fig. 1). The spatial location of the feature in the pixel coordinates with depth gives:

$$\begin{pmatrix} x_f \\ y_f \\ d \end{pmatrix} \in \mathbb{R}^3, \tag{1}$$

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Algorithm 1: Random Sample Consensus

```
Requirement:
tutorial $dataset \leftarrow D_{ran}$
while $D_{ran}$ do
    Select $k$ data set randomly
    Estimate the corresponding to $k$ point(s) $\leftarrow M_{ran}$
    Find the data items of $D_{ran}$ fit in the model $M_{ran}$ within a tolerance $\leftarrow T_{ran}$
    if $T_{ran}$ is acceptable then
        Quit the process
    end
end
```

which can be converted into 3D Euclidean feature position:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3. \tag{2}$$

The relative transformation between the real world to the camera is given by

$$(c_u, c_v, d) \rightarrow (x, y, z)\,, \tag{3}$$

$$y = \frac{z\left(c_v - y_f\right)}{l_f}, \tag{4}$$

$$x = \frac{z\left(c_u - x_f\right)}{l_f}, \tag{5}$$

$$z = \frac{l_f}{d}, \tag{6}$$

where l_f is the camera focal length and d is distance and unit of that is m, (x_f, y_f) is the center of the image, and the camera calibration gives the baseline between the infrared emitter and the infrared camera (the intrinsic calibration of the RGB-TOF camera). One of the most important algorithms in the transformation section in RGB-D SLAM is the RANSAC algorithm. The RANSAC algorithm is an iterative method to estimate the parameters of a transformation between the matched features. At first, some points are chosen randomly to define a model in the dataset; then, this model is evaluated for the whole dataset. This process is repeated numerous times by selecting new models for particular iterations and keeping the best transformation. This method cannot transfer all points every time, and thus some points are considered failed; those points that this technique can transfer by this algorithm are called inliers, and the others are outliers. The big challenge in the RANSAC algorithm is to find the number of iterations N_{ran}, which can be defined according to the anticipated probability; P_{ran} indicates the probability that at least one point is not in the outlier section, the minimum number of points m, and the likelihood of observing an inlier u_{ran}.[10]

$$o_{ran} = 1 - u_{ran}, \tag{7}$$

o_{ran} here is probability of sensing one outlier in RANSAC algorithm.

$$1 - P_{ran} = \left(1 - u_{ran}^m\right)^{N_{ran}}, \tag{8}$$

where m is a minimum number of points, then we can get the number of iterations:

$$N_{ran} = \left(\frac{log}{log\left(1 - (1 - o_{ran})^m\right)}\,(1 - P_{ran})\right). \tag{9}$$

The Algorithm 1 contains dataset (D_{ran}), at first, this algorithm chooses some points randomly from dataset (D_{ran}) which we call it k , after choosing the k data, then estimate a model (M_{ran})

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Table I. Vector configurations, 0 = false and 1 = true.

EKF	x	y	z	r	p	y	vx	vy	vz	vr	vp	vy	ax	ay	az
VO	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
IMU	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

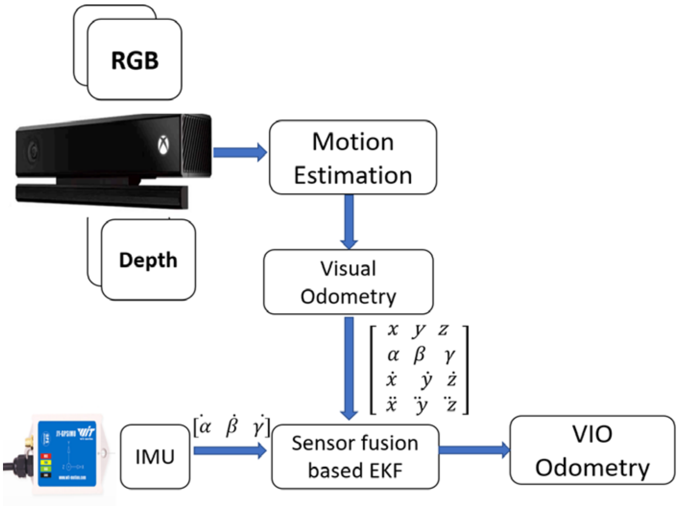
RGB
Depth
Motion Estimation
Visual Odometry
$\begin{bmatrix} x & y & z \\ \alpha & \beta & \gamma \\ \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}$
IMU
$[\dot{\alpha}\ \ \dot{\beta}\ \ \dot{\gamma}]$
Sensor fusion based EKF
VIO Odometry

Fig. 2. Schematic overview of whole localization of our approach.

corresponding with k point which is chosen in the last step. In the next step, check whether the selected points(k) are acceptable for the estimated model within a tolerance (T_{ran}). If the selected points are acceptable, then the process is done and exit the algorithm; otherwise will try different k points with a different model. In a fast-motion scenario, in a low-texture environment, it is easy to mismatch between two frames, and the transformation fails, that is, the system cannot estimate the pose of the last few frames. Therefore, the odometry will be lost and causes significant trouble for the whole system. In order to overcome this problem and make this method accurate in similar situation, we use EKF sensor fusion using a MEMS IMU and a RGB-TOF camera.

4. Accurate Localization Approach

In this section, an IMU-camera sensor fusion system based on an EKF is introduced to overcome the limitations of localization using a single camera that was described in Section 3.

4.1. Extended Kalman Filter (EKF)

The extended Kalman filter (EKF) is a nonlinear mathematical model calculation, which estimates the state of a process based on efficient computational. This nonlinear system can fuse the multiple inputs,[9,26] in our work, there are two inputs for EKF, one of the inputs is visual odometry(VO) that VO's data comes from the calculation of orientation and position of the RGB-TOF camera though a sequence and another one is from IMU. A MEMS IMU is the main module of the inertial supervision systems used in the aerospace and robotics fields, including guided vehicles. The principle of an IMU is based on sensing motion. An IMU is comprised of accelerometers and gyroscopes and works by sensing the present proportion of acceleration, which measures the mobile robot's rotational attributes, including yaw, roll, and pitch.[27] Full 6DOF solutions have become more common due to improvements in low-cost inertial sensing. The fused odometry provided by EKF applies for localization and also influences the quality of the map. In our approach, as shown in Fig. 2 and Table I. Figure 2 is an overview of our approach in which visual-inertial odometry (VIO) is fused data that is provided by our method.

