Trajectory Drift–Compensated Solution of a Stereo RGB-D Mapping System

Shengjun Tang, Qing Zhu, You Li, Wu Chen, Bo Wu, Renzhong Guo, Xiaoming Li, Chisheng Wang, and Weixi Wang

Abstract

Multiple sensors are commonly used for three-dimensional (3D)-mapping or robotic-vision applications, as they provide a larger field of view and sufficient observations to fulfill frame-registration and map-updating tasks. However, the data sequences generated by multiple sensors can be inconsistent and contain significant time drift. In this paper, we describe the trajectory drift–compensated strategy that we designed to eliminate the influence of time drift between sensors, remove the inconsistency between the sequences from various sensors, and thereby generate a coarse-to-fine procedure for robust camera-tracking based on two-dimensional–3D observations from stereo sensors. We present the mathematical analysis of the iterative optimizations for pose tracking in a stereo red, green, blue plus depth (RGB-D) camera. Finally, complex indoor scenario experiments demonstrate the efficiency of the proposed stereo RGB-D simultaneous localization and mapping solution. The results verify that the proposed stereo RGB-D mapping solution effectively improves the accuracies of both camera-tracking and 3D reconstruction.

Introduction

Recently, the widespread availability of red, green, blue plus depth (RGB-D) sensors, such as Google Tango, Kinect V1, Kinect V2, and Structure Sensor has led to substantial progress in three-dimensional (3D) scanning for indoor mapping, as this sensor equipment is inexpensive, lightweight, and has high-quality 3D-perception capabilities (Endres *et al.* 2012; Mur-Artal and Tardos 2017; Newcombe *et al.* 2011). In effect, this technology can be regarded as a combination of laser and visual systems that enables synchronous, high-speed capture of depth and intensity data. Thus, due to financial constraint and accuracy requirements, RGB-D sensors are the optimal choice for indoor 3D reconstruction.

Many researchers have endeavoured to combine information from single RGB-D sensors (Henry *et al.* 2014; Kerl, Stuckler, and Cremers 2015; Mur-Artal and Tardos 2017; Newcombe *et al.* 2011; Olivier *et al.* 2018; Whelan *et al.* 2015). However, the accuracy and precision of indoor 3D reconstruction with RGB-D devices is highly dependent on the accuracy and robustness of the frame registration and global-optimization processing. Moreover, the frame-matching procedures of single RGB-D mapping systems fail when insufficient features are present in the available fields-of-view of scenes (Chow *et al.* 2014).

One solution to this problem is the use of visual simultaneous location and mapping (SLAM) algorithms, which benefit from a large field of view (Davison, Cid, and Kita 2004). To achieve more robust locating and mapping during visual SLAM, robotics researchers' use of multiple cameras has recently grown, because multiple cameras enable a larger field of view and yield a greater number of observations for frame-registration and map-updating tasks. This implies that the robustness of camera-tracking can be improved by extending the SLAM solution from a monocular camera to multiple cameras (Mazaheri Tehrani 2015).

So far, multiple RGB-D mapping systems of different configurations have been developed. The researchers responsible concluded that accurate calibration and data synchronization of multiple RGB-D cameras are an important prerequisite for such systems (Chen *et al.* 2018; Yang *et al.* 2015; Yong *et al.* 2011). Typically, the calibration of a multiple camera systems is achieved using an optical approach or a geometric approach. The optical approach enables the location of the rigid transformation by minimizing the reprojection error of all correspondences in two-dimensional (2D) space, and the geometric approach obtains the calibration parameters by minimizing the residual error of all 3D correspondences. However, even after using a careful calibration method to reduce the influence of the depth error, the alignment from the global registration is inaccurate due to the inconsistences of distance measurement-error spreading over the depth frames (Deng *et al.* 2014). Furthermore, because synchronization of multiple RGB-D sensors is not practical, significant trajectory drift exists between different sensors.

We address this problem by developing a trajectory drift–compensated (Td-C) solution for stereo RGB-D mapping, which enables the use of observations from multiple views for accurate camera-tracking. Thus, our Td-C model is used to eliminate the inconsistencies of measurements between the data from different sensors. After presenting a literature review on the RGB-D mapping solutions, we introduce a novel camera calibration procedure that incorporates intrinsic calibration for single sensors and a coarse-to-fine boresight calibration for stereo RGB-D sensors. A Td-C model is then presented in detail for accurate synchronization between data streams from different sensors, and a coarse-to-fine multiple camera-tracking method is introduced for map updating tasks. The performance and robustness of the proposed solution is validated using two sets of data sets collected in real scenes. Finally, conclusions and recommendations for future work are presented.

Shengjun Tang, You Li, Renzhong Guo, Xiaoming Li, Chisheng Wang, and Weixi Wang are with the Guangdong Key Laboratory of Urban Informatics & Shenzhen Key Laboratory of Spatial Smart Sensing and Services & Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) & Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, PR China.

Qing Zhu is with the Faculty of Geosciences and Environmental Engineering of Southwest Jiaotong University, Chengdu, P.R. China.

Wu Chen and Bo Wu are with the Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

Corresponding authors: Chisheng Wang and Weixi Wang (sherwoodwang88@gmail.com; wangwx@szu.edu.cn)

Photogrammetric Engineering & Remote Sensing
Vol. 86, No. 6, June 2020, pp. 359–372.
0099-1112/20/359–372

doi: 10.14358/PERS.86.6.359

Related Work

In recent years, a great many 3D dense mapping and visual SLAM solutions based on RGB-D devices have been proposed. State-of-art RGB-D mapping studies usually use a single sensor to gather point clouds, and RGB-D SLAM systems can generally be categorized as dense, sparse, or direct according to their method of frame registration.

The first dense-tracking system, KinectFusion, was designed for single RGB-D modelling and functioned by registering the depth-frame and point-cloud streamed from the sensor into a single global volumetric model. To achieve real-time camera updating, the iterative closest points (ICP) algorithm was used to track the streamed RGB-D frame to a global surface model (Newcombe *et al.* 2011). However, the proposed SLAM system consumed a large amount of computer resources and its working range was limited to volumes of less than 7 m^3. Moreover, the dense tracking system ignored the cumulative drift-error that occurs during processing of frame-by-frame tracking. Subsequently, extensive efforts were made to reduce the computational burden of dense tracking, as exemplified by the development of an improved KinectFusion system (Whelan *et al.* 2012; Whelan *et al.* 2016), a volumetric reconstruction based on a spatial hashing scheme (Nießner *et al.* 2013), and KinectFusion with Octree (Zeng *et al.* 2012). Nowadays, a global optimization method is used for reducing the drift error during SLAM (Dubbelman and Browning 2013; Grisetti *et al.* 2011). In addition, the depth measurements and RGB image sequence can be integrated to enable an extended mapping-range and coverage (Wu *et al.* 2019).

In another advance, sparse, feature-based SLAM systems can be used. As unlike dense RGB-D SLAM systems, the former use few feature-matching points for camera pose updating and mapping tasks. This greatly reduces the computational cost, meaning that the sparse, feature-based system can be used for scene mapping over a larger range. The early feature-based RGB-D SLAM system proposed by Engelhard *et al.* (2011) used speeded-up robust features for feature detection. The 2D feature-matches detected from the adjacent color frames were then mapped to the corresponding depth frames, which transformed the features from 2D to 3D. Then, all 3D matches were used for camera pose estimation and a vertex-edge graph optimization method was used to reduce the trajectory drift during pose-tracking. Extensive efforts were also made to enhance the robustness and accuracy of camera-tracking. These efforts involved investigation of the robustness, accuracy, and time-efficiency of various kinds of feature descriptors and matches (Endres *et al.* 2014; Henry *et al.* 2012; Mur-Artal and Tardos 2017), estimation of the camera motion by integration of different types of features (Kerl *et al.* 2013; Kim, Coltin, and Kim 2018; Le and Kosecka 2017; Shi *et al.* 2018; Tang *et al.* 2018; Zeng *et al.* 2017), and exploration of the uncertainty of depth measurements (Park *et al.* 2012; Tang *et al.* 2019; Vestena *et al.* 2016).

To enhance the tracking performance in textureless regions, the direct sparse odometry (DSO) method was proposed by Alismail, Browning, and Lucey (2016) and Engel, Koltun, and Cremers (2017). The DSO method does not depend on keypoint detectors or descriptors; rather, it can naturally sample pixels from across all image regions that have intensity gradients, including edges or smooth intensity variations on essentially featureless walls. Gao *et al.* (2018) improved the DSO method, developing an extended DSO method with loop-closure handling. Furthermore, Schops, Sattler, and Pollefeys (2019) recently proposed a direct Bundle Adjustment approach to ensure global consistency during RGB-D SLAM; this approach enables simultaneous optimization of poses and geometry, thus limiting the size of the individual optimization problems.

However, the above-mentioned RGB-D mapping system is equipped with a single camera, which means that the camera pose tracking algorithm may easily fail in complex environments due to the very limited field of view of single camera, meaning it fails to identify a sufficient number of visual features (Chen *et al.* 2018). Furthermore, the 3D scenes obtained by a single RGB-D sensor are often incomplete due to occlusion and the sensor's limited scanning range. In another approach, the utilisation of multiple sensors has been a popular option in a variety of mapping applications as it can provide sufficient measurements to fulfill the requirements of frame-registration and map-updating tasks (He and Habib 2018). This means that it is possible to achieve better accuracy and more robust camera-tracking by using a multiple sensor setup. In their early research, Fuhrmann, Langguth, and Goesele (2014) and Pless (2003) constructed a multiple visual camera system for mapping. They presented the theoretical detail of utilisation of multi-camera systems in structure-from-motion studies. Kaess and Dellaert (2006) introduced an eight-camera rig system for better camera-tracking, and described a sparse SLAM approach for real-time reconstruction from multi-camera configurations. Hee Lee, Faundorfer, and Pollefeys (2013) presented a visual ego-motion estimation algorithm for a self-driving car, which was equipped with a multi-camera system. They also introduced a generalized camera model for a multi-camera system by using a two-point random-sample consensus (RANSAC)

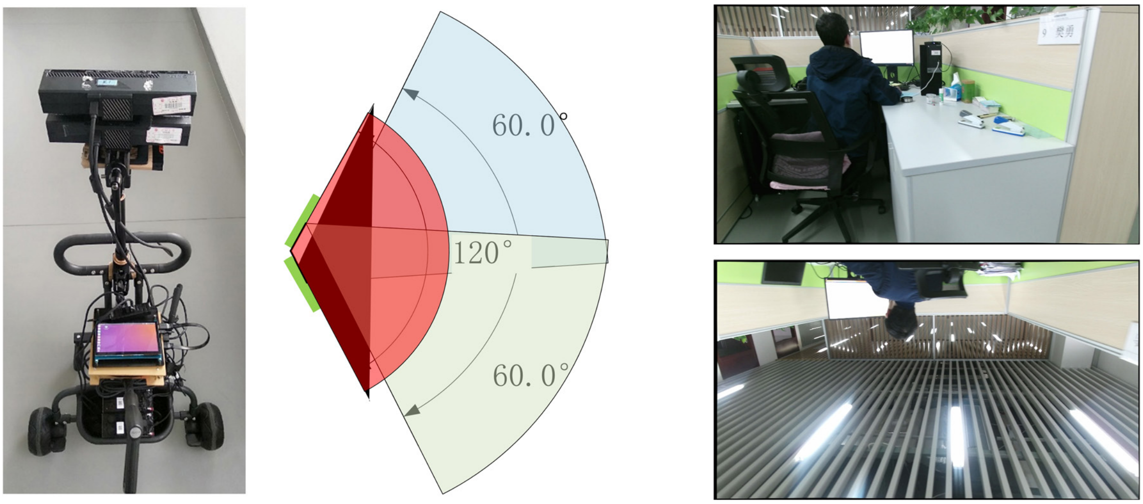


Figure 1.Visual SLAM system using two Kinects mounted on a trolley. Left: Stereo camera SLAM system using two Kinect V2 mounted on an NVIDIA Jetson TX2. Middle: View angle of the stereo RGB-D sensor. Right: Downward camera view and upward camera view at the same stamp.

scheme. Based on the parallel tracking and mapping system (Klein and Murray 2007), a stereo-camera visual SLAM system was proposed by Yang, Scherer, and Zell (2014), in which the iterative optimizations for pose tracking and map refinement that use the observations from stereo cameras were detailed and verified. The results of their experiments implied that their proposed system was more resistant to tracking failure than a monocular method. Furthermore, Yang, Scherer, and Zell (2016) presented a more robust SLAM solution for modular autonomous vehicle systems (MAVs) based on a dual-camera system, in which they used the integrated loop-closure detection and global optimization processes to achieve better tracking accuracy. The loop-closing method is especially important in multiple camera SLAM, and Lee, Fraundorfer, and Pollefeys (2013) introduced a structureless pose-graph loop-closure framework in which the relative pose was obtained from the epipolar geometry of the multiple camera system.

