AN RGB-D DATA PROCESSING FRAMEWORK BASED ON ENVIRONMENT CONSTRAINTS FOR MAPPING INDOOR ENVIRONMENTS

Walid Darwish[1,2,*], Wenbin Li[2], Shengjun Tang[3], Yaxin Li[2], Wu Chen[2]

[1]Vrije Universiteit Brussels, Department of Electronics and Informatics, 1050 Brussels, Belgium - w.darwish@connect.polyu.hk
[2]The Hong Kong Polytechnic University, Department of Land Surveying and Geo-Informatics,
Hung Hom, Hong Kong - (wb.li, wu.chen)@polyu.edu.hk, Yaxin.pu.li@connect.polyu.hk
[3]Shenzhen University, Shenzhen Key Laboratory of Spatial Smart Sensing and Services &
The Key Laboratory for Geo-Environment Monitoring of Coastal Zone of the National Administration of Surveying,
Mapping and GeoInformation, Shenzhen 518060, China - shengjun.tang@whu.edu.cn

Commission VI, WG VI/4

KEY WORDS: RGB-D Sensor, Indoor Reconstruction, 3D features, SLAM, Constraint Mapping

ABSTRACT:

The adoption of RGB and depth (RGB-D) sensors for surveying applications (i.e., building information modeling [BIM], indoor navigation, and three-dimensional [3D] models) to replace expensive and time-consuming methods (e.g., stereo cameras, laser scanners) has recently attracted great attention. Due to the distinctive structure and scalability of indoor environments, the depth quality produced from RGB-D cameras and the simultaneous localization and mapping (SLAM) system responsible for the cameras pose estimation are substantial problems with existing RGB-D mapping systems. This study introduces a new RGB-D data processing framework that adopts two-dimensional and 3D features from RGB and depth images. To overcome the self-repetitive structure of indoor environments, the proposed framework uses novel description functions for both line and plane features extracted from RGB and depth images for further matching between successive RGB-D frame features. Also, the framework estimates the camera pose by minimizing the combined geometric distance of both two-dimensional and 3D features. Using the previously known structure of the indoor environment, the framework leverages the structural constraints to enhance 3D model precision. The framework also adopts a graph-based optimization technique to distribute the closure error to the graphs nodes and edges when a loop closure is detected. The visual RGB-D SLAM system and the default sensor tracking system (SensorFusion) were used to assess the performance of the proposed framework. The results show that the proposed framework can achieve significant improvement in 3D model accuracy.

1. INTRODUCTION

The advent of consumer–grade depth sensors (e.g., RGB-Depth [RGB-D] cameras) and their great potential for surveying applications have recently led to valuable progress in indoor positioning and mapping (Tang et al., 2016). RGB-D sensors have been used in two main research areas. The first is robotic navigation and control; in this area, the main purpose is to guide the robot to avoid indoor obstacles, so real–time performance and pose precision are the focused targets (Endres et al., 2014, Huang et al., 2017). The second area is surveying and three-dimensional (3D) model reconstruction applications; here, the main purpose is to produce a rich and complete 3D model of the surrounding environment (Dos Santos et al., 2016, Tang et al., 2016, Tsai et al., 2015).

The main limitations that prevent the deployment of RGB-D sensors in high–precision surveying application are the limited depth range of the sensor, around three meters for structured light-based RGB-D sensors (e.g., Structure Sensor) and five meters for time–of–flight-based RGB-D sensors (e.g., Kinect v2), and successive brittle frame (i.e., frames have fewer distinctive features) registration that leads to lost tracking or camera pose bias.

Extensive studies have been carried out to improve depth quality since the first version of the structured light RGB-D camera (Kinect v1) was released as a remote videogame controller in 2010. The depth precision of an RGB-D sensor can be enhanced with a proper calibration method (e.g., disparity-based calibration model (Darwish et al., 2017), distortion-based error model (Herrera et al., 2011, Herrera et al., 2012), photogrammetric bundle adjustment method (Chow , Lichti, 2013), covariance-based error model (Pagliari , Pinto, 2015)). Compensation for the systematic error and depth distortion were achieved by revealing the calibration parameters and the error model coefficients.