In Table I, r is roll, p is pith, y is yaw, vx , vy, vz are velocity axes, and ax, ay, az are acceleration axes. Table I shows how we configure the inputs sensors for EKF. We use three parameters from the IMU, including roll, pitch, and yaw velocities; other parameters of 6DOF are used by the VO of localization using the RGB-TOF camera. We have tried different configurations from IMU, but due to the noise of this IMU with this configuration, we reached the best performance. The EKF formulation and algorithm are well-known;[28,29] when combining VO with the IMU, this EKF framework can estimate the full 3D (6DOF) pose of a mobile robot over time. The EKF algorithm in this work contains three steps: the EKF model, the prediction, and the update.[9] These equations reflect the nonlinearity of our sensor. These equations, step by step, show the implementation of EKF in three stages which mentioned, time prediction update is given by

$$x_k = f_{k_{-1}}\left(x_{k_{-1}}, w_{k-1}\right), \tag{10}$$

$$y_k = h\left(x_k\right) + v_k. \tag{11}$$

These two equations describe the state of the system we are observing, where x_k here indicates the current step of our system, $x_{k_{-1}}$ is its last state, where w_k is called the noise process. v_k is the current noise associated with a representation action; h is defined as a sensor function. Q_k and R_k are covariance matrices of the noise.

$$\begin{aligned} w_k &\sim (0, Q_k), \\ v_k &\sim (0, R_k). \end{aligned} \tag{12}$$

Our system in this state is not linearized, so we need to linearize them with the use of Taylor series to expand around $w_{k-1} = 0$ and $x_{k-1} = \hat{x}^+_{k-1}$ as below:

$$\begin{aligned} x_k &= f_{k-1}\left(\hat{x}^+_{k-1}, 0\right) + \frac{\partial f_{k-1}}{\partial x_{k-1}}|\hat{x}^+_{k-1} \times \left(x_{k-1}, \hat{x}^+_{k-1}\right) \\ &\quad + \frac{\partial f_{k-1}}{\partial w_{k-1}}|\hat{x}^+_{k-1} \times w_{k-1} - 0 \\ &= f_{k-1}\left(\hat{x}^+_{k-1}, 0\right) + F_{k-1}\left(x_{k-1} - \hat{x}^+_{k-1}\right) + L_{k-1} w_{k-1} \\ &= F_{k-1} x_{k-1} + \tilde{w}_{k-1}, \end{aligned} \tag{13}$$

where $\hat{x}^+_k$ is a posteriori estimate, F_k is Jacobian matrix of f_k, and L_k is Jacobian matrix of w_k that changes both overtime. $\tilde{x}_k$ is error between the true state (x_k) and estimate state $(\hat{x})$ for $\tilde{w}_k$ and $\tilde{v}_k$, respectively, indicates for process noise error and sensor noise error.

To find the $\tilde{w}_{k-1}$, first we obtain the Q as

$$E\left[w_k w_k^T\right] = Q. \tag{14}$$

E is expected value. Then, from Eq. (14), we calculate the $\tilde{w}_{k-1}$:

$$\begin{aligned} E\left[\tilde{w}_k \tilde{w}_k^T\right] &= E\left[L_k w_k w_k^T L_k^T\right] \\ &= L_k E\left[w_k w_k^T\right] L_k^T. \end{aligned} \tag{15}$$

From Eqs. (14) and (15), we can get

$$\hat{w}_k \sim \left(0, L_k Q_k L_k^T\right). \tag{16}$$

In the next step, we should linearize the measurement part around $v_k = 0$ and $x_k = \hat{x}^-_k$ to get

$$\begin{aligned} y_k &= h_k\left(\hat{x}^-_{k-}, 0\right) + \frac{\partial h_{k-1}}{\partial x_k}|\hat{x}^-_k \times \left(x_k - \hat{x}^-_k\right) \\ &\quad + \frac{\partial h_{k-1}}{\partial v_k}|\hat{x}^-_k \times v_k \\ &= h_x\left(\hat{x}^-_k, 0\right) + H_k\left(x_k - \hat{x}^-_k\right) + S_k v_k \\ &= H_k x_k + z_k + \tilde{v}_k, \end{aligned} \tag{17}$$

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

which z_k is our current observation of the system, $\hat{x}_k^-$ illustrates prior estimate, in order to find the $\tilde{v}_k$ we do same as Eqs. (14) and (15):

$$\hat{v}_k \sim \left(0,\, S_k R_k S_k^T\right). \tag{18}$$

H_k is Jacobian of h function, S_k is Jacobian of v function, z_k is known signal. After linearized the state-space system and linearized the measurement, the equation needs to calculate the noise. But before that need to compute the mean of the estimation error, which can be calculated as:

$$\begin{aligned} E\left(\varepsilon_{x,k}\right) &= E\left(x - \hat{x}_k\right) \\ &= \left(I - K_k H_k\right) E\left(\varepsilon_{x,k-1}\right) - K_k E\left(v_k\right). \end{aligned} \tag{19}$$

I is the identity matrix, covariance matrix P and gain G for Kalman flter; next time, we consider the finding of the optimal value of K_k, Since the estimator is unbiased regardless of what value of K_k we use, we must choose some other optimality method to determine K_k. Then we consider the cost function J, but this time, we decided to minimize is the sum of the variances of the estimation errors at time k:

$$\begin{aligned} J_k &= E\left[\left(x_1 - \hat{x}_1\right)^2\right] + E\left[\left(x_n - \hat{x}_n\right)^2\right] \\ &= T_r + P_k. \end{aligned} \tag{20}$$

This time we use the similar way as Eq. (19) to get a recursive formula:

$$\begin{aligned} P_k &= E\left(\varepsilon_{x,k}\varepsilon_{x,k}^T\right) \\ & E\Big[\left(I - K_k H_k\right) \varepsilon_{x,k-1} K_k v_k\Big]\Big[\left(I - K_k H_k\right) \varepsilon_{x,k-1} K_k v_k\Big]^T. \end{aligned} \tag{21}$$

Estimation error is $\varepsilon_{x,k}$, ε_x is vector of x at the time k that is independent from measurement noise v_k, thus:

$$P_k = E\left(\varepsilon_{x,k}\varepsilon_{x,k}^T\right) = E\left(v_k\right) E\left(\varepsilon_{z,k-1}\right) = 0. \tag{22}$$

Once the expected value of both sides is zero, then the above equation becomes:

$$P_k = \left(I - K_k H_k\right) P_{k-1} \left(I - K_k H_k\right)^T + K_k R_k K_k^T. \tag{23}$$

To find the k_k by considering this $\frac{\partial T_r(ADA^T)}{\partial A} = 2AD$ partial derivatives equation if D is symmetric, we can use the Eqs. (20) and (23) with chain rule to obtain:

$$\frac{\partial J_K}{\partial K_k} = 2\left(I - K_k H_k\right) P_{k-1}\left(-H_k^T\right) + 2K_k R_k. \tag{24}$$

To get the value of K_k which provides the minimizes J_k; we set the Eq. (24) to zero then solve the K_k as

$$\begin{aligned} K_K R_k &= \left(I - K_k H_k\right) P_{k_1} H_k^T, \\ K_k &= P_{k-1} H_k^T \left(H_k P_{k-1} H_k^T\right)^{-1}. \end{aligned} \tag{25}$$

Once G is determined, then we need to calculate the covariance matrix P.