To the best of the authors' knowledge, multiple RGB-D mapping systems have rarely been investigated. Chow *et al.* (2014) constructed a hybrid mobile-mapping system with an inertial measurement unit , two Kinect sensors, and a laser scanner. However, instead of tracking with the observations from multiple views, a point-to-plane ICP algorithm was used for tracking each Kinect pose individually, and then integrated into an implicit iterative-extended Kalman filter. Yang *et al.* (2015) introduced a stereo RGB-D SLAM system, which involved all observations detected from the adjacent frames being streamed from multiple cameras for camera-tracking. They compared the results from a single-sensor and dual-Kinect system, and found that the latter provided better pose-tracking performance and achieve higher mapping-accuracy. However, there are two problems with the Yang *et al.* (2015) approach. First, the potential tracking error in SLAM was not considered in the external calibration procedure for stereo sensors, and thus loop-closure detection was not implemented. Second, the system ignored the significant time drift of data streamed from different sensors, which may result in inaccurate camera-tracking. Chen *et al.* (2018) introduced a triple RGB-D system mounted horizontally on a rig. In this system, sensors were driven with the opensource frame OpenKinect. However, instead of a SLAM framework, they concentrated on calibrations of single and multiple sensors, and verified the effectiveness of mapping using multiple RGB-D cameras. However, the external calibration in this work was achieved with a global rigid transformation by minimizing the residual error of all correspondences, which ignored the inconsistences in the accuracy of correspondences. Thus far, several multiple RGB-D mapping systems have been developed and introduced, and it has been found that accurate calibration and data synchronization of multiple RGB-D cameras are a prerequisite for these systems (Chen *et al.* 2018; Yang *et al.* 2015; Yong *et al.* 2011).

However, as mentioned above, existing multiple RGB-D mapping systems achieve extrinsic calibration based on the traditional chessboard, which may generate inaccurate registration due to the inconsistences of distance measurement-error spreading over the depth frame. Meanwhile, the manipulation of data synchronization for data sequences from different sensors has not been addressed. Therefore, this study focuses on these problems and presents a Td-C solution for stereo RGB-D mapping.

Our work is innovative for two reasons. First, in consideration of the influences of depth errors on external calibration results, a careful calibration procedure is presented in detail. Second, a Td-C model specifically designed for data synchronization between multi-sensors is incorporated into a coarse-to-fine multiple camera-tracking procedure.

Coarse-to-Fine Stereo RGB-D Camera Tracking

Overview of Approach

Herein we present a Td-C solution for stereo RGB-D SLAM that eliminates the influence of time drift between cameras during motion-tracking. Figure 2 shows the framework of our Td-C solution for stereo RGB-D mapping, which consists of a calibration, a front-end, and a back-end. The calibration work is separated into two parts: calibration of a single RGB-D sensor, and calibration of the stereo-RGB-D sensors. First, the camera parameters for the single sensor are obtained with a standard camera calibration process. Second, we use a coarse-to-fine calibration scheme to calibrate the stereo-RGB-D sensors, solve the initial exterior orientation parameters (EoPs) from sparse control markers, and further refine the initial value by an ICP variant that minimizes the distance between the RGB-D point clouds from the reference and the slave sensors.

Although a fixed rigid transformation should in theory be sufficient to register the frames with the same time stamp from

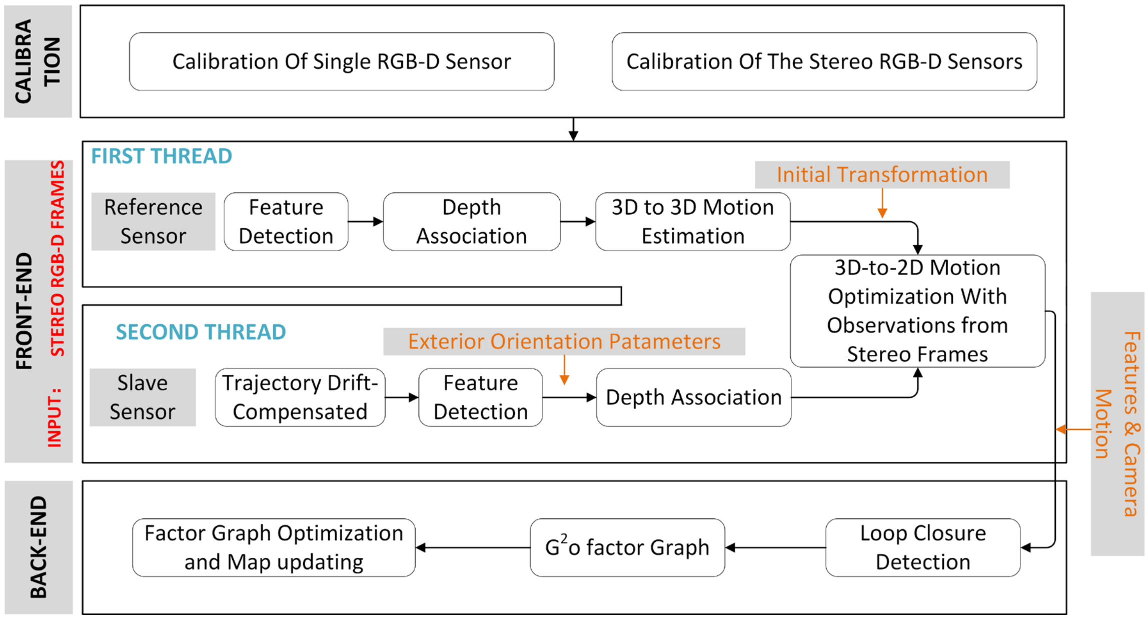

CALIBRATION
Calibration Of Single RGB-D Sensor
Calibration Of The Stereo RGB-D Sensors
FRONT-END
INPUT: STEREO RGB-D FRAMES
FIRST THREAD
Reference Sensor
Feature Detection
Depth Association
3D to 3D Motion Estimation
Initial Transformation
3D-to-2D Motion Optimization With Observations from Stereo Frames
SECOND THREAD
Exterior Orientation Patameters
Slave Sensor
Trajectory Drift-Compensated
Feature Detection
Depth Association
Features & Camera Motion
BACK-END
Factor Graph Optimization and Map updating
G2o factor Graph
Loop Closure Detection

Figure 2. Framework of –Td-C solution for stereo RGB-D mapping (where g2o stands for general (hyper)graph optimization).

two sensors, multiple Kinect sensors cannot be synchronized due to hardware limitations. This results in significant time drift between the published RGB-D streams of different sensors, which may cause inaccurate registration. To compensate for this time drift between data streams from different sensor and enable the use of observations from stereo sensors, we separate the stereo RGB-D tracking into two threads. In the first thread, reference camera pose tracking is conducted, and the 2D and 3D feature matches detected with adjacent RGB-D frames are used for pose-recovery. The second thread involves first integrating the trajectory drift–compensated strategy to avoid inconsistency between the streams from different sensors.

In particular, in this approach the frame streamed from the reference sensor is defined as the "reference frame" and the frame from the slave sensor is defined as the "slave frame". To facilitate the use of stereo RGB-D tracking, we collectively define the frames captured with the same time stamp or a minimum time-difference in different sensors as a "bundle frame", which is to be used for motion optimization by integrating all observations from stereo views. Therefore, to reduce the influence of time drift in bundle frames, a trajectory-compensated strategy is applied to translation, and rotation is introduced to recover an accurate relationship between the slave frame and the reference frame. It should be noted that the drift of the translation and rotation of each "bundle frame" depends on the time-difference and movement speed of the system. After this, the new compensated keyframes from the slave sensor are integrated for pose refinement and used to create new map points. Finally, experiments in complex indoor scenarios demonstrate the efficiency of our proposed multiple RGB-D SLAM algorithm.

RGB-D Sensor Calibration

The calibration procedure is divided into two threads. The first thread handles the intrinsic calibration of the RGB and depth cameras' geometric parameters, namely focal length, principal point, and distortion parameters, and calibrates the RGB-D baseline. The core concept of intrinsic calibration of a single sensor is based on the pinhole camera model, which represents the relationship between the 2D image-point and the corresponding 3D ground point as a function of the camera's internal and external parameters.

The second thread deals with the calibration for EoPs, which enables precise registration of the point cloud from different sensors. In this work, we derive the accurate EoPs by minimizing the residual errors of 3D correspondences; the 3D cone-markers shown in Figure 3a and 3b are used for calibration purposes to ensure the consistent measurement accuracy of correspondences. The feature matches are detected from RGB images by a scale-invariant feature transform (SIFT) operator (Lowe 2004). The corresponding 3D point pairs are obtained by mapping feature-matches to depth images, in which $P^r$and P^s represent the peak points of 3D cones in the reference sensor and the slave sensor, respectively. Using RANSAC and the least-squares method, the optimal rigid transformation T_s^r between the downward and upward cameras can initially be calculated by minimizing the cost function according to Equation 1 below:

$$T_s^r = \underset{T}{\operatorname{argmin}}\left(\frac{1}{|A|}\sum_{i\in A} w_i \left|T\left(P_r^i\right) - P_s^i\right|^2\right) \tag{1}$$

Here, $P_r^i \cong T_s^r \cdot P_s^i$, where T_s^r consists of a rotation matrix R and a translation t, A contains the associations between feature points of the frames from two sensors, and w_i is the weight for each point based on the theoretical error-of-depth measurement (Khoshelham and Elberink 2012). After that, we further refine the EoPs T_s^r with an ICP variant by minimizing the distance between the point cloud from two sensors. As shown in Figure 3c and 3d, the point clouds from the reference sensor and the slave sensor are captured at the same time and can thus be registered with high precision. Quantitatively, the recovered external parameter provides a 0.006-m root-mean-square error (RMSE).

Trajectory Drift–Compensated (Td-C) Approach

During the stereo RGB-D mapping, two sets of RGB-D data sets are streamed, and each frame is labeled with its corresponding time stamp. To facilitate the use of stereo RGB-D tracking, we collectively define the frames captured with the same time stamp or minimum time difference in different sensors as a "bundle frame", which is to be used for motion optimization by integrating all observations from stereo views. Although a fixed rigid transformation obtained by the external calibration method in the section "RGB-D Sensor Calibration" should in theory be sufficient to register the frames in a bundle frame, multiple Kinect sensors cannot be synchronized, and a significant time drift can thus be seen in the frames of a bundle frame due to the unstable topic-publishing rate of sensors. As shown in Figure 4, this time drift in each bundle frame is plotted together with the sensor trajectory, in which the RGB-D data sets are streamed at 5 Hz. As measured, for each bundle frame there is an average 0.03 s time drift between the frames captured by different sensors.