The RGB-D camera location problem can be recovered using a suitable tracking algorithm (i.e., the early and basic tracking algorithm is Kinect Fusion (Newcombe et al., 2011)). The widely–used family of algorithms called RGB-D simultaneous localization and mapping (SLAM) (Stachniss et al., 2007) uses visual features with corresponding depth information to obtain the relative movement between successive RGB-D frames. Due to the depth quality and mismatching between pairs of RGB images, random sample consensus (RANSAC) (Fischler , Bolles, 1981) and iterative closest point (ICP) (Bae , Lichti, 2008, Rusinkiewicz , Levoy, 2001) algorithms are applied to the computed relative pose to further filter out the matched outliers and refine the pose, respectively.

However, RANSAC and ICP can only in theory reduce random error related to matching and depth noise. In practice, the performance of RGB-D SLAM algorithms suffers from a

[*]Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-263-2019 | © Authors 2019. CC BY 4.0 License.

systematic pose drift problem. The main reasons for system drift are the point cloud quality, the SLAM minimization cost function, and the geometric distribution of matched feature points, which is related to the scene structure (e.g., coplanarity of feature points) and the local minimum of the ICP algorithm (Bose , Richards, 2016). Most RGB-D SLAM algorithms use visual loop closure to enhance the reconstructed 3D model of the environment; the loop closure concept can be applied locally (i.e., searching every 5 frames for a possible loop) or globally (i.e., closing the first and last frames). The widely used algorithm for detection and correction of the closure error was based on a graph–optimization technique (Kümmerle et al., 2011). Others researchers are using photogrammetric bundle adjustment to detect and correct both local and global loop closure (Melbouci et al., 2015).

In this study, we propose a new framework that uses all possible information from the surrounding indoor environment to precisely reconstruct a 3D model. The proposed framework uses not only visual two-dimensional (2D) feature points but also 3D features such as lines and planes. In addition, the framework overcomes the minimal problem of the ICP algorithm by using a descriptor for each selected 3D feature for further matching. After matching of 2D and 3D features, a new objective function is introduced to be minimized for precise construction of the 3D model from the captured RGB-D frames. If a loop closure constraint is present, a graph-optimization technique is used to correct the closure error and optimize the final camera pose.

The remainder of the manuscript is organized as follows. In Section 2, we overview the current RGB-D SLAM systems, and in Section 3, we describe our proposed framework. Section 4 presents the experimental results of the framework performance evaluation, and Section 5 presents our conclusions and directions for future work.

2. RELATED WORK

The earlier algorithm that deals with construction of 3D models from successive RGB-D frames is the Kinect Fusion system (Newcombe et al., 2011). The system mainly uses the depth information and an ICP algorithm to register successive RGB-D frames, whereas the visual information from the RGB images is used to color the final 3D model. Henry et al. (2012) proposed a basic visual RGB-D SLAM system that uses visual features, the ICP algorithm, and loop closure correction to estimate the camera pose. Many other features are added to enhance the RGB-D SLAM systems performance, starting from simple matched visual features (e.g., SIFT (Lowe et al., 1999, Lowe, 2004), SURF (Bay et al., 2008)) and applying the concept of structure from motion (SFM) (Koenderink , Van Doorn, 1991) to recover the relative transformation between each successive RGB-D frame. Dos Santos et al. (2016) used a disparity-based model with maximum stable color regions to estimate the relative movement between two successive RGB-D frames.

Regarding the visual RGB-D SLAM concept, many studies have explored the ability of integration of various algorithms and various methods to enhance RGB-D SLAM performance for specific applications (Stachniss et al., 2007). The application of a visual–based RGB-D SLAM algorithm with continuous searching for loop closure optimization between each key frame was introduced as DVO SLAM (Kerl et al., 2013) . Dai et al. (2017) designed a system that can handle an on–the–fly RGB-D SLAM system with implementation of sparse points and dense model optimization.