First, we define our system like this, which comes from Eq. (10):

$$x_k = f_{k-1} x_{k-1} + w_{k-1}. \tag{26}$$

Then from two sides of Eq. (26), we take the expected value:

$$\begin{aligned} \bar{x}_k &= E\left(x_k\right), \\ &= F_{k-1}\bar{x}_{k-1}. \end{aligned} \tag{27}$$

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Then we use the Eqs. (26) and (27) to prevail:

$$(x_k - \bar{x}_k) \times (x_k - \bar{x}_k)^T = [F_{k-1}(x_{k-1} - \bar{x}_{k-1}) + w_{k-1}] \times [F_{k-1}(x_{k-1} - \bar{x}_{k-1}) + w_{k-1}]^T. \tag{28}$$

In the last step, we calculate the expected value of the above equation, which is equal to P:

$$\begin{aligned} P_k &= E\left[(x_k - \bar{x}_k) \times (x_k - \bar{x}_k)^T\right] \\ &= F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1} \end{aligned} \tag{29}$$

Here, we explain the summary of the EKF algorithm in some simple steps:

- In the beginning, we define our state of system and observation system, as mentioned in Eqs. (10)–(12).
- In the next step, we initialize the filter:

$$\begin{aligned} x_0^+ &= E(x_0) \\ P_0^+ &= E\left[\left(x_0 - \hat{x}_0^+\right)\left(\left[\left(x_0 - \hat{x}_0^+\right)^T\right].\right.\right. \end{aligned} \tag{30}$$

- Then for every k we do as follows:
- Compute the partial derivative matrices in Eq. (13) F_{k-1} and L_{k-1} .
- Next, we perform the time update of the state estimate and estimation error covariance by referring to Eqs. (10), (16), and (29):

$$\begin{aligned} P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T, \\ \hat{x}_k^- &= f_{k_{-1}}(x_{k_{-1}}^+, 0). \end{aligned} \tag{31}$$

- To compute the H_k and S_k partial derivative matrices, we need to refer to the structure of Eq. (17).
- Perform the measurement update of the state estimate and estimation error covariance as

$$\begin{aligned} K_k &= P_{k-1} H_k^T \left(H_k P_{k-1} H_k^T + S_k R_k S_k^T\right)^{-1}, \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k[y_k - h_k\left(\hat{x}_k^-, 0\right)], \\ P_k^+ &= (I - K_k H_k)\, P_k^- \,(I - K_k H_k)^T + K_k R_k K_k^T. \end{aligned} \tag{32}$$

P_k^+ is a posteriori covariance and P_k^- is a prior covariance.

To get translation and rotation between the IMU and RGB-TOF camera, we mount the IMU on the top pf RGB part of the camera, which is zero coordinate of the RGB-TOF camera. Therefore, we could consider the rotation is zero.

4.2. Sensor fusion

This section describes the core of the accurate localization approach and presents the details of the mapping and localization processing. The whole schematic model is as follows:

1. Receive a stream of RGB-D data from the RGB-TOF camera;
2. Present the feature extraction and matching of BRIEF/GFTT;
3. Compute the relative transformations with RANSAC;
4. Compute the initial poses and translate them into g2o nodes and edges;
5. Sensor fusion is based on EKF, which uses the VO and a MEMS IMU as inputs.

Figure 3 shows the whole schematic of our approach, and the key part of this figure is the fusion part, which shows the fusion of visual odometry and how IMU is integrated with RGB-D SLAM. The accurate localization approach takes advantage of the RGB-TOF camera and inertial sensor; the inertial sensor aids the RGB-TOF camera in fast motion and the low-texture environment. The EKF fuses both of these sensors: the single RGB-TOF camera localization provides the VO and the inertial sensor provides the roll, pitch, and yaw angular velocity for the EKF. The combination of these two sensors provides accurate results.

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

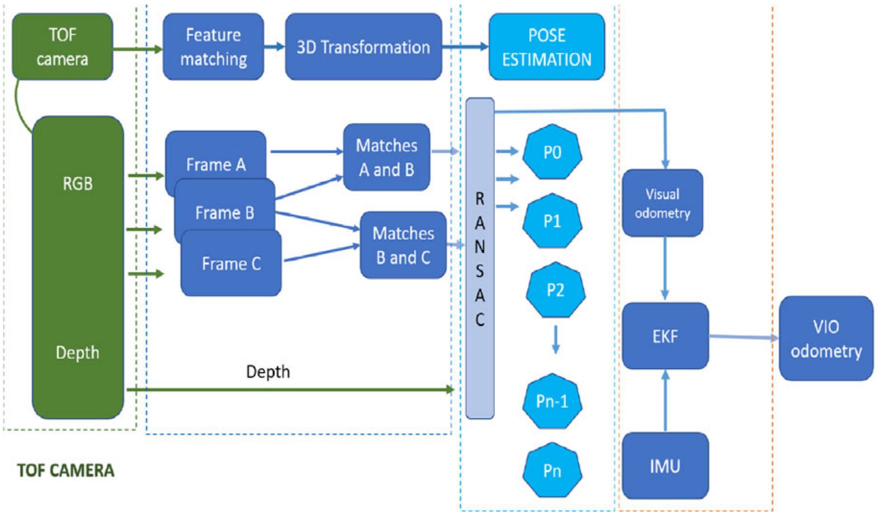

Fig. 3. Schematic overview of whole localization and mapping of our approach.