As demonstrated in Figure 4, the sensors are synchronized with the network time protocol time service, which enables millisecond-level synchro error. This means that the starting

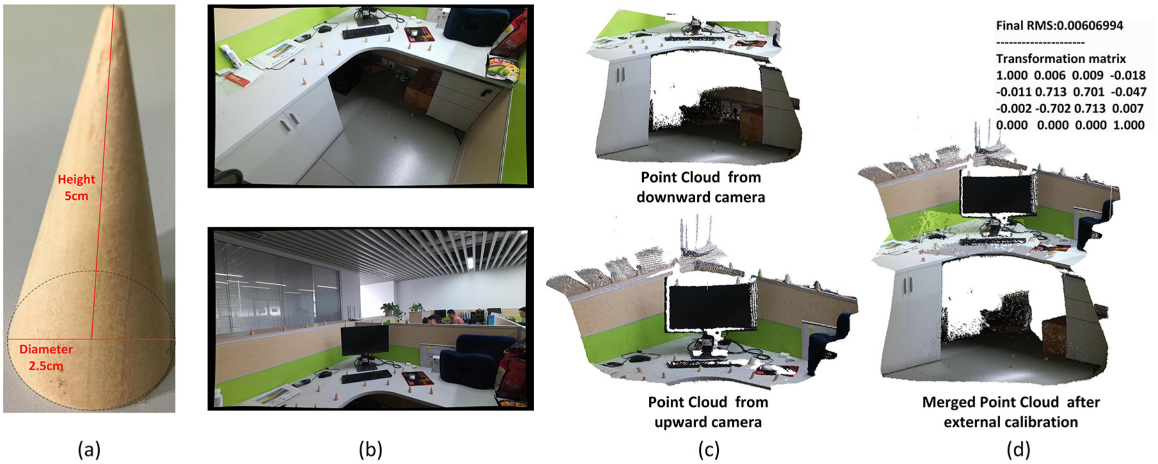


Figure 3. External calibration of multiple stereo RGB-D cameras, with (a) 3D markers for external calibration; (b) RGB frames from stereo cameras for calibration; (c) point clouds from stereo cameras for calibration; (d) the merged point cloud after external calibration and external parameter-setting.

frames that are captured by different sensors generally share the same time stamp and that the drift is negligible. As shown in Figure 5b, the blue box represents one bundle frame, which consists of one frame plotted with the blue dot and one frame plotted with the red dot, representing data captured by the reference sensor and slave sensor, respectively. In this condition, the frames from reference F^r and the slave sensor F^s can be precisely registered with the calibrated external parameter T^r_s, as shown in the left of Figure 5a.

As mentioned above, time drift in the frames of a bundle frame is inevitable. As shown in the right of Figure 5a, a significant time drift exists between F^r and F^s in condition 2. To enable accurate use of observations from multiple cameras, T^{drift} is applied to compensate for the drift of each bundle frame. A Td-C strategy is proposed to derive the compensating transformation and to eliminate the discrepancy of the data streams from the reference and the slave sensors.

In this Td-C strategy, we derive the accurate trajectory drift for each bundle frame in a spatially variant way. In Figure 5b, two bundle frames, $BF1$ and $BF2$, represent the adjacent key bundle frames captured by stereo sensors, which consist of $BF1^r$ and $BF1^s$, and $BF2^r$ and $BF2^s$, respectively. By mapping the time stamp of the slave frame $BF1^s$ to the timeline of the reference sensor, we hypothesize that one fictitious frame $BF1^{r'}$ exists in the data stream of the reference sensor, which is denoted by the same time stamp of $BF1^s$ and plotted with a yellow dot in Figure 5b. Therefore, frame $BF1^s$ and frame $BF1^{r'}$ can be precisely registered according to Equation 4 (below),

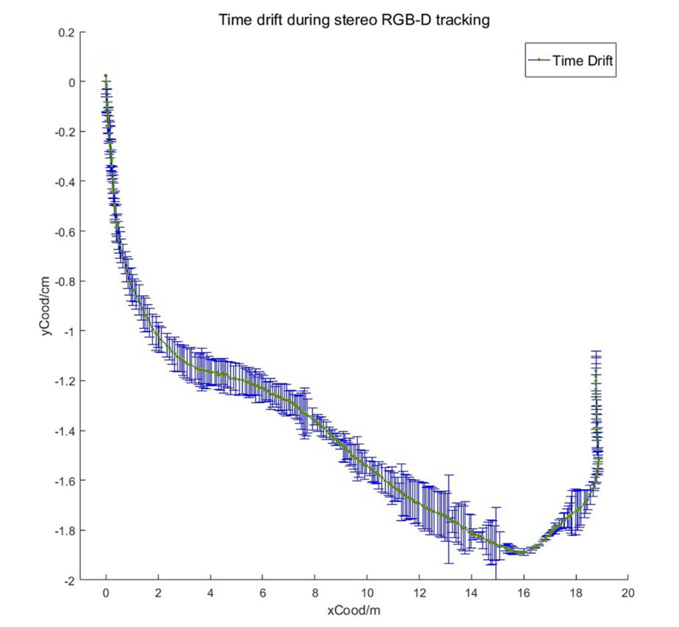


Figure 4. Time drift in each bundle frame.

and the relation of $BF1^r$ and $BF1^{r'}$ can be described by Equation 3 (below). Based on Equations 2 and 3, an accurate relative pose of the reference frame $BF1^r$ and the slave frame $BF1^s$ in $BF1$ can then be derived as Equation 4, below.

$$BF1^{r'} = T_s^r \cdot BF1^s \tag{2}$$

$$BF1^r = T^{\text{drift}} \cdot BF1^{r'} \tag{3}$$

$$BF1^r = T^{\text{drift}} \cdot T_s^r \cdot BF1^s \tag{4}$$

From a global perspective, the camera pose is in a nonlinear variant rule. In our method, only two adjacent key bundle frames are considered and used for trajectory drift compensation. Locally, we hypothesise that the translation and rotation vary linearly with time. Therefore, in our method, a linear basis is imposed on the translation and rotation to recover the accurate relative pose T^{drift} of the fictitious frame $BF1^{r'}$ and the reference frame $BF1^r$. Using $\text{ts.interval} = ||\text{ts}^{BF2^r} - \text{ts}^{BF1^r}||$ to represent the time interval between $BF1^r$ and $BF2^r$, and $\text{ts.drift} = ||\text{ts}^{BF1^s} - \text{ts}^{BF1^r}||$ for the time drift in $BF1$, which is the time difference of the frame captured by the reference sensor and the frame captured by the slave sensor, a scale parameter S is computed using Equation 5, as follows:

$$S = \frac{\text{ts.drift}}{\text{ts.interval}} = \frac{||\text{ts}^{BF1^s} - \text{ts}^{BF1^r}||}{||\text{ts}^{BF2^r} - \text{ts}^{BF1^r}||}, \text{ with } S \in [0,1] \tag{5}$$

where ts is the time stamp of a specific frame, ts^{BFi^r} is the time stamp of the reference frame in the i^{th} "bundle frame", and similarly, ts^{BFi^s} is the time stamp of the slave frame in the i^{th} "bundle frame". It should be noted that the acquisition time of the slave frame $BF1^s$ is always located between the time stamp of $BF2^r$ and $BF1^r$. Therefore, the value S always lies in interval [0,1]. As the SLAM framework is separated into two threads, the camera motion $T_{BF1^r} = (t_{BF1^r}, rot_{BF1^r})$, $T_{BF2^r} = [t_{BF2^r}, rot_{BF2^r}]$ of $BF1^r$ and $BF2^r$ is derived in the first thread. Using the linear basis, the camera position $t^T_{BF1^{r'}} = (x_{BF1^{r'}}, y_{BF1^{r'}}, z_{BF1^{r'}})^T$ of the fictitious frame $BF1^{r'}$ can then be calculated using Equation 6, as follows:

$$t^T_{BF1^{r'}} = t^T_{BF1^r} + S \cdot \left(t^T_{BF2^r} - t^T_{BF1^r}\right) \tag{6}$$

where $t^T_{BF1^r} = (x_{BF1^r}, y_{BF1^r}, z_{BF1^r})^T$ is the camera position of frame $BF1^r$, and $t^T_{BF2^r} = (x_{BF2^r}, y_{BF2^r}, z_{BF2^r})^T$ is the camera position of frame $BF2^r$.

Similarly, a linear basis is used to interpolate rotation quantities. This is achieved by the spherical linear interpolation (slerp) operation, which interpolates the rotation over the sphere, as shown in Equation 7:

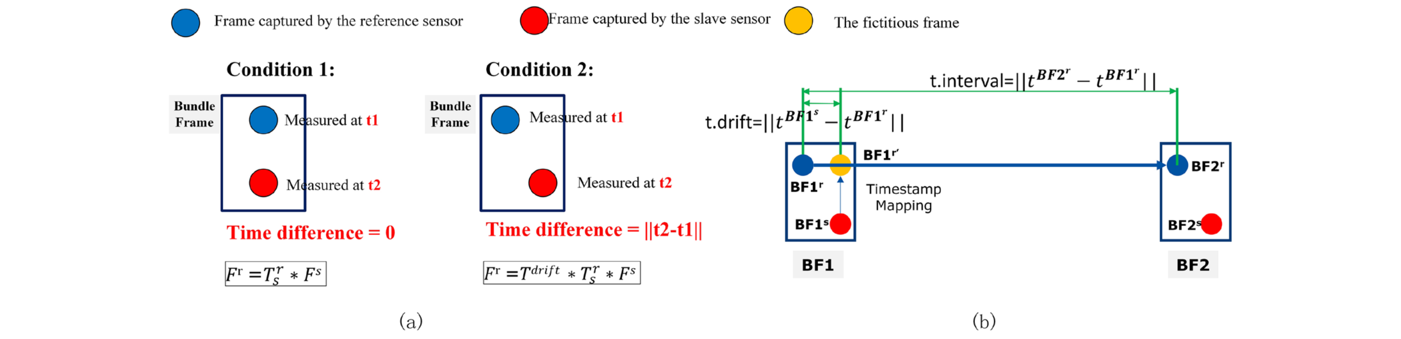


Figure 5. Two conditions in bundle frames and Td-C strategy. (a) Time-drift problems in bundle frame. (b) Td-C strategy for the bundle frame.

$$\begin{aligned}&\mathrm{rot}_{BF1^{r'}} = \mathrm{slerp}\left(S, \mathrm{rot}_{BF1^{r}}, \mathrm{rot}_{BF2^{r}}\right) = \\ &\frac{\sin\left((1-S)\alpha\right)}{\sin(\alpha)} \cdot rot_{BF1^{r}} + \frac{\sin(S\alpha)}{\sin(\alpha)} \cdot \mathrm{rot}_{BF2^{r}} \\ &\text{with } S \in [0,1]\end{aligned} \tag{7}$$

which linearly interpolates between two quaternions $\mathrm{rot}_{BF1^{r}}$, $\mathrm{rot}_{BF2^{r}}$ respectively, and where $\cos(\alpha) = \mathrm{rot}_{BF1^{r}} \cdot (\mathrm{rot}_{BF2^{r}})^{T}$. More information on the slerp operation was given by Shoemake (1985).