A plethora of SLAM algorithms have been published (Stachniss et al., 2007), each with specific performance based on the available observations, the optimization technique used, and the applications (e.g., surveying, robotics, indoor and outdoor navigation, looped environments). Fioraio and Konolige (2011) used bundle adjustment with ICP (Besl , McKay, 1992) and used graph optimization for final pose optimization. Endres et al. (2012) developed a system that first implements the visual matched feature with local loop closure using g2o (Kümmerle et al., 2011). The system gave precise results when tested in a small room with many distinctive visual features with near depth. The common limitation of these methods is that the operation distance should be less than three meters. When the matched features are farther from the camera (i.e., the depth is greater than three meters) and the scene lacks distinguishing visual details, the visual RGB-D SLAM system can easily fail.

Instead of depending on visual features, the edge RGB-D SLAM algorithm (Bose , Richards, 2016) introduces edge detection based on depth image with an ICP algorithm to compute the cameras relative pose; this method works well in 3D spaces with rich 3D edges even if they have less texture. The main constraint of this method is the mapping speed and local minimal problem of ICP, as the edges were extracted only without matching. Because the point cloud produced from depth images is noisy, especially for points at a depth of more than two meters, the classical method to detect, extract, and match 3D features (Diez et al., 2015) has not yet been implemented because the descriptor of the 3D feature is based on the 3D structure and is greatly affected by the depth noise. Hsiao et al. (2017) used the planar constraint to reduce the drift problem of the visual RGB-D SLAM system for a 30-Hz frame rate. To overcome the RGB-D depth precision problem of distant matched visual points, integration between SFM technique and RGB-D SLAM system was carried by several investigators (Concha , Civera, 2017, Dai et al., 2017, Kerl et al., 2013, Melbouci et al., 2015, Stückler , Behnke, 2012). Kerl et al. (2013) minimized both geometric (depth) and photometric (color) distances with the g2o algorithm as a global optimization container to recover both the 3D model and the camera pose, and the system achieved a 3-cm error (mean error of camera trajectory) compared with a 4-cm error for MRSMap (Stückler , Behnke, 2012).

Other studies have concentrated on offline enhancement of the RGB-D SLAM performance, which has mainly been used for surveying applications and 3D model reconstruction. Halber and Funkhouser (2017) introduced an offline constrained method to refine the global 3D model by manually extracting the models predefined geometry (e.g. orthogonality, parallelism). The system can work for large spaces, but it must have previous knowledge of the environments structure because the system uses planar constraints iteratively to refine the global 3D model. Tang et al. (2016) integrated SFM with visual RGB-D SLAM concepts to produce a complete 3D model for near and far range; to some extent, this method can be applied in outdoor environments.

In this study, an offline framework of a RGB-D SLAM system is proposed for continuous mapping of indoor spaces. The proposed framework considers all possible constraints in an indoor environment. We first use the point features-based method to estimate the initial relative pose and then extract the line features from both RGB and depth images based on both normal and depth information. After line feature detection, we extract, describe, and match the feature lines, and at the same time we extract the existing 3D planes based on RANSAC. Plane extraction, selection, description, and matching are all applied. The proposed framework uses all possible visual features (SIFT points, visual Hough line) and 3D features (lines based on depth, lines based on normal, and planar objects) with a crucial description and matching functions to overcome the minimal problem of ICP.

3. THE PROPOSED FRAMEWORK

The proposed framework considers all possible features in both RGB and depth images to precisely reconstruct a 3D model of the indoor environment. Figure 1 shows the proposed framework, which consists of five major threads. The calibration thread, which is mandatory to improve the depth precision and to eliminate lens distortion. The feature extraction thread which focuses mainly on extraction of features from both 2D space (RGB) and 3D space (point cloud). The feature description thread which deals with description of the nominated features for further matching. Before the environment constraints thread, in case of loop closure correction, the final thread deals with the tracking algorithm, which keeps the RGB-D camera in the same framework.