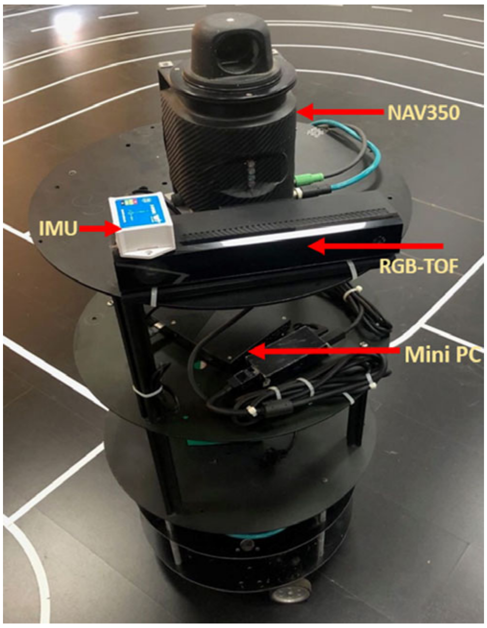

Fig. 4. Mobile robot platform used in the present study.

5. Experimental Results

The experimental results focused on fast motion and textureless for odometry. Therefore, our experimental results focused on these two main parts. The experiment was conducted in an indoor environment where the mobile robots could work. Thereafter, we compared localization using RTAB-Map, which is one of the state-of-the-art configurations that uses only a single RGB-TOF camera,[30] with our approach, which uses both an inertial sensor and a RGB-TOF camera. The experimental platform is a mobile robot (Fig. 4), on top of which an RGB-TOF camera and an IMU are mounted, and there are no other sensors, such as a wheel encoder. We also used the "Sequence-freiburg2-desk" file from the TUM RGB-D dataset.[31] These data are similar to self-collected data, which have been presented in Fig. 4. However, there is some difference in the quality of the RGB-D data and the IMU data. All experiments were run on a laptop equipped with a 2.40 GHz Intel Core i7-3630HQ CPU, 16 GB of RAM, and an NVIDIA GTX 750 Ti GPU with 2 GB of memory. For the first experiment shown in Fig. 6, we use Microsoft Kinect V2. The depth resolution of this camera is 512 × 424 pixel,

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

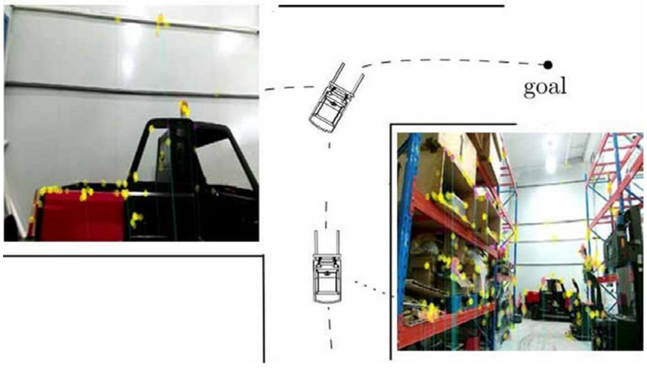

Fig. 5. Mobile robot moving along a planned path around a corner.

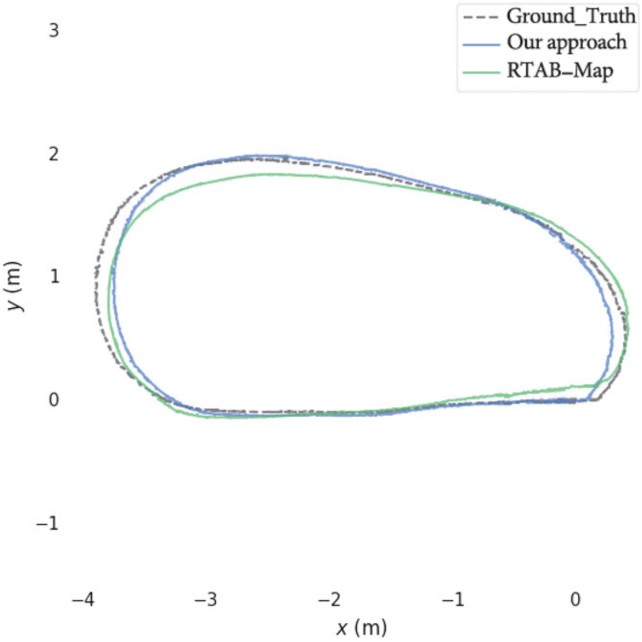

Fig. 6. Localization trajectories in a feature-rich area between the ground truth, our approach, and the RTAB-Map; the units are in meter (m).

and the FOV is 70 × 60 degrees . The ground truth data in this study are obtained using a Lidar sensor with an accuracy of approximately 1 cm, which is mounted on a rotation head; the application of this sensor is especially useful for industrial vehicles in indoor environments. This sensor measures its surroundings as 2D polar coordinates using reflector markers that are mounted in the environment. The drawback of this industrial sensor is that it requires continuous detection of a minimum number of reflector markers to begin operation and during vehicle operation. If, for any reason, it cannot identify the minimum number of reflectors, it will no longer work in the indoor environment. In our approach, by contrast, there is no need to use an artificial landmark, as the sensors installed on the industrial vehicle are sufficient. This is the advantage of using a method based on natural landmark detection compared to the Lidar-based sensor in industrial vehicle applications. However, in our approach, the extraction of features from environments is required.

Figure 5 demonstrates a scenario in which a mobile robot moves along a curved path in the real world. The experimental results reveal that in a low-texture environment, such as white walls, the system extracted fewer features than it did in a feature-rich area. This issue affects the accuracy of the entire framework of localization when a single RGB-TOF camera is used (Fig. 6).

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

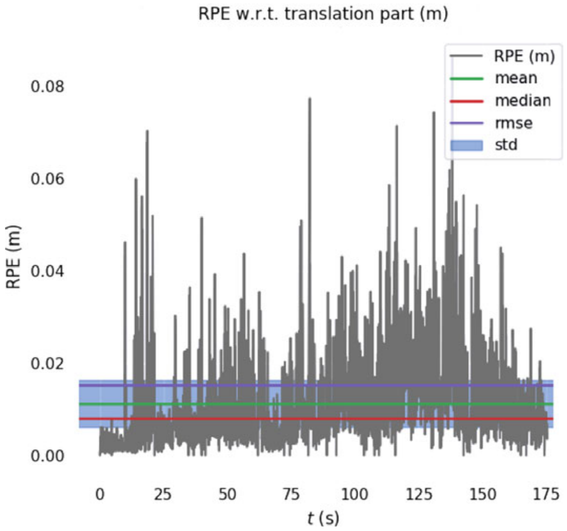

Fig. 7. RPE w.r.t. the translation part (m) For delta = 1 (frames), using consecutive pairs (with SE(3) Umeyama alignment), between the ground truth and RTAB-Map; the units are in meter (m).