After that, the camera motion $T_{BF1^{r'}} = \left(t_{BF1^{r'}},\ \mathrm{rot}_{BF1^{r'}}\right)$ of the fictitious frame $BF1^{r'}$ is obtained, and T^{drift} for each bundle-frame can be recovered as Equation 8, as below:

$$T^{\mathrm{drift}} = T_{BF1^{r'}}{}^{-1} \cdot T_{BF1^{r}} \tag{8}$$

Coarse-to-Fine Stereo RGB-D Tracking

In our stereo RGB-D visual SLAM system, the camera-tracking module consists of two separate threads. The first thread is responsible for key-frame detection and initial tracking with the data stream from the reference sensor. The second thread is then used for Td-C and pose optimization by integrating all observations from the stereo bundle frames.

Camera Projection Model and Pose Update

In our system, two camera projection models are used, a depth camera projection model, and a camera projection model for pose updating. The depth camera projection model describes the relationships of image space in the depth frame and in local object space. Based on the calibrated internal parameters, i.e., focal length, principle points, and distortions, each pixel with valid measurement information is projected to object space, which enables the corresponding 3D points to be calculated. The depth camera projection model is given by Equation 9, as follows:

$$u_j^i = \frac{1}{d} K^i P_j^i \tag{9}$$

where $u_j^i = (x,y)^T$ are the image coordinates of the j^{th} point of the sensor C_i; $P_j^i = (X,Y,Z)^T$ are the 3D coordinate of the j^{th} point of the sensor C_i; d is the corresponding depth value in the depth image, which is equal to the Z value of P_j^i; and

$$K^i = \begin{bmatrix} f_{dx}^i & 0 & c_{dx}^i \\ 0 & f_{dy}^i & c_{dy}^i \\ 0 & 0 & 1 \end{bmatrix},$$

the interior matrix of the depth camera of sensor C_i.

The depth camera projection is used for 2D–3D mapping from the depth image to 3D space, which provides an absolute constraint during the pose update. Based on the calibrated RGB camera parameters, the camera projection model for the pose update is constructed per Equation 10, as follows:

$$u_j^i = \ell_{C_i}\left(T_{C_i}{}^{k} P_j^i\right) \tag{10}$$

where ℓ_{C_i} is the projection model of the RGB sensor of sensor C_i with consideration of lens distortion, and $T_{C_i}{}^{k}$ is the camera pose of the k^{th} key bundle frame of sensor C_i, which consists of a rotational and a translational component.

In the stereo RGB-D mapping system, the pose update is computed by integrating all observations from all cameras, in which the relative pose of sensors dynamically derived by the Td-C strategy is used as an absolute constraint. The pose update of the reference sensor can be expressed with one transformation matrix μ, where $T_{C_1}{}^{k}$ and $T_{C_1}{}^{k-1}$ are the reference sensor pose of the k^{th} and $(k–1)^{th}$ key bundle frame, as described by Equation 11, below:

$$T_{C_1}{}^{k} = \mu\ T_{C_1}{}^{k-1} \tag{11}$$

The pose of the slave sensor is updated by applying the dynamically derived transformation relative to the reference sensor. The relations between the reference sensor and the slave sensor in the adjacent key-frame can be represented as Equations 12 and 13, as follows:

$$T_{C_1}{}^{k} = T^{\mathrm{drift}}{}_{k} \cdot T_s^r \cdot T_{C_2}{}^{k} \tag{12}$$

$$T_{C_1}{}^{k-1} = T^{\mathrm{drift}}{}_{k-1} \cdot T_s^r \cdot T_{C_2}{}^{k-1} \tag{13}$$

By combining Equations 11, 12, and 13, the pose update for the slave sensor can be derived as the following Equation 14:

$$T_{C_2}{}^{k} = (T^{\mathrm{drift}}{}_{k} \cdot T_s^r)^{-1} \cdot \mu \cdot T^{\mathrm{drift}}{}_{k-1} \cdot T_s^r \cdot T_{C_2}{}^{k-1} \tag{14}$$

Therefore, the problem of the SLAM system now mainly consists of how to obtain an optimized pose update for the stereo RGB-D system.

Initial Camera Tracking with Reference Sensor

In the first thread, initial poses of the reference sensor are derived by minimizing the reprojection error of all observations detected from the adjacent key-frames. In our system, all key points detected by SIFT descriptor and feature matches are obtained with the graphics processing unit (GPU)-SIFT algorithm (Wu 2011). In the initial pose tracking stage, only feature points with valid depth information are used. Therefore, each feature point with valid depth information is projected to object space based on the depth camera projection model. The corresponding 3D coordinates are subsequently used as an absolute constraint during the pose-update calculation. According to Equation 15, the objective function with respect to the reprojection error of all observations (O_i) can be derived and the camera pose update can be achieved by an iterative least squares calculation:

$$F\left(P_j^i, T_{C_1}{}^{k}\right) = \underset{u}{\operatorname{argmin}} \sum_{i=1} \sum_{j \in O_i} \left(E_{ji}\right)^T \cdot \Omega_{ji} \cdot (E_{ji}) \tag{15}$$

where $E_{ji} = u_j^i - \overline{u_j^i}$ is the residual error of each feature point, in which u_j^i are the image coordinates detected from the color image, and $\overline{u_j^i}$ are the image coordinates of the reprojection points. Specifically, Ω_{ji} is defined for weight representation, which is related to the reliability of the feature point and represented as an information matrix. It should be noted that the accuracy of depth measurement determines the weight of each correspondence, and in our solution, a feature with a depth less than 5 m is fixed during bundle adjustment to provide an absolute constraint.

In this work, the initial pose of the reference sensor is derived by iteratively solving the problem using a nonlinear least-squares method. To estimate the rotational and translational parameters and optimize the position of correspondences, the corresponding Jacobians related to $T_{C_1}{}^{k}$ and P_j^i are derived by differentiating the error model. To enable convenient mathematical computation, quaternions are used to represent roll, yaw, and pitch rotations. Thus, in this solution, for each feature point j in the k^{th} keyframe, the Jacobian matrix of E_{ji} with respect to the parameters of translation and rotation $T_{C_1}{}^{k}$ can be derived by using a chain rule, as in Equation 16 below:

$$\mathcal{J}_{T_{C_1}{}^{k}} = \frac{\partial\left(E_{ji}\right)}{\partial\left(T_{C_1}{}^{k}\right)} = \frac{\partial\left(E_{ji}\right)}{\partial(C)}\Big|_{C=T_{C_1}{}^{k} \cdot P_j^i} \cdot \frac{\partial(C)}{\partial\left(T_{C_1}{}^{k}\right)} \tag{16}$$

where the first item of the above equation represents the Jacobian matrix of the camera projection function, and the second item is the Jacobian related to the translational and rotational components. The second item is also given by Equation 17, as follows:

$$\begin{matrix} \frac{\partial\left(E_{jix}\right)}{\partial\left(q_0\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(q_1\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(q_2\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(q_3\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(t_x\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(t_y\right)} & \frac{\partial\left(E_{jix}\right)}{\partial\left(t_z\right)} \\ \frac{\partial\left(E_{jiy}\right)}{\partial\left(q_0\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(q_1\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(q_2\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(q_3\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(t_x\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(t_y\right)} & \frac{\partial\left(E_{jiy}\right)}{\partial\left(t_z\right)} \end{matrix} \tag{17}$$

Simultaneously, the 3D position of each map point is also optimized during this iterative processing. We derive the Jacobian matrix of E_{ji} relative to the position P_j^i in a similar way, according to Equation 18:

$$\mathcal{J}_{P_j^i} = \frac{\partial\left(E_{ji}\right)}{\partial\left(P_j^i\right)} = \frac{\partial\left(E_{ji}\right)}{\partial(C)}\Big|_{C=T_{C_1}{}^{k}\cdot P_j^i} \cdot \begin{bmatrix} \frac{\partial\left(E_{jix}\right)}{\partial(X)} & \frac{\partial\left(E_{jix}\right)}{\partial(Y)} & \frac{\partial\left(E_{jix}\right)}{\partial(Z)} \\ \frac{\partial\left(E_{jiy}\right)}{\partial(X)} & \frac{\partial\left(E_{jiy}\right)}{\partial(Y)} & \frac{\partial\left(E_{jiy}\right)}{\partial(Z)} \end{bmatrix} \tag{18}$$

Pose Refinement with a Drift-Compensated "Bundle Frame"

As shown in Figure 6, the poses of the reference frames can be obtained by the initial camera-tracking progress. The trajectory drift in the bundle frame is then compensated for, and the accurate relations between the frames in the bundle frame are recovered.

When stereo RGB-D cameras are used, we use all observations detected from the adjacent bundle frame for bundle adjustment and pose refinement. As defined previously, each bundle frame consists of one frame BF_k^r from the reference sensor and one slave frame BF_k^s from the slave sensor, and the relationship between these can be represented by a rigid transformation as described in Equation 4. Therefore, for the adjacent bundle frames, two sets of 3D observations P_1^m and P_1^n of the corresponding image observations O_1 and O_2 are detected from the adjacent frames of the reference sensor and the slave sensor, respectively. The reprojection error for each set of observations can be represented by Equation 19, as shown below:

$$\begin{aligned} E_{1,k}^m &= \ell_{C_1}\cdot\left(T_{C_1}{}^{k}\cdot P_1^m\right)-u_{1,k}^m \\ E_{1,k-1}^m &= \ell_{C_1}\cdot\left(T_{C_1}{}^{k-1}\cdot P_1^m\right)-u_{1,k-1}^m \\ E_{2.k}^n &= \ell_{C_2\cdot}\left(T_{C_2}{}^{k}\cdot P_2^n\right)-u_{2k}^n,\ \text{in which}\ T_{C_2}{}^{k}=T^{\text{drift}}{}_{k}\cdot T_s^r\cdot T_{C_1}{}^{k} \\ E_{2,k-1}^n &= \ell_{C_2\cdot}\left(T_{C_2}{}^{k-1}\cdot P_2^n\right)-u_{2,k-1}^n,\ \text{in which}\ T_{C_2}{}^{k-1}=T^{\text{drift}}{}_{k-1}\cdot T_s^r\cdot T_{C_1}{}^{k-1} \end{aligned} \tag{19}$$

Here, $E_{1,k}^m$, $E_{1,k-1}^m$, $E_{2,k}^m$, and $E_{2,k-1}^m$ are the reprojection errors of the feature points of the reference sensor and the slave sensor in the k^{th} and the $(k–1)^{th}$ key "bundle frame", $T_{C_1}{}^{k}$, $T_{C_1}{}^{k–1}$, $T_{C_2}{}^{k}$, and $T_{C_2}{}^{k–1}$ are the corresponding poses of the kth and the $(k–1)^{th}$ key bundle frame of the reference and the slave sensor, respectively, in which the relations between the frames in each bundle frame are derived by external calibration and a Td-C processes, u_1^m, $u_{1,k–1}^m$, u_{2k}^n, and $u_{2,k–1}^n$ are the image coordinates of the feature points of the reference sensor and the slave sensor in the kth and the $(k–1)^{th}$ key bundle frame, and ℓ_{C_1} and ℓ_{C_2} are the projection functions of the RGB camera of sensor C_1 and C_2, with consideration of lens distortion. Thus, a unified error function can be modelled in Equation 20, as follows below. This method allows the full integration of 2D and 3D observations in O_1 and O_2.