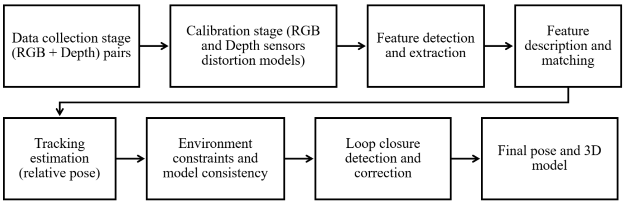

Data collection stage (RGB + Depth) pairs
Calibration stage (RGB and Depth sensors distortion models)
Feature detection and extraction
Feature description and matching
Tracking estimation (relative pose)
Environment constraints and model consistency
Loop closure detection and correction
Final pose and 3D model

Figure 1. The proposed framework to reconstruct 3D models from RGB-D captured frames

In the first thread, we developed a mobile application to capture both processed and raw data for further offline processing. In the second thread, we adopted a method (Darwish et al., 2017) to calibrate all captured data. In the third and fourth threads, 2D and 3D features were extracted, described, and matched. In the fifth thread, a general cost function that minimizes the geometric distances of all matched features was applied to estimate the relative camera transformation between successive frames. In the sixth thread, the environment constraints stage used perpendicular and parallelism constraints to optimize the 3D model. The seventh thread presents loop closure detection and correction, for which we adopted the method presented by (Kümmerle et al., 2011). The final thread presents the final output products for pose information and 3D modeling.

The following subsections describe in detail the aforementioned threads except for the calibration strategy; more details about RGB-D calibration and the graph-based optimization technique can be found in the literature (Chow , Lichti, 2013, Herrera et al., 2012, Lachat et al., 2015, Mallick et al., 2014, Pagliari , Pinto, 2015, Raposo et al., 2013, Zhang , Zhang, 2014). The major contributions of this study include the stages of feature description and matching; tracking estimation; and environment constraints optimization.

3.1 Feature Detection and Extraction

The proposed framework depends on the extracted features from both RGB and depth images. The extracted features are divided into two categories. The first category is presented as 2D features, which contain SIFT feature points and edges from the RGB images. The second category, which is assumed to include 3D features, contains edges and planes extracted from depth images. The extracted edges are based on either the normal difference or the depth difference. For 3D planes, the RANSAC method is used to extract all possible planes from the depth images. After the 2D and 3D features are detected, the nomination criterion is proposed to select the distinctive features. For 2D features, a color gradient based on a Hessian matrix (Bay, 2006) is applied with a certain threshold to define the nominated feature points (Bay et al., 2008, Cornelis , Van Gool, 2008). A new nomination criterion is proposed for each 3D feature type (i.e., lines or planes). The nomination method of the line features is based on length; lines whose length exceeds a certain threshold are nominated as a feature. For the 3D plane feature, a nominated plane is based on the number of inliers and its distance to nearby planes. The plane that has the largest number of inlier points within a certain threshold is nominated as a plane feature. Equations 1, 2, and 3 show the formulae used to identify the nominated planes among the detected (m) planes. Assuming that the depth image converted to point cloud (P) and (m) planes (PL) were extracted with the RANSAC method, for each recognized plane (PL_i) and for a certain distance threshold (t_{pts}), the number of inliers for each plane was computed, and the planes were sorted in descending order using (1). Equation 2 represents the distance between a pair of detected planes. Finally, equation 3 is a filtering formula that returns only the nominated planes $(PL_{norm}\ )$ among the detected planes (PL).

$$I = \Omega\Big(\sum_{i=1}^{m} \phi\big(PL_i, P\big) \leq t_{pts}\Big) \tag{1}$$

where PL_i = parameters define the i^{th} plane
P = point cloud generated from depth image
t_{pts} = distance threshold that defines the point outliers
ϕ = function computes the orthogonal distance between points P and plane PL_i
Ω = function sorts the detected planes PL based on number of inliers
I = indices of sorted planes based on point inliers
m = the total number of detected planes

$$R_{plane} = \frac{\sum_{i=1}^{m-1} \omega\big(PL_{I(i)}, PLc_{I(i+1)}\big) \geq t_{pls}}{size\big(PLc_{I(i+1)}\big)} \tag{2}$$