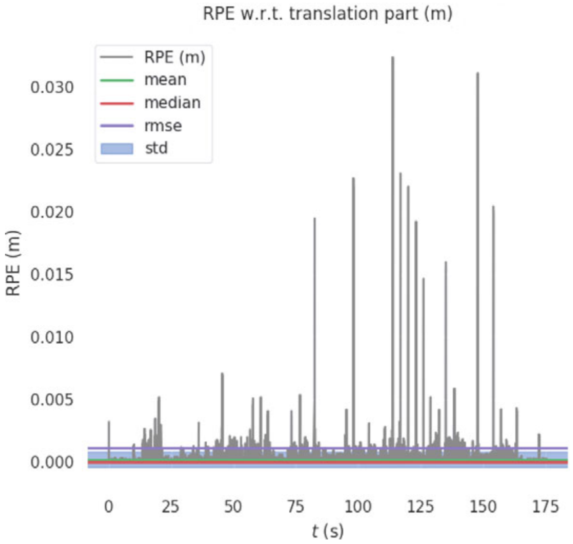

Fig. 8. RPE w.r.t. the translation part (m) for delta = 1 (frames), using consecutive pairs (with SE(3) Umeyama alignment), between the ground truth and our approach; the units are in meter (m).

The results presented in Figs. 6, 7, and 8 indicate that in indoor environments, this approach is accurate enough for unmanned mobile robots for simple navigation applications. This experiment was conducted at a speed of approximately 0.5 m/s, which was chosen because at high speeds, the RTAB-Map cannot work well and, therefore, fails to localize.

The experimental results presented in Fig. 6 are a self-collected dataset. The remainder of the experiment (Figs. 9, 10, and 11) considers a textureless environment with fast motion using the TUM dataset, thereby demonstrating the accuracy of our approach. The TUM dataset is partially similar to our data. The quality of the RGB and depth data is worse than that of the self-collected experiment shown in Fig. 6. Additionally, the IMU data are slightly different, but our algorithms can work for both. Table II presents the localization errors of our approach and the RTAB-Map in an indoor environment. In some scenarios, the error is greater for a curved route than for a straight route because the FOV of the camera is not wide enough, thereby causing mismatch during the rotation, that is, when fewer features are detected, there are more errors in localization. The error in some areas is not very big; however, in full localization, the results demonstrate that this approach can

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Table II. Localization errors of our approach (bold numerical) and the RTAB-Map in an indoor environment using self-collected data.

ERROR	MAX	MEAN	RMS	SSE
OUR APPROACH	**0.034**	**0.00021**	**0.0012**	**0.012**
RTAB-Map	0.086	0.01126	0.0152	0.5648

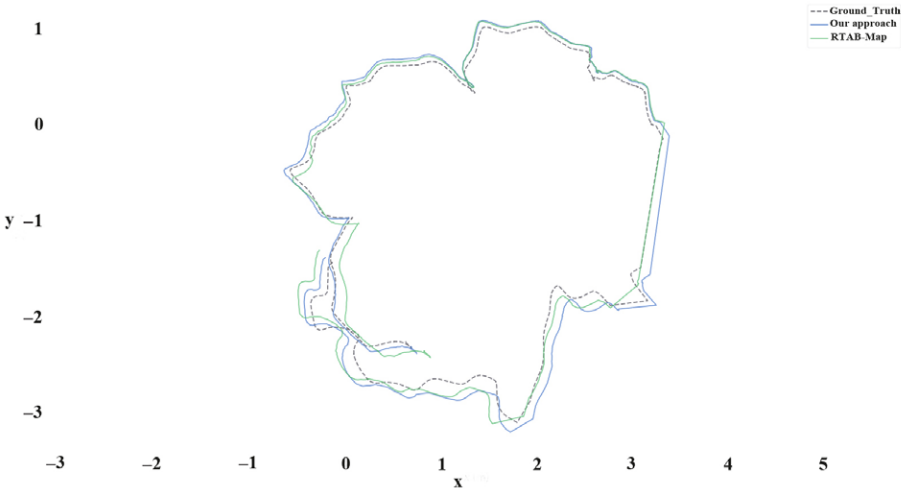

Fig. 9. Localization trajectories in fast motion using the TUM dataset of the ground truth, our approach, and the RTAB-Map; the units are in meter (m).

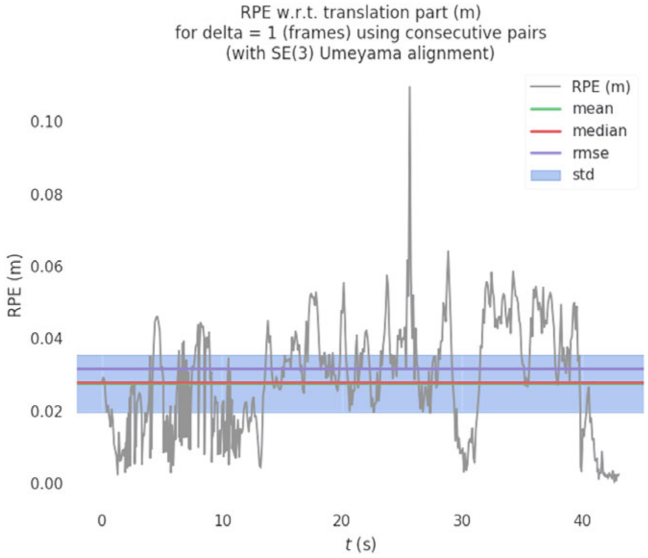

Fig. 10. RPE w.r.t. the translation part (m) for delta = 1 (frames), using consecutive pairs (with SE(3) Umeyama alignment), between the ground truth and the RTAB-Map in fast motion; the units are in meter (m).

improve the accuracy and also boost the accurate in fast-motion condition. The results indicate that our approach is accurate enough in low-texture areas.

In the second part, the experimental results belong to the trajectory with fast motion. In this part, we use the TUM dataset. Figure 9 illustrates the localization trajectories in fast motion using the TUM dataset; these trajectories present the differences between our approach, the RTAB-Map, and the ground truth. Figures 10 and 11 show the related pose error (RPE) for the translation part in fast motion between ground truth and RTAB-Map and our approach, respectively.