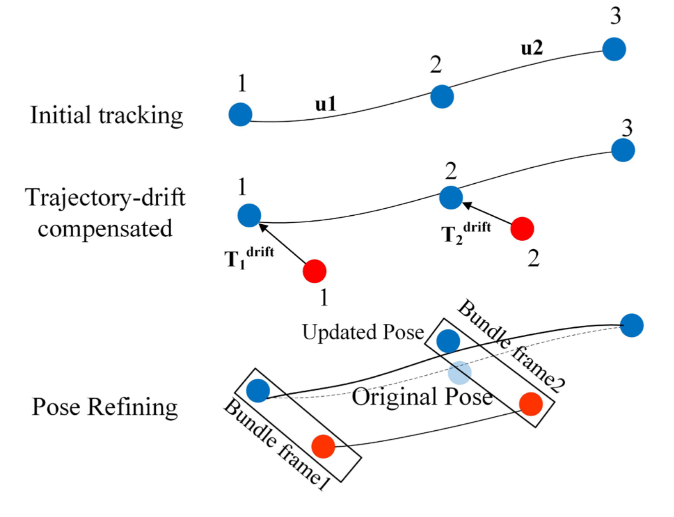


Figure 6. Demonstration of coarse-to-fine stereo RGB-D tracking.

$$\begin{aligned} &F\left(P_1^m,P_2^n,T_{C_i}{}^{k},T_{C_i}{}^{k-1}\right)= \\ &F\left(P_1^m,P_2^n,T_{C_i}{}^{k},T_{C_i}{}^{k-1}\right)= \\ &F\left(P_1^m,P_2^n,T_{C_i}{}^{k},T_{C_i}{}^{k-1}\right)= \\ &+\Sigma\sum_{n\in O_2}\left(E_{2.k}^n\right)^T\cdot\Omega_{j1}\cdot E_{2.k}^n)+\Sigma\sum_{n\in O_2}\left(E_{2.k-1}^n\right)^T\cdot\Omega_{j1}\cdot E_{2.k-1}^n) \end{aligned} \tag{20}$$

For the pose-tracking of stereo sensors, the optimization problem is to find the optimal pose update μ for the system between the $(k–1)^{th}$ key bundle-frame and the kth key bundle-frame, with reference to Equation 21:

$$\mu = \underset{\mu}{\operatorname{argmin}}\left(P_1^m,P_2^n,T_{C_i}{}^{k},T_{C_i}{}^{k-1}\right) \tag{21}$$

In this condition, the problem is solved iteratively by a nonlinear least-squares method. Thus, the pose updates related to the next key frame can be refined and improved, and the pose of the reference and slave cameras can then be derived by Equations 10 and 13.

The abovementioned work enables robust camera-tracking by integrating all of the observations from the stereo-RGB-D sensors. However, drift-error inevitably occurs during successive frame-registration, which then accumulates over trajectory length and time. In this solution, we use a bag-of-word-based technique (Gálvez-López and Tardos 2012) for loop-closure detection. After that, a vertex-edge pose graph proposed by Grisetti *et al.* (2011) is used to represent the loop-closure constraint, in which vertices contain poses of all key frames, and the edges are the corresponding relations between the key frames obtained during frame alignment. Therefore, the core idea of the global optimization problem is to distribute the error over the whole loop, which is then solved by a nonlinear least-squares optimization.

Experimental Analysis

Data Acquisition and Error Metrics

In our system, all sensors are locked on a stable stem and connected to an NVIDIA nano-development board running Ubuntu 16.04 and ROS Kinetic via a USB 3.0 interface mounted on a trolley, as shown in Figure 1. As the official software development kit for Kinect V2 can only support a single sensor, the open-source driver OpenKinect is used to power the stereo Kinect v2s system for data collection. The RGB-D sensor comprises a depth camera and an RGB camera, and the raw streamed depth and color images are initially not aligned. We use the OpenNI-driver to guarantee pixel-level alignment of depth and color images.

To obtain the absolute camera pose of the RGB-D system, we use an external laser system, GeoSLAM ZEB-REVO (Cadge 2016), which provides 1–3 cm relative mapping-accuracy. To ensure the consistency of mapping results from RGB-D and ZEB-REVO systems, a GeoSLAM ZEB-REVO system is fixed on the platform, as shown in Figure 7a. Careful extrinsic calibration is conducted between the RGB-D sensor and the ZEB-REVO sensor. In our system, the initial transformation between each RGB-D sensor and the ZEB-REVO system is calculated with dozens of markers attached on the wall. An accurate rigid transformation is then derived from their respective ICP progress. Figure 7b shows the sample point-cloud collected by the stereo RGB-D system and ZEB-REVO system.

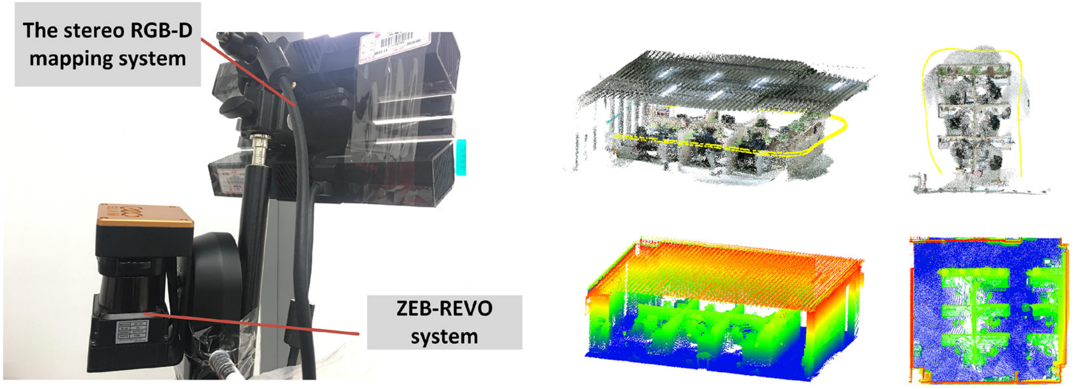
In our experiments, two data sets are collected to verify the performance of the proposed stereo RGB-D mapping solution. Figure 8 depicts the RGB and depth images taken at various camera poses for various trajectories in the office and hall-space scenes, respectively. All frames are recorded at 640 × 480 resolution and streamed at a 10 Hz frame rate. Correspondingly, the point cloud and trajectory from the ZEB-REVO system is used for accuracy evaluation, as shown in Figure 9.

Generally, an RGB-D SLAM system generates the camera pose and the corresponding 3D point cloud. While it is necessary to evaluate the quality of the generated point cloud and camera trajectories for algorithm verification, for each set of data, the results from a single RGB-D sensor and from the stereo RGB-D sensors are both evaluated. Therefore, two kinds of metrics are used to quantify the accuracy of camera-tracking and 3D mapping, as described below.

1. Our trajectory estimation statistics are inspired by previous studies (Handa *et al.* 2014; Sturm *et al.* 2012) that used an absolute trajectory error (ATE) to quantify the accuracy of an entire trajectory. This method involves calculation of the RMSE of the Euclidean distances between the estimated trajectory P_i and the ground truth trajectory obtained from the ZEB-REVO system Q_i. To unify the coordinate frames of both systems, we register the trajectory of the reference RGB-D sensor and the slave sensor to that of the ZEB-REVO system by a rigid-body transformation S^{ref}, S^{slave} calculated with a carefully extrinsic calibration process. Based on this transformation, the absolute error of the trajectory at time stamp i can be calculated by Equation 22, as below:

$$\mathrm{RMSE}(e_i)\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{trans}(e_i)^2\right)^{\frac{1}{2}} \tag{22}$$

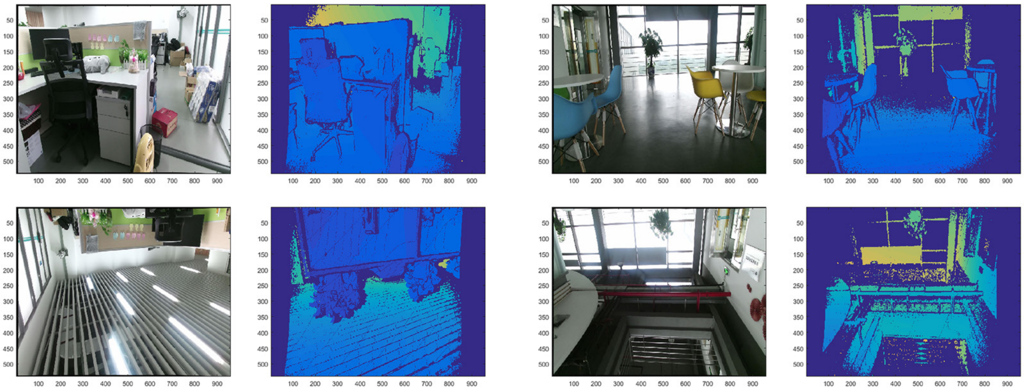
Thus, we evaluate the RMSE over all time indices of the translational components as Equation 23 as below, where trans(e_i) refers to the translational components of the relative pose-error e_i:



(a) data collection system

(b) point cloud from the stereo Kinect system and the ZEB-REVO system

Figure 7. Data collection system and point cloud from ZEB-REVO and RGB-D systems.



(a) RGB-D frames of office scene taken at different camera poses

(b) RGB-D frames of hall-space scene taken at different camera poses

Figure 8. RGB-D frames of two scenes taken at different camera poses.

$$\mathrm{RMSE}(e_i)\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{trans}(e_i)^2\right)^{\frac{1}{2}} \tag{23}$$

2. To quantify the accuracy of 3D reconstruction, the reconstructed point cloud from the RGB-D system is first coarsely aligned with the point cloud from the ZEB-REVO system by manually selecting point correspondences. Point cloud from the RGB-D is then finely aligned to the point cloud from the ZEB-REVO via ICP. Finally, for each point, the closest point in the point cloud from the ZEB-REVO is located, as is the perpendicular distance between the point and the reference point cloud. The standard deviation is computed over the distances for all points.

Experimental Results

Office Scene

The office scene data sets contain two sets of RGB-D sequences recorded from the stereo RGB-D system. The stereo sensor works at 10 Hz with a resolution of 640 × 480 pixels after rectification. As discussed in the section "Trajectory- Drift–Compensated (Td-C) Approach", each bundle frame consists of two frames, one from the reference sensor and one from the slave sensor, and a significant time drift was seen in each bundle frame due to the unstable topic-publish rate of the sensors. As shown in Figure 10a, the time drift of each bundle frame in the office scene is plotted together with the camera trajectory, in which the red dots are the reference sensor's position after camera-tracking, the yellow dot is the starting point of this scan, and the time drift is represented by the blue error bar. Except for the starting point, the time drift is randomly distributed over the whole trajectory.

Similarly, Figure 10b comprises a plot of the time drift together with the time stamp of the data stream. Quantitatively, the maximum time drift in this scene is approximately 55 ms, and the average drift is about 17.4 ms. Figure 10b shows that the time drift in each bundle frame is generally irregular and unpredictable, which is difficult to model with a unified mathematic model. This will have a large effect on the tracking accuracy of the stereo RGB-D system. Therefore, the time drift in each bundle frame is compensated for by adding an extra transformation to the relationship between the reference sensor and the slave sensor during stereo RGB-D tracking, as detailed in the section "Trajectory- Drift– Compensated (Td-C) Approach".

In this experiment, the camera trajectory and the point cloud obtained by the ZEB-REVO device are used as the ground truth for accuracy evaluation. Camera tracking experiments were conducted and compared with the data set before and after the Td-C process.

The performance of the proposed Td-C stereo RGB-D mapping method was initially evaluated with the absolute translation RMSE of the camera trajectory. Based on the calibrated external parameters between the RGB-D sensor and the ZEB-REVO system, the estimated camera trajectories are transformed to the coordinate system of the ZEB-REVO system for comparison.