where PL_i = parameters define the i^{th} plane
$PLc_{I(i+1)}$ = point cloud defines $I(i+1)^{th}$ plane
$size(PLc_{I(i+1)})$ = number of points defined $I(i+1)^{th}$ plane
t_{pls} = the distance threshold to filter out the identical planes
ω = the function returns the orthogonal distance between two planes points
R_{plane} = the percentage of points of the $I(i)^{th}$ plane that lie outside the $I(i+1)^{th}$ plane

within distance t_{pls}
I = indices of sorted planes based on point inliers
m = the total number of detected planes

$$PL_{nom} = PL\Big(1, R_{plane} \geq t_{nom}\Big) \tag{3}$$

where R_{plane} = the percentage of points of the $I(i)^{th}$ plane that lie outside the $I(i+1)^{th}$ plane
PL = cell array contains parameters that define all detected planes
t_{nom} = the threshold defined the overlap between two planes
PL_{nom} = the nominated planes parameters

3.2 Feature Description and Matching

After the distinctive 2D and 3D features are detected and extracted, the description of each feature is applied for further matching to remove outliers. In case of 2D features, a SIFT descriptor based on the color gradient is used with correction of scale and orientation of each nominated feature point (i.e., the SIFT descriptor length is 64 bits), whereas in the case of 3D features, the description vector of each selected feature is based on the Euclidean distances between the 3D feature and the 3D position of the matched SIFT points. The descriptor of the 3D feature length depends on the number of matched SIFT points between two successive frames. Equations 4 and 5 show the descriptors of 2D and 3D features, respectively. Because the matching between nominated features is based on the descriptors, the 2D feature is matched based on the sum of squared differences (SSD) between each pair of descriptors, and the best match corresponds to the minimal of SSD (Cornelis , Van Gool, 2008). 3D features are matched based on the normalized Pearsons cross correlation concept. Equations 6 and 7 present the matching concept between 2D features and 3D features, respectively.

$$D_{2d} = \cup_{block=1}^{block=16}\Big(\textstyle\sum_{i=1}^{i=4} dx_i, \sum_{i=1}^{i=4}|dx_i|, \sum_{i=1}^{i=4} dy_i, \sum_{i=1}^{i=4}|dy_i|\Big) \tag{4}$$

where D_{2d} = the descriptor of 2D feature image point
dx_i = the image gradient of sub block (i) along x direction
dy_i = the image gradient of sub block (i) along y direction

Normally, the SIFT descriptor uses a 4×4-pixel sub block size with a global block of 4×4 sub blocks, which means that the descriptor has a vector length of 64.

$$D_{3d} = \cup_{k\in m}^{j\in n}\|F_j - P_k\|^2 \tag{5}$$

where D_{3d} = the descriptor of 3D feature, presented as a line extracted from the RGB image and projected back to the point cloud, directly extracted from the depth image based on normal, or presented as a plane directly extracted from the point cloud
F_j = the 3D feature information, a line uses two points and a plane uses three points
P_k = the coordinate of the projected matched SIFT point to the 3D point cloud
m, n = are the total numbers of matched SIFT points and extracted 3D features, respectively

Matching between both 2D sets of descriptors can be carried out by SSD, and the pair of matched features reflects the minimum value of SSD. Equation 6 represents the SSD method.

$$SSD_{f_1,f_2} = \sum_{i=1}^{i=64}\Big(D_{f_1}(i) - D_{f_2}(i)\Big)^2 \tag{6}$$

where f_1,f_2 = point features that exist in the first and second images, respectively.
SSD_{f_1,f_2} = the sum of squared difference distances between point features.
D_{f_1}, D_{f_2} = the descriptor of point features located on first image and second image, respectively.

$$S_{ik} = \frac{cov\big(D_i, D_k\big)}{\sigma_{d_i}\sigma_{d_k}} \tag{7}$$

where S_{ik} = matching score between descriptor i and k
cov = covariance between descriptor i and k
D_i, D_k = descriptor of 3D feature i and k, respectively
σ_{d_i}, σ_{d_k} = descriptor standard deviation of 3D feature i and k, respectively

After illustrating all possible features from the depth and RGB images (i.e., the matched SIFT points and the 3D features all collaborate to compute the relative pose between two successive RGB-D frames), the following section discusses comprehensively the proposed tracking algorithm used to update the camera pose for the proposed framework.