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Table III. Localization errors of our approach and RTAB-Map in fast motion using TUM dataset.

ERROR	MAX	MEAN	RMS	SSE
OUR APPROACH	**0.077**	**0.017**	**0.020**	**0.426**
RTAB-Map	0.109	0.027	0.031	0.492

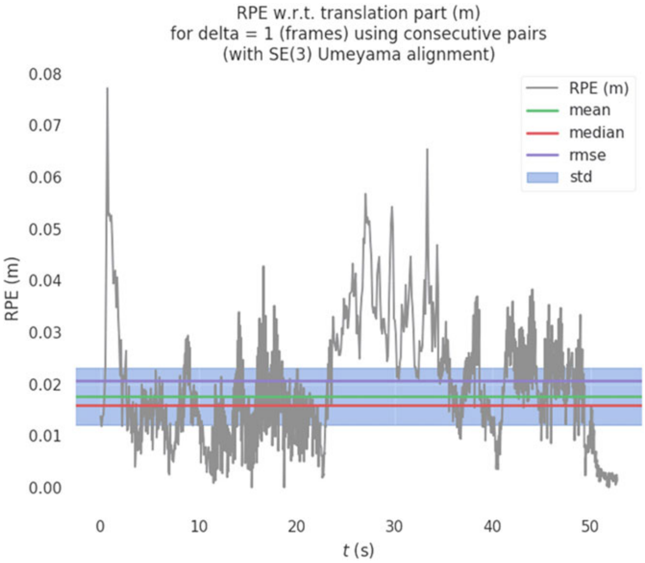

Fig. 11. RPE w.r.t. the translation part (m) For delta = 1 (frames), using consecutive pairs (with SE(3) Umeyama alignment), between the ground truth and our approach in fast motion; the units are in meter (m).

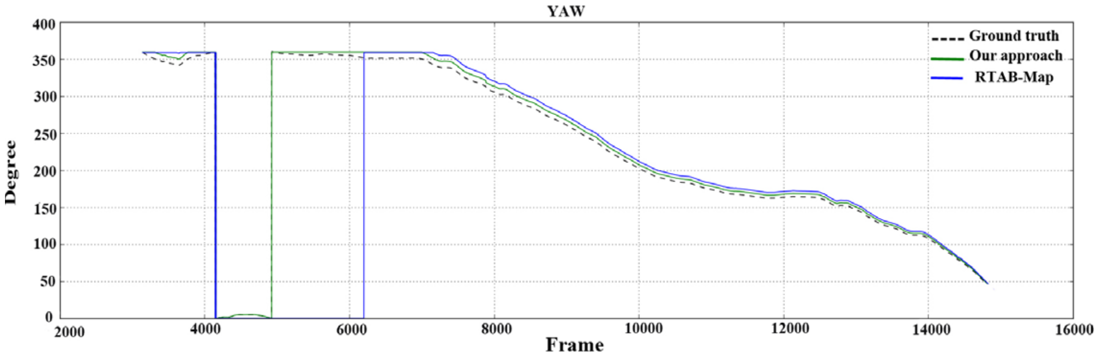

Fig. 12. Comparison of the yaw value for the ground truth (black) and a single RGB-TOF camera (blue) in fast motion with that for our approach (green).

The third part of the experimental results are considered for the trajectory and the RPE error in fast motion. The speed of this part is twice than that of the results in Fig. 6 (1 m/s). Table II lists the localization errors between our approach and RTAB-Map in the indoor environment. Here, we chose fast motion to show the accuracy of our method. However, a comparison of the self-collected data and fast motion in Tables II and III shows that for both methods, the error is greater than that in slow motion (self-collected). One reason for this is that a part of our method uses the data from RGB-TOF, which is not accurate in fast motion.

Figure 12 compares the yaw value between our approach and the RTAB-Map. The main challenge of extracting natural landmarks from the environment is the decreased texture area, making it difficult for the descriptor to extract features and to match the detected features from two related frames. This figure shows another challenge for RTAB-Map methods when the speed gets much higher (higher than Fig. 9 around 1.3 m/s) in a curve or fewer textures area, which can cause mismatching between

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

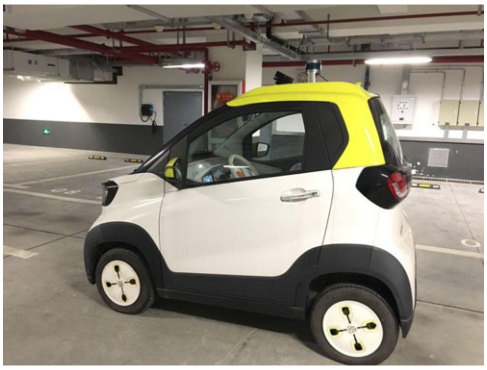

Fig. 13. Industrial vehicle platform used in fourth part of the experiment.

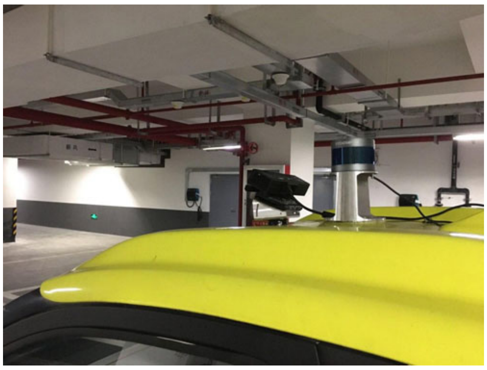

Fig. 14. RGB-TOF camera and Lidar which is mounted on top of the Industrial vehicle platform.