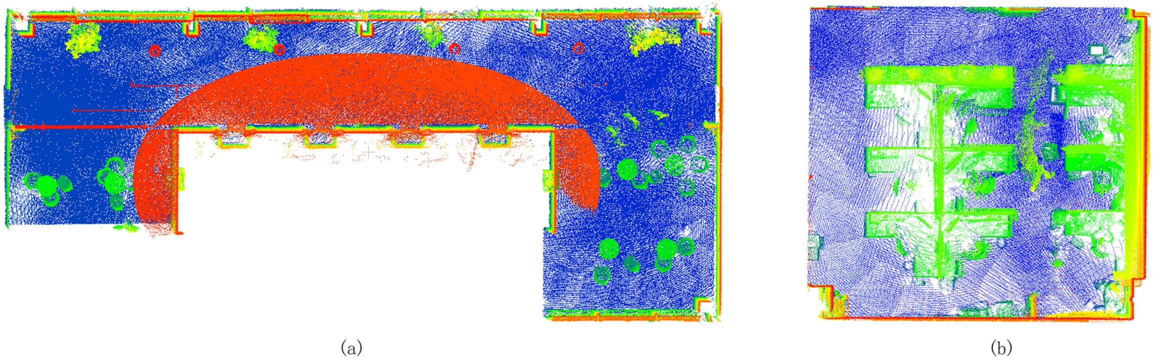


Figure 9. Corresponding point clouds of office and hall space captured by a ZEB-REVO system.

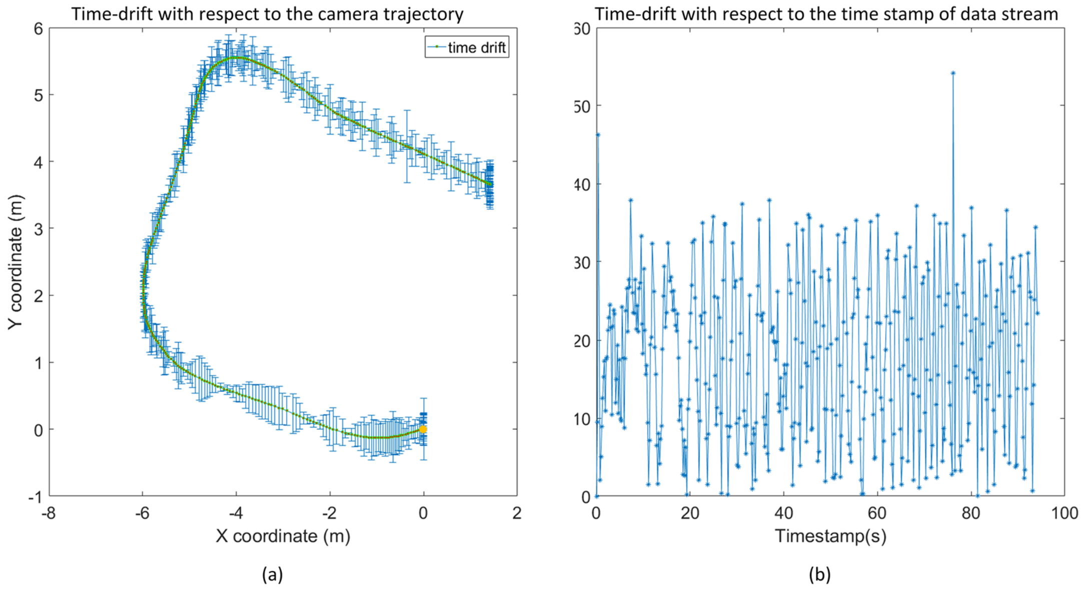


Figure 10. Time drift between the data stream of stereo RGB-D sensors in office scene.

As demonstrated in Figure 11, the estimate trajectories and the ground truth trajectories are plotted and the translation errors of all key bundle frames are represented, together with the trajectories. Table 1 lists all statistics for the accuracy of the reconstruction, including the RMSE in the X, Y, and Z directions, the RMSE of the translation error, and the relative error of the tracking length. As shown in Figure 11, the discrepancies represented by the red lines between the estimated trajectory and the ground truth are improved in the experiment with the data set after the Td-C process than before, perhaps due to more accurate relationships between the reference and the slave sensor.

Table 1 summarizes the RMSE of the translation error for the two conditions. The results after the Td-C process were better, again verifying the performance of the proposed method. The RMSE of the translation error is improved from 0.287 m to 0.335 m, and the relative error is improved from 1.42% to 1.66%. This can be explained by the fact that more reliable visual features can be obtained from the data stream after the Td-C process, which provide a better alignment. The inconsistency between the features from stereo frames before the Td-C process could introduce larger pose drifts, which will accumulate throughout the operation.

The trajectory error explains how the camera-tracking method performs in frame-to-frame tracking but does not imply a better reconstruction is possible. In addition, the absolute mapping error is calculated by comparison with the point cloud generated from the ZED-REVO system. The estimated point cloud from the stereo RGB-D sensor is first registered to the laser system, and ICP is used to refine the alignment. The standard deviation computed over the error for all reconstructed points is used as a metric. Figure 12 shows the estimated point cloud, the heat maps of errors for the 3D reconstructions, and a histogram of the approximate distances of the office scene. The heat maps highlight the least accurate areas of the reconstruction. The range of errors in the heat map of error and the histogram of approximate distances are scaled to a range of 0 to 0.2 m, for comparison purposes. As expected, the odometer without Td-C processing generates worse results and has a large proportion of least accurate areas. In the heat map, a sizable discrepancy can be found in the loop region in the 3D reconstruction result before the Td-C process, which is observed both at the start point and at the end point of this scanning. In the histogram of approximate distances (Figure 13), the accumulated percentage of the points within 5 cm error are calculated and the odometer with Td-C processing and without Td-C processing achieves 88.539% and 87.897% accuracy respectively, which again verifies the performance of the proposed method. Table 2 lists the average error of the reconstruction before Td-C and after Td-C, and the data show that the average error of the reconstruction improves from 0.018 m to 0.014 m. Therefore, in these scenes, the Td-C strategy improves the tracking accuracy of the stereo system and the 3D reconstruction.

Hall-Space Scene

The stereo RGB-D sequences are recorded in a hall space with a 26.5-m trajectory length. Similarly, the stereo sensor operates at 10 Hz with a resolution after rectification of 640 × 480 pixels. The distribution of the time drift is shown in Figure 13. As expected, the value of the time drift is mainly distributed between 10 ms and 35 ms. Quantitatively, the maximum time drift in this scene is approximately 60 ms and the average drift is about 22 ms. A Td-C strategy is applied to each bundle frame during camera-tracking.

The performance of the proposed stereo RGB-D mapping approach was initially evaluated with the absolute translation RMSE of the camera trajectory. Figure 14 presents the estimate trajectories, the ground truth trajectories, and the translation errors of all key bundle frames with respect to the trajectories. Table 1 lists the statistics for the accuracy of the reconstruction, including the RMSE in the X, Y, and Z directions, the RMSE of the translation error, and the relative error of the tracking length. As Table 1 shows, the tracking accuracy is better in the experiment after Td-C than in the experiment before Td-C, confirming that the proposed Td-C solution improves the accuracies in all three directions. The RMSE of the translation error improves from 0.397 m to 0.443 m, and the relative error improves from 1.50% to 1.67%.

Figure 15 shows the estimated point cloud, heat maps of errors for 3D reconstructions, and a histogram of approximate distances of the hall-space scene. The error of the 3D

Table 1. Comparison of the ATE for incremental registration of the RGB-D sequences before and after Td-C processing.

Data set	Td-C used?	Length (m)	RMSE.X (m)	RMSE.Y (m)	RMSE.Z (m)	RMSE (m)	Proportion (%)
Office scene	No	20.2	0.213	0.242	0.092	0.335	1.66
	Yes	20.2	0.183	0.211	0.069	0.287	1.42
Hall space scene	No	26.5	0.323	0.277	0.124	0.443	1.67
	Yes	26.5	0.301	0.242	0.092	0.397	1.50

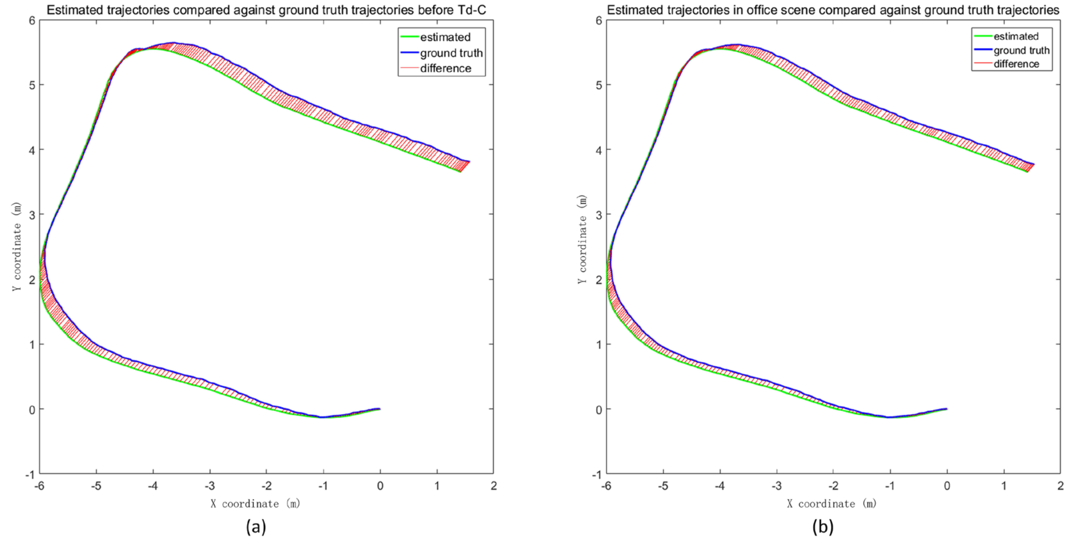


Figure 11. Estimated trajectories from the office scene compared against ground truth trajectories: (a) estimated trajectories before the Td-C strategy; (b) estimated trajectories after the Td-C strategy.

reconstruction is accumulated with the mapping distance, which is consistent with the trend of trajectory error. For comparison purposes, the range of errors in the heatmap of error and the histogram of approximate distances are unified, and the extent is set from 0 to 1.6 m. As expected, the mapping results without Td-C processing generate worse results with a large least-accurate area. In the histogram of approximate distances, the accumulated percentage of points with accuracy greater than 0.2 m is quantified. The odometer achieves 88.992% and 78.045% accuracy with Td-C processing and without Td-C, respectively. According to Table 2, the average error of the reconstruction improves from 0.094 m to 0.057 m, again verifying the performance of the proposed method.

Table 2. Comparison of absolute error for 3D reconstruction before and after Td-C.