3.3 Tracking Estimation

Computing the relative movement between two captured RGB-D frames is a crucial stage for continuous tracking of RGB-D cameras. The visual RGB-D SLAM system minimizes the geometric distance of the corresponding SIFT matched points between RGB-D frames to recover the cameras relative pose. The proposed framework mainly uses geometric information to compute the relative pose between RGB-D frames. Three different pieces of information are extracted from the successive RGB-D frames: the matched 3D points from the point cloud corresponding to the point features from RGB images; the line features extracted from both the RGB and the depth information; and the planes extracted from the point cloud.

Annotations: For each extracted feature type, we adopt different representation; thus, different formulae were introduced to compute the relative movement between two RGB-D frames depending on the feature type. It is assumed that the scene has (m) matched points, (n) matched lines, and (q) matched planes. For the corresponding 3D points, we present P_1 and P_2 as two matrices of dimension of m×3, with each containing all point information, and each row has $[X_iY_iZ_i]$. For the extracted lines, we present L_1 and L_2 as two matrices of dimension of n×6, with each containing the line information as $[XC_{li}YC_{li}ZC_{li}XD_{li}YD_{li}ZD_{li}]$, where the first three elements refer to the coordinates of the center point of the line (we choose the nearest matched SIFT point to the line and compute its projected coordinate to the line) and the next three elements refer to the direction vector of the extracted line. For the extracted planes, we present PL_1 and PL_2 as two matrices with dimension of $q \times 6$, where each row represents the plane information as $[XC_{ni}YC_{ni}ZC_{ni}Nx_{ni}Ny_{ni}Nz_{ni}]$ and the first three elements refer to the center point of the plane. We use the same concept of line to detect such points, and the next three elements refer to the normal vector of the matched plane.

Three geometric quantities should be minimized during the pose estimation process: E_p is the back-projection error of the matched 3D point features between the RGB-D frames (8), E_l is the residual vector between matched lines (9), and E_n is the vector of residuals between matched planes (10). It is assumed that R and T are the rotation and translation of the rigid relative transformation between two RGB-D frames, respectively.

$$E_p = \|RP_1 + T - P_2\|^2 \tag{8}$$

where P_1, P_2 = $m \times 3$ matrix containing coordinates of all matched points for RGB-D frame 1 and 2, respectively.

$$E_l = \|(RC_{l1} + T)^t.f(D_2) - f(D_2).C_{l2}\|^2 \tag{9}$$

where f = function converts the direction vector to a normal vector.
C_{l1}, C_{l2} = matched lines center point of RGB-D frame 1 and 2, respectively
D_1, D_2 = direction vectors of matched lines between RGB-D frame 1 and 2, respectively.

$$E_l = \|(RC_{n1} + T)^t.N_2 - N_2.C_{n2}\|^2 \tag{10}$$

where C_{n1}, C_{n2} = matched planes center point coordinates for RGB-D frame 1 and 2, respectively.
N_1, N_2 = normal vectors of matched planes between RGB-D frame 1 and 2, respectively.

The tracking estimation of the RGB-D camera can be represented as in (11); as from the geometry principles, lines and planes introduce more constraints on rotation rather than the translation opposite to the point features, one can apply different weight for each objective function stated in (8), (9), and (10). We keep this weighting factor as a future work which can be examined.

$$\{\hat{R}, \hat{T}\} = agrmin\Big(E_p + E_l + E_n\Big) \tag{11}$$

where $\hat{R}, \hat{T}$ = estimated camera rotation and translation, respectively.
E_p, E_l, E_n = the point, line, and plane features re-projection error, respectively.

The system first begins to initialize the pose between two RGB-D frames by adopting the principal concept of visual RGB-D SLAM and the point feature correspondences used to calculate the relative pose. This information is then used to match all extracted planes and lines extracted from both depth and RGB images. The final pose is calculated with (11). After optimizing the pose information between successive RGB-D frames, the environment constraints stage is introduced to further smooth the reconstructed 3D models.