the from and finally visual odometer turn to zero. However, these limitations depend on the environment and the extent of motion, which differ between frames and motion conditions. In the area with white walls, which are common in indoor environments, fewer features were extracted. Moreover, the matching rate between the two frames for low-feature areas is lower than that in the feature-rich area, and this can reduce further for fast motion. This can result in mismatch between two frames. For fast motion, once the single RGB-TOF camera starts moving quickly, the system cannot match two frames, leading to localization failure. This is the main drawback of using a single RGB-TOF camera, that is, it is easy to lose track of landmarks during rotation along curves. As shown in Fig. 12, our approach addresses this problem, making the system sufficiently accurate for situations that a mobile robot can face in an indoor environment. Figure 12 shows that the RTAB-Map exhibits sudden data loss for fast motion; however, our approach is stable. The RTAB-Map loses the matching for a brief period, and once the camera returns to the previous position, the frames are matched again. The IMU is very small and easy to use. Additionally, the IMU has no limitations in terms of changing light and low-texture areas, making our approach sufficiently accurate for low-texture areas and fast motion. When this system cannot match two frames and visuals, the RANSAC algorithm cannot compute the transformation between two frames; therefore, visual odometry returns to zero value. We explained this issue in detail in Section 3.2. In the second part, the experimental results for the trajectory with fast motion are shown. For this, we used the TUM dataset. The fourth part of the experimental results are self-collected data, which use the industrial vehicle platform, as shown in Figs. 13 and 14. Our self-collected experimental results in Figs. 13 and 14 were obtained in an underground parking lot in fast motion (1 m/s). We collected the lidar data with centimeter-level accuracy (model RS-lidar-16)[32] as the ground truth localization data. Figure 15 shows the localization trajectories in fast motion that uses the self-collected data. This trajectory shows the difference between our approach, RTAB-Map, and ground truth. Figures 16 and 17 show the RPE for the translation part in fast motion between ground truth and RTAB-Map and our approach, respectively.

Table IV lists the localization errors between our approach and RTAB-Map in the indoor environment. Here, we chose fast motion to show the accuracy of our method.

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

Table IV. Localization errors of our approach and RTAB-Map in fast motion using self-collected data.

ERROR	MAX	MEAN	RMS	SSE
OUR APPROACH	**2.86**	**0.370**	**0.540**	**0.389**
RTAB-Map	3.961	1.20	1.420	0.760

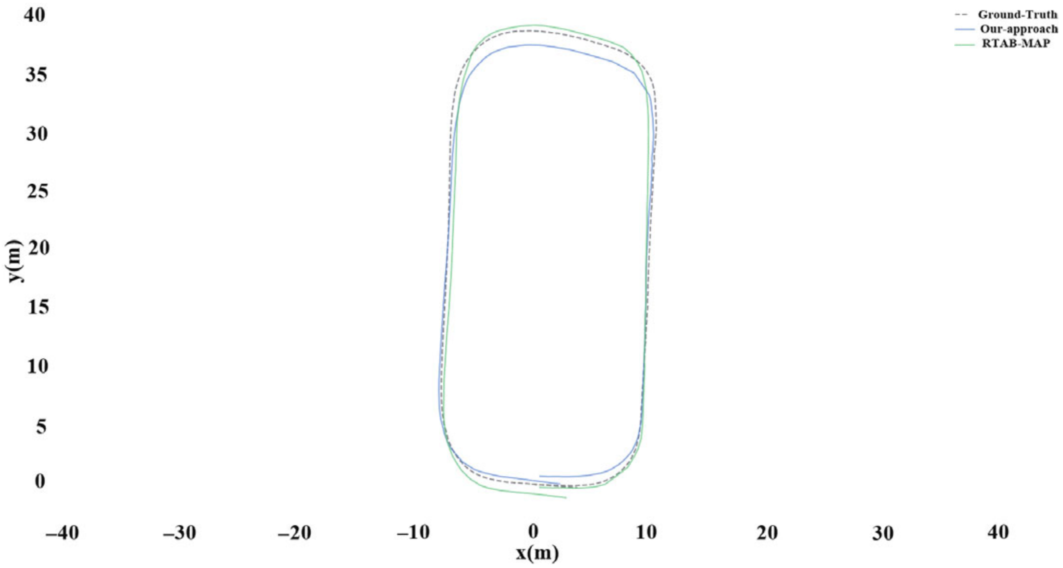

Fig. 15. Localization trajectories in fast motion using self-collected data in large environment for ground truth, our approach, and RTAB-Map. Units are in meter (m).

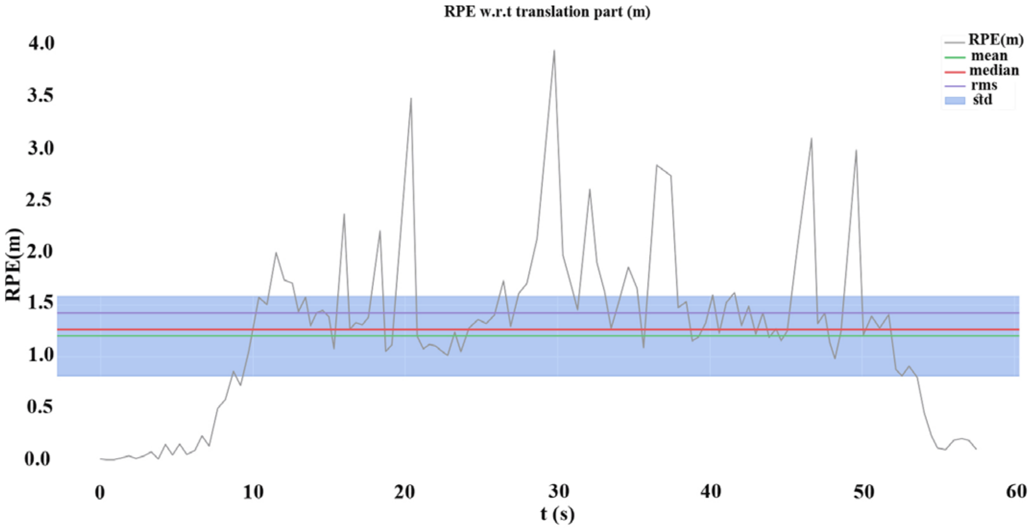

Fig. 16. RPE w.r.t. translation part (m) for delta = 1 (frames) using consecutive pairs (with SE(3) Umeyama alignment), between ground truth and RTAB-Map in fast motion. Units are in meter (m).