Data Set	Td-C used?	Avg. error (m)
Office scene	No	0.018
	Yes	0.014
Hall-space scene	No	0.094
	Yes	0.057

Technical University of Munich Data Sets

As there are no stereo RGB-D data sets available for accuracy comparison, the Technical University of Munich (TUM) public data set, which is collected with a single RGB-D sensor, is used to further demonstrate the performance of our proposed SLAM

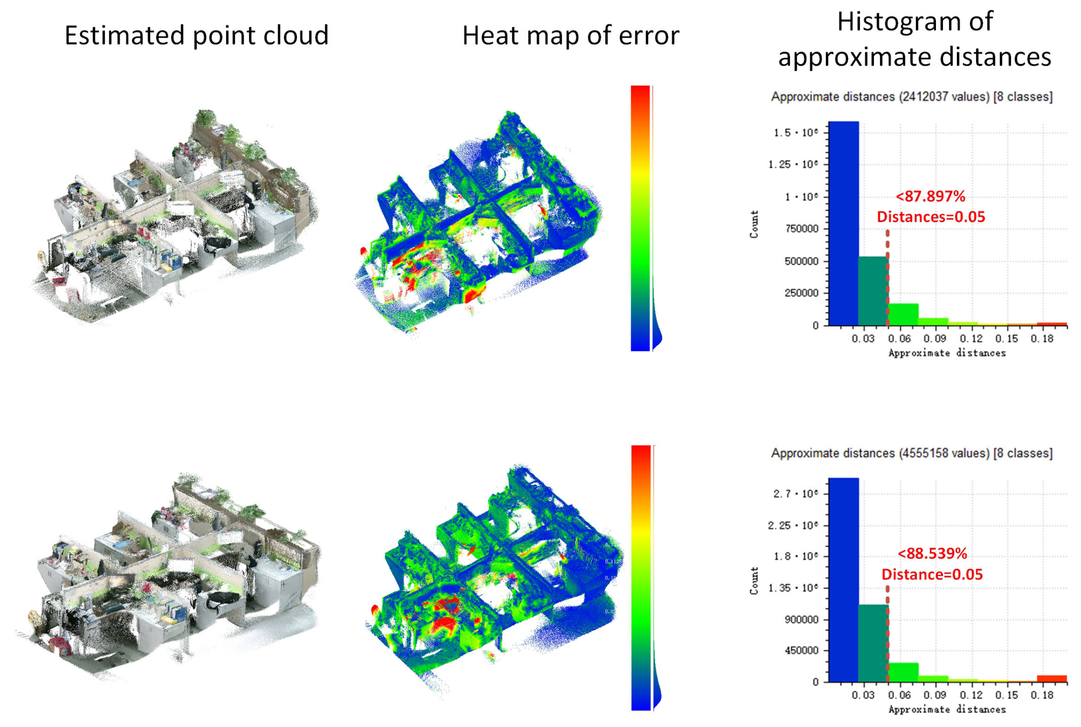


Figure 12. Estimated point cloud heat maps for 3D reconstruction and histogram of approximate distances in the office scene. The color coding is relative to the error obtained, in terms of the (a) accuracy of the 3D reconstruction before Td-C processing, and the (b) accuracy of the 3D reconstruction after Td-C processing.

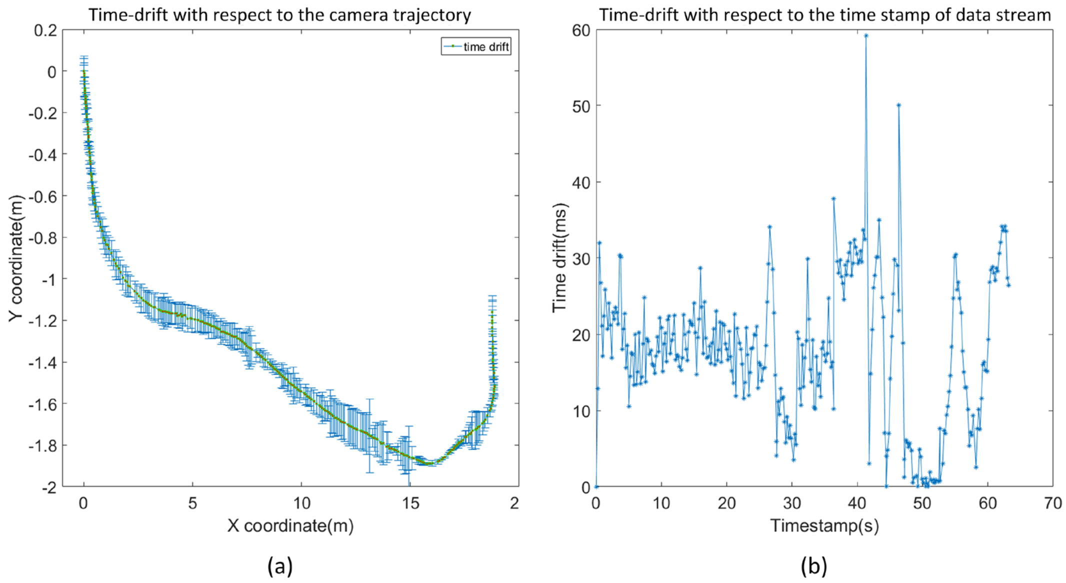


Figure 13. Time drift between data stream of stereo RGB-D sensors in the hall-space scene.

pipeline. We apply our solution on four sequences with different texture, illumination and structure conditions, and compare the experimental results of our system with the following state-of-the-art SLAM methods: Kintinuous (Whelan *et al.* 2012), dense visual odometry (DVO)-SLAM (Kerl *et al.* 2013), and RGB-D SLAM (Endres *et al.* 2014). As shown in Table 3, the proposed RGB-D SLAM system achieves the best performance in two sequences, namely fr1/room, fr2/xyz. In addition, Figure 16 shows the point clouds that result from back-projecting the sensor-depth maps from the computed keyframe poses in four sequences. The good definition and the straight contours of the point clouds prove the highly accurate localization of our approach.

Conclusions and Discussion

In this study, we propose the use of stereo RGB-D cameras in visual SLAM for better pose tracking performance and more detailed indoor environment mapping. In the stereo RGB-D system, a time drift in each bundle frame is inevitable and changes irregularly, which cannot be mathematically modelled against time drift. We propose a Td-C method to eliminate the influence of time drift during stereo camera motion tracking, which imposes an extra transformation upon the relationships of the reference sensor and the slave sensor in each bundle frame. To enable the use of observations from stereo sensors, a coarse-to-fine stereo RGB-D tracking method is proposed. A detailed mathematical analysis is presented to explain how to fuse the measurements from stereo camera for pose tracking. Via theoretical analysis and experimental validation, we conclude that the proposed Td-C stereo RGB-D mapping solution can eliminate the inconsistency between the data sequence obtained from the stereo sensors and can achieve better pose performance in both camera-tracking and 3D reconstruction.

The Td-C stereo RGB-D mapping method discussed here enables the synchronization of sequences from multiple sensors and the integration of observations from multiple sensors. This permits the full comparative and synergistic use of different data streams from different sensors, even though the system cannot synchronize them precisely. The proposed Td-C strategy can also be used in other similar systems, such as integrated processing of RGB-D sensors and laser systems.

Table 3. Comparisons of the RMSE of ATE, (in m) for incremental registration of RGB-D sequences of the TUM benchmark data set.[a]

Sequences	Our SLAM	Kintinous Fusion	DVO SLAM	RGB-D SLAM
fr1/desk	0.03	0.037	0.021	0.026
fr1/room	0.042	0.075	0.043	0.087
fr2/desk	0.062	0.34	0.017	0.057
fr2/xyz	0.011	0.029	0.018	/

[a] The best results are indicated in bold.

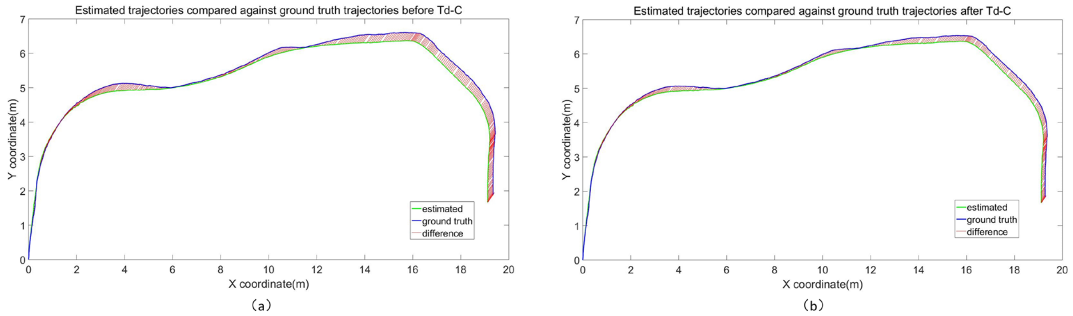


Figure 14. Estimated trajectories from hall-space scene compared against ground truth trajectories: (a) estimated trajectories before Td-C strategy, (b) estimated trajectories after Td-C strategy.

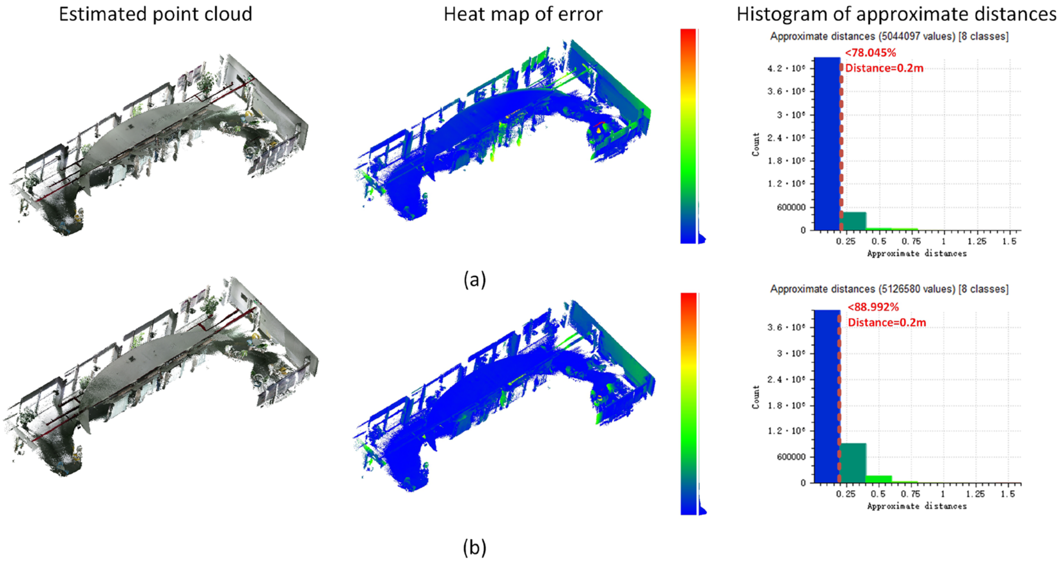


Figure 15. Estimated point cloud, heat maps for 3D reconstructions, and histogram of approximate distances in hall-space scene. The color-coding is relative to the error obtained, where (a) is the accuracy of 3D reconstruction before the Td-C strategy and (b) is the accuracy of 3D reconstruction after the Td-C strategy.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (Projects Nos. 41801392, 41901329, 41971354, and 41971341), and in part by the Research Program of Shenzhen S and T Innovation Committee under Grant (Project Nos. JCYJ20180305125131482).

References

Alismail, H., B. Browning and S. Lucey. 2016. Photometric bundle adjustment for vision-based SLAM. Pages 324–341 in *Proceedings of the Asian Conference on Computer Vision*.

Cadge, S. 2016. Welcome to the ZEB REVOlution. *GEOmedia* 20 (3).

Chen, C., B. Yang, S. Song, M. Tian, J. Li, W. Dai and L. Fang. 2018. Calibrate multiple consumer RGB-D cameras for low-cost and efficient 3D indoor mapping. *Remote Sensing* 10 (2):328.

Chow, J.C.K., D. D. Lichti, J. D. Hol, G. Bellusci and H. Luinge. 2014. IMU and Multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning. *Robotics* 3 (3):247.

Davison, A. J., Y. G. Cid and N. Kita. 2004. Real-time 3D SLAM with wide-angle vision. *IFAC Proceedings Volumes* 37 (8):868–873.

Deng, T., J.-C. Bazin, T. Martin, C. Kuster, J. Cai, T. Popa and M. Gross. 2014. Registration of multiple RGBD cameras via local rigid transformations. Pages 1–6 in *Proceedings of the 2014 IEEE International Conference on Multimedia and Expo (ICME)*.