3.4 Environment Constraints

The environment constraints stage is the final refinement process before the loop closure concept (if any) is applied. In this stage, the spatial relations (i.e., perpendiculars, parallels) are basically generated from the camera pose to roughly determine the 3D shape constraints. Thus, the global model gM is divided into separate sub models sM for further accurate alignment. The environment constraints stage is based on the planar objects between the successive sub models. The structure sensor coordinate system is defined as shown in figure 2.

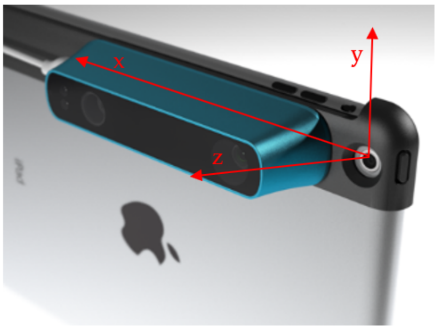


Figure 2. Structure sensor coordinate system.

Using the coordinate system information, the rotation around y direction (θ_y) indicates planar movement constraints in the 2D floor plan. The structure sensor is normally attached to an iPad, so the rotation around the z and x directions is around 15 degrees. In an indoor environment, the dominant axis that controls the scanning process is the y-axis. Equation 12 shows the formula used to detect the turned frames (the most likely valuable frame that contain the constrained structures) from the y-axis rotation, after which equation (13) is used to construct the sub models for further alignment.

$$N = \left[1, PE\Big(G\Big(\frac{\partial \theta_y}{\partial y}\Big)\Big)\right] \tag{12}$$

where N = the IDs of the turned RGB-D frames
PE = the function that detects the peaks from time series
G = Gaussian filter function to smooth the gradient of y-axis rotation
θ_y = rotation angle around the y-axis

$$sM(i) = gM\Big(N(i) : N(i+1)\Big) \tag{13}$$

where N = the IDs of the turned RGB-D frames
gM = global model to be smoothed
sM = sub models divided based on the turned frames indices N

Once the sub models were constructed, the environment constraints stage is carried out based on the previously known spatial relationship between each sub model. It is assumed that the spatial relationships were stored in S, which contained the turn angles around the three axes. S is reconstructed based on the planar relation between two successive sub model, perpendiculars, parallels, and artificially defined angles (e.g., $\pi/4$,$\pi/8$, $3\pi/4$). The environment constraints stage deals with enforcing back these artificial angles. Equation (14) presents the formula used in the environment refinement stage.

$$\{rpose, rM\} = itercon\big(sM, S\big) \tag{14}$$

where $rpose$ = the refined camera poses after constrained stage
rM = the refined model after refinement stage
S = spatial constrained information
$itercon$ = constrained function iteratively enforces predefined spatial information S

After estimating and refining the relative pose between successive RGB-D frames, detection and correction of loop closure, if any, was performed.