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

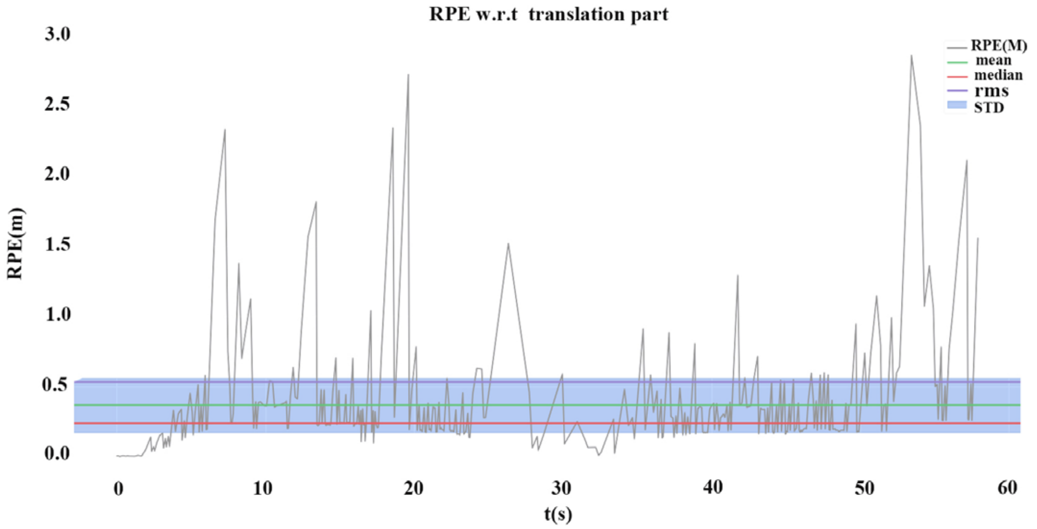

Fig. 17. RPE w.r.t. translation part (m) for delta = 1 (frames) using consecutive pairs (with SE(3) Umeyama alignment), between ground truth and our approach in fast motion, Units are in meter (m).

6. Conclusions

In this paper, we propose an accurate 3D localization method for an industrial vehicle in an indoor environment. The method combines an RGB-TOF camera and a MEMS IMU based on the EKF sensor fusion. In the indoor environment, the localization data of this system can be used in some simple tasks for navigating an industrial vehicle. The method uses only two sensors without any GPS, and it does not require any floor sensors for guidance. The proposed method is accurate for any indoor environment, such as a large warehouse. The IMU and sensor fusion based on EKF, aided by the RGB-TOF camera, overcome the limitation of the single RGB-TOF localization method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (U1764264/61873165), Shanghai Automotive Industry Science and Technology Development Foundation(1807).

References

1. D. Ronzoni, R. Olmi, C. Secchi and C. Fantuzzi, "AGV Global Localization Using Indistinguishable Artificial Landmarks," *2011 IEEE International Conference on Robotics and Automation* (IEEE, 2011).
2. S.-Y. Lee and H.-W. Yang, "Navigation of automated guided vehicles using magnet spot guidance method," *Robot. Comput. Integr. Manuf.* **28**(3), 425–436 (2012).
3. S. Chumkamon, P. Tuvaphanthaphiphat and P. Keeratiwintakorn, "A Blind Navigation System Using RFID for Indoor Environments," *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, vol. 2 (IEEE, 2008).
4. X.Kinectspecs. [Online]. Available: http://www.develop-online.net/news/next-xbox-leak-reveals-kinect-2-specs/0114096.
5. C. Qian, M. Yang, M. Qi, C. Wang and B. Wang, "Swinging single-layer LiDAR based dense point cloud map reconstruction system for large-scale scenes," *Jiqiren/Robot* **41**(4), 464–492 (2019).
6. M. Qi, M. Yang, C. Wang and B. Wang, "3D map reconstruction in indoor environment based on normal distribution transformation under gravity constraint," *Shanghai Jiaotong Daxue Xuebao/J. Shanghai Jiaotong Univ.* **52**(1), 26–32 (2018).
7. P. Henry, M. Krainin, E. Herbst, X. Ren and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Robot. Res.* **31**(5), 647–663 (2012).
8. F. Endres, J. Hess, J. Sturm, D. Cremers and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.* **30**(1), 177–187 (2013).
9. T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," **In:** *Intelligent Autonomous Systems*, vol. 13 (Springer, Cham, 2016) pp. 335–348.
10. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**(6), 381–395 (1981).
11. D. Hahnel, W. Burgard, D. Fox, K. Fishkin and M. Philipose, "Mapping and Localization with RFID Technology," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, vol. 1 (IEEE, 2004).

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press

12. Z. Song, X. Wu, T. Xu, J. Sun, Q. Gao and Y. He, "A New Method of AGV Navigation Based on Kalman Filter and a Magnetic Nail Localization," *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2016).
13. C. Chen, B. S. Yang and S. Song, "Low Cost and Efficient 3D Indoor Mapping Using Multiple Consumer Rgb-D Cameras," **In:** *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 41 (2016).
14. D. Abeywardena, S. Huang, B. Barnes, G. Dissanayake and S. Kodagoda, "Fast, On-Board, Model-Aided Visual-Inertial Odometry System for Quadrotor Micro Aerial Vehicles," *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016).
15. W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-Time Loop Closure in 2D LIDAR SLAM," *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016).
16. L. Li, M. Yang, C. Wang and B. Wang, "Hybrid filtering framework based robust localization for industrial vehicles," *IEEE Trans. Ind. Inform.* **14**(3), 941–950 (2017).
17. T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.* **34**(4), 1004–1020 (2018).
18. K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.* **3**(2), 965–972 (2018).
19. M. Bloesch, M. Burri, S. Omari, M. Hutter and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.* **36**(10), 1053–1072 (2017).
20. Z. Cai, M. Yang, C. Wang and B. Wang, "Monocular Visual-Inertial Odometry Based on Sparse Feature Selection with Adaptive Grid," *2018 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2018) pp. 1842–1847.
21. S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, "Keyframe-based visualâĂŞinertial odometry using nonlinear optimization," *Int. J. Robot. Res.* **34**(3), 314–334 (2015).
22. M. Calonder, V. Lepetit, C. Strecha and P. Fua, "Brief: Binary Robust Independent Elementary Features," *European Conference on Computer Vision* (Springer, Berlin, Heidelberg, 2010).
23. J. Shi, "Good Features to Track," *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 1994).
24. H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding* **110**(3), 346–359 (2008).
25. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
26. D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear approaches* (John Wiley and Sons, Hobokben, NJ, 2006).
27. D. Hazry, M. Rosbi and M. Sofian, "Study of inertial measurement unit sensor," *Proceedings of the International Conference on Man-Machine Systems (ICoMMS), 11–13 October 2009*, (Batu Ferringhi, Penang, Malaysia, 2009).
28. P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 3 (Academic Press, New York, 1982).
29. M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice* (Prentice-Hall, Englewood Cliffs, NJ, 1995/1983).
30. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large scale and long-term online operation," *J. Field Robot.* **36**(2), 416–446 (2019).
31. J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012).
32. https://www.robosense.ai/rslidar/rslidar-16.

https://doi.org/10.1017/S0263574720001526 Published online by Cambridge University Press