Dubbelman, G. and B. Browning. 2013. Closed-form online pose-chain SLAM. Pages 5190–5197 in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, 6–10 May 2013.

Endres, F., J. Hess, N. Engelhard, J. Sturm, D. Cremers and W. Burgard. 2012. An evaluation of the RGB-D SLAM system. Pages 1691–1696 in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, 14–18 May 2012.

Endres, F., J. Hess, J. Sturm, D. Cremers and W. Burgard. 2014. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics* 30 (1):177–187.

Engel, J., V. Koltun and D. Cremers. 2017. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (3):611–625.

Engelhard, N., F. Endres, J. Hess, J. Sturm and W. Burgard. 2011. Real-time 3D visual SLAM with a hand-held RGB-D camera. Pages 1–15 in *Proceedings of the* RGB-D *Workshop on 3D Perception in Robotics at the European Robotics Forum*, held in Vasteras, Sweden.

Fuhrmann, S., F. Langguth and M. Goesele. 2014. MVE-A multi-view reconstruction environment. Pages 11–18 in *Proceedings of the GCH*.

Gálvez-López, D. and J. D. Tardos. 2012. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28 (5):1188–1197.

Gao, X., R. Wang, N. Demmel and D. Cremers. 2018. LDSO: Direct sparse odometry with loop closure. 2018. Pages 2198–2204 in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .

Grisetti, G., R. Kümmerle, H. Strasdat and K. Konolige. 2011. g2o: A general framework for (hyper)graph optimization. Pages 9–13 in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, held in Shanghai, China.

Handa, A., T. Whelan, J. McDonald and A. J. Davison. 2014. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. Pages 1524–1531 in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May —7 June 2014.

He, F. and A.,Habib. 2018. Three-point-based solution for automated motion parameter estimation of a multi-camera indoor mapping system with planar motion constraint. *ISPRS Journal of Photogrammetry and Remote Sensing* 142:278–291.

Hee Lee, G., F. Faundorfer and M. Pollefeys. 2013. Motion estimation for self-driving cars with a generalized camera. Pages 2746–2753 in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

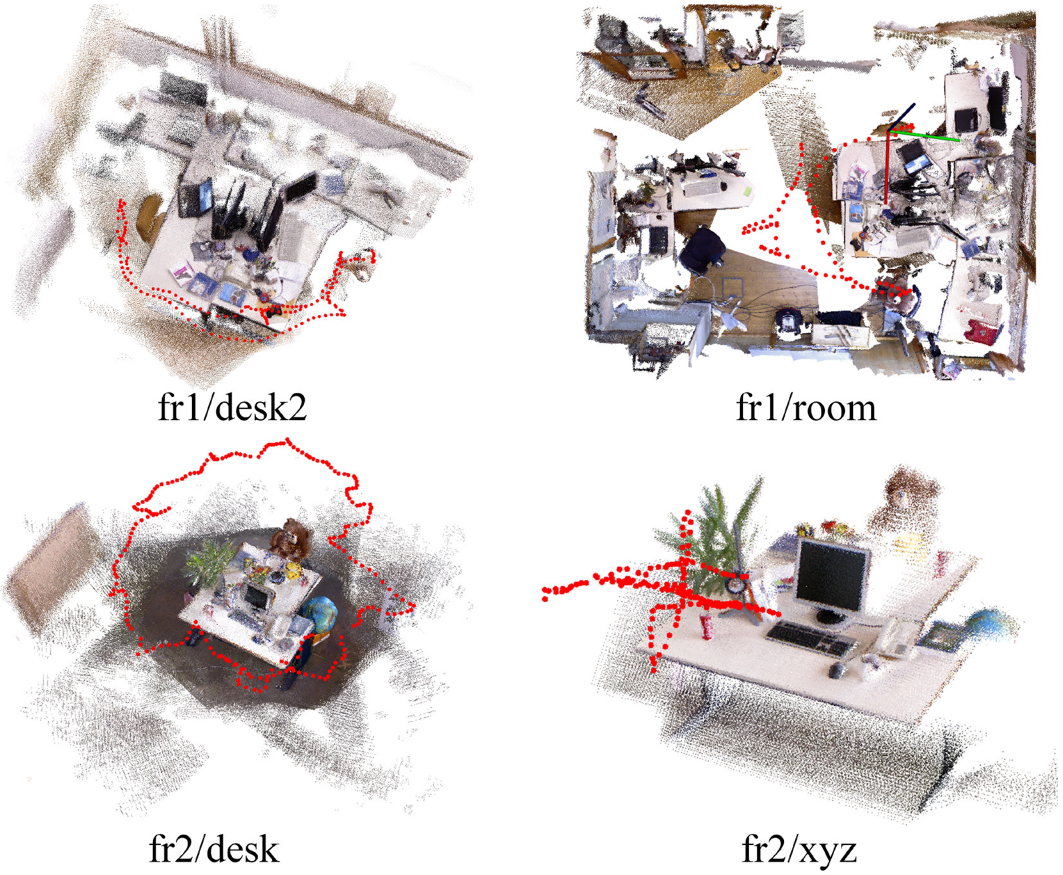


Figure 16. Dense point-cloud reconstructions from estimated key-frame poses and sensor depth-maps in TUM data sets.

Henry, P., M. Krainin, E. Herbst, X. Ren and D. Fox. 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research* 31 (5):647–663.

Henry, P., M. Krainin, E. Herbst, X. Ren and D. Fox. 2014. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental Robotics*, edited by O. Khatib, V. Kumar and G. Sukhatme, 477–491. Heidelberg/Berlin, Germany: Springer.

Kaess, M. and F. Dellaert. 2006. Visual SLAM with a multi-camera rig. Georgia Institute of Technology. <http://hdl.handle.net/1853/8726> Accessed 19 February 2020.

Kerl, C., J. Sturm and D. Cremers. 2013. Dense visual SLAM for RGB-D cameras. Pages 2100–2106 in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3–7 November 2013.

Kerl, C., J. Sturm and D. Cremers. 2013. Robust odometry estimation for RGB-D cameras. Pages 3748–3754 in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 6–10 May 2013.

Kerl, C., J. Stuckler and D. Cremers. 2015. Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras. Pages 2264–2272 in *Proceedings of the IEEE International Conference on Computer Vision*.

Khoshelham, K. and S. Elberink. 2012. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors* 12 (2):1437–1454.

Kim, P., B. Coltin and H. Jin Kim. 2018. Linear RGB-D SLAM for planar environments. Pages 333–348 in *Proceedings of the European Conference on Computer Vision (ECCV)*.

Klein, G. and D. Murray. 2007. Parallel tracking and mapping for small AR workspaces. Pages 1–10 in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*.

Le, P.-H. and J. Kosecka. 2017. Dense piecewise planar RGB-D SLAM for indoor environments. *arXiv* preprint arXiv:1708.00514.

Lee, G. H., F. Fraundorfer and M. Pollefeys. 2013. Structureless pose-graph loop-closure with a multi-camera system on a self-driving car. Pages 564–571 in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60 (2):91–110.

Mazaheri Tehrani, M. 2015. *Constrained Motion Estimation for a Multi-Camera System*. Master's Thesis, University of Calgary, 105 pp.

Mur-Artal, R. and J. D. Tardos. 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics* PP (99):1–8.

Newcombe, R. A., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges and A. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. Pages 127–136 in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 26–29 October 2011.

Newcombe, R. A., S. J. Lovegrove and A. J. Davison. 2011. DTAM: Dense tracking and mapping in real-time. Pages 2320–2327 in *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)*.

Nießner, M., M. Zollhöfer, S. Izadi and M. Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)* 32 (6):169.

Olivier, N., H. Uchiyama, M. Mishima, D. Thomas, R. Taniguchi, R. Roberto, J. P. Lima and V. Teichrieb. 2018. Live structural modeling using RGB-D SLAM. Pages 6352–6358 in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 21–25 May 2018.

Park, J.-H., Y.-D. Shin, J.-H. Bae and M.-H. Baeg. 2012. Spatial uncertainty model for visual features using a Kinect™ sensor. *Sensors* 12 (7):8640.

Pless, R. 2003. Using many cameras as one. Page II-587 in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Schops, T., T. Sattler and M. Pollefeys. 2019. BAD SLAM: Bundle adjusted direct RGB-D SLAM. Pages 134–144 in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Shi, Y., K. Xu, M. Nießner, S. Rusinkiewicz and T. Funkhouser. 2018. Planematch: Patch coplanarity prediction for robust RGB-D reconstruction. Pages 750–766 in *Proceedings of the European Conference on Computer Vision (ECCV)* .

Shoemake, K. 1985. Animating rotation with quaternion curves. Pages 245–254 in *Proceedings of the ACM SIGGRAPH Computer Graphics*.

Sturm, J., N. Engelhard, F. Endres, W. Burgard and D. Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. Pages 573–580 in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7–12 October 2012.

Tang, S., W. Chen, W. Wang, X. Li, W. Darwish, W. Li, Z. Huang, H. Hu and R. Guo. 2018. Geometric integration of hybrid correspondences for RGB-D unidirectional tracking. *Sensors* 18 (5):1385.

Tang, S., Y. Li, Z. Yuan, X. Li, R. Guo, Y. Zhang and W. Wang. 2019. A vertex-to-edge weighted closed-form method for dense RGB-D indoor SLAM. *IEEE Access* 7:32019–32029.

Vestena, K., D. Dos Santos, E. Oliveira Jr., N. Pavan and K. Khoshelham. 2016. A weighted closed-form solution for RGB-D data registration. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*: 403–409.

Whelan, T., M. Kaess, M. Fallon, H. Johannsson, J. J. Leonard and J. McDonald. 2012. Kintinuous: spatially extended KinectFusion. Pages 1–8 in *RSS Workshop on* RGB-D*: Advanced Reasoning with Depth Cameras*, held in Sydney, Australia.

Whelan, T., M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard and J. McDonald. 2015. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research* 34 (4–5):598–626.

Whelan, T., R. F. Salas-Moreno, B. Glocker, A. J. Davison and S. Leutenegger. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35 (14):1697–1716.

Wu, B., X. Ge, L. Xie and W. Chen. 2019. Enhanced 3D mapping with an RGB-D sensor via integration of depth measurements and image sequences. *Photogrammetric Engineering & Remote Sensing* 85 (9):633–642.

Wu, C. 2007. SiftGPU: A GPU implementation of scale invariant feature transform. *SIFT*. <http://cs. unc. edu/~ ccwu/siftgpu>.

Yang, S., S. A. Scherer and A. Zell. 2014. Visual SLAM for autonomous MAVs with dual cameras. Pages 5227–5232 in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*.

Yang, S., S. A. Scherer and A. Zell. 2016. Robust onboard visual SLAM for autonomous MAVs. In *Intelligent Autonomous Systems 13*, 361–373. Springer.

Yang, S., X. Yi, Z. Wang, Y. Wang and X. Yang. 2015. Visual SLAM using multiple RGB-D cameras. Pages 1389–1395 in *Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

Yong, D., C. Lei, W. Yucheng, Y. Min, Q. Xiameng, H. Shaoyang and J. Yunde. 2011. A real-time system for 3D recovery of dynamic scene with multiple RGBD imagers. Pages 1–8 in *Proceedings of the 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Zeng, A., S. Song, M. Nießner, M. Fisher, J. Xiao and T. Funkhouser. 2017. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. Pages 1802–1811 in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zeng, M., F. Zhao, J. Zheng and X. Liu. 2012. A memory-efficient KinectFusion using Octree. In *Computational Visual Media*, edited by S.-M. Hu and R. Martin, 234–241. Berlin/Heidelberg, Germany: Springer.