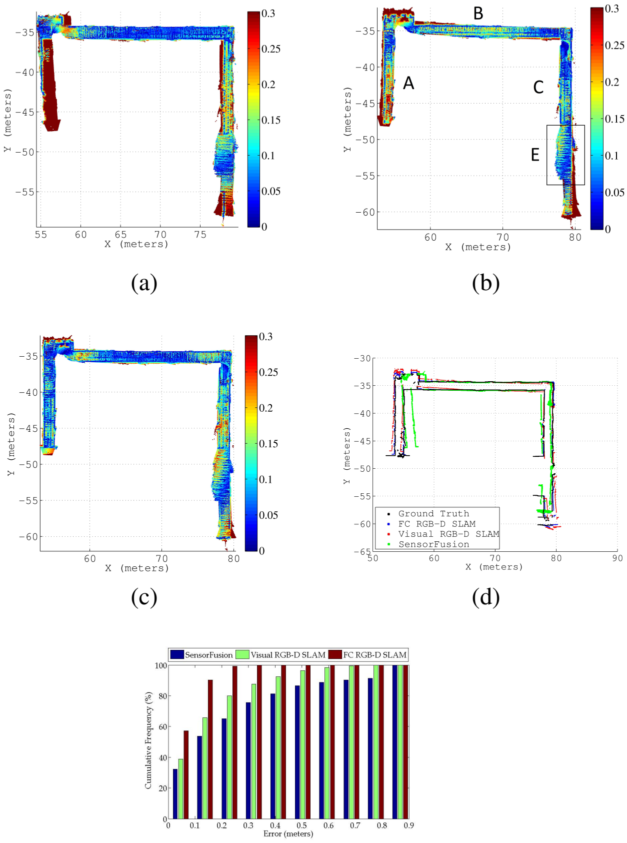
3.5 Loop Closure

Loop closure is a basic concept for correcting a closed mapped space; the reputed method is the graph optimization technique based on nonlinear least-squares optimization (Kümmerle et al., 2011). This method constructs a graph problem based on nodes and edges, where each node represents the pose information of each RGB-D frame and the edges represent the 6DoF relative baseline between two successive RGB-D frames. The algorithm was adopted for most of the previous RGB-D SLAM algorithm and reflected stable results; thus, we adopt the loop closure problem in the proposed framework.

4. EXPERIMENTS AND DISCUSSIONS

The experiments were used to evaluate the performance of the proposed framework against other RGB-D SLAM systems (e.g., offline visual RGB-D SLAM (Tang et al., 2016), SensorFusion). For simplifying the figures representations, the proposed framework here and after will be noted as Fully Constrained RGB-D SLAM (FC RGB-D SLAM), however it is an offline process pipeline. FC RGB-D SLAM is designated to precisely reconstruct 3D models of long indoor corridors with few distinctive features. The data were captured with a Structure Sensor attached to an iPad Air 2, which is used to capture and process the data. The Structure Sensor has a framework (SDK; SensorFusion) to process the captured depth images, color images, and IMU data from the iPad to create a 3D model of the captured environment. The proposed framework was compared to both visual RGB-D SLAM (Tang et al., 2016) and SensorFusion methods. Data were captured for a narrow corridor, 58 m long, 2.5 m high, and 1.5 m wide. A laser scanner was used to capture the ground truth of the corridor for further quantitative evaluation.

The accuracy assessment is based on the model quality of the point difference. Each model was compared to the ground truth. The error of each point was quantified, and the histogram of each model was used to check the model accuracy. Figure 3 shows the quantitative results for the three RGB-D SLAM systems. It can be clearly seen that the proposed framework can enhance the overall model accuracy and its alignment. The model is divided to three patches (A, B, and C) that are completely perpendicular, as shown in figure 3(b). At the corner, the angles are perfect right angles in the reconstruction model using the proposed framework, but they are not right angles with either the visual RGB-D SLAM or SensorFusion methods; however, the visual RGB-D SLAM system achieves better accuracy than the SensorFusion system. Parts A and C show a severe drift in both the SensorFusion and visual RGB-D SLAM results, whereas the drift was significantly reduced in the FC RGB-D SLAM result. The largest error was present in the corner between A and B, possibly because of the material of the wall; because this part of the corridor is a glass window, the depth result from the RGB-D camera includes a lot of noise. The error at the end of part C is due to the structure of area E (highlighted by a black line). This area lacks 2D and 3D features because it is an open space.



(a) (b) (c) (d) (e)

Figure 3. Spatial error distributions for a scanned corridor in vertical scanning mode: (a) model from SensorFusion; (b) model from visual RGB-D SLAM; (c) model from FC RGB-D SLAM; (d) Projected wall to the ground of four models (black is ground truth, blue is FC RGB-D SLAM, red is visual RGB-D SLAM, and green is SensorFusion); (e) Error histogram of the three systems

To validate the proposed framework, three more data sets were captured using the same sensor with different data collection procedures. One set was obtained with the iPad horizontal, another set was obtained with the iPad horizontal and with a low frame rate (5 fps), and the third set was obtained with the iPad horizontal and facing the ground (uncaptured ceiling). Figure 4 presents a histogram of the average cumulative error to summarize the results of the four experiments. It can be clearly seen that the error of 95% of points does not exceeded 0.20 m for the FC RGB-D SLAM compared to 1.00 m and 1.20

m for the visual RGB-D SLAM and SensorFusion methods, respectively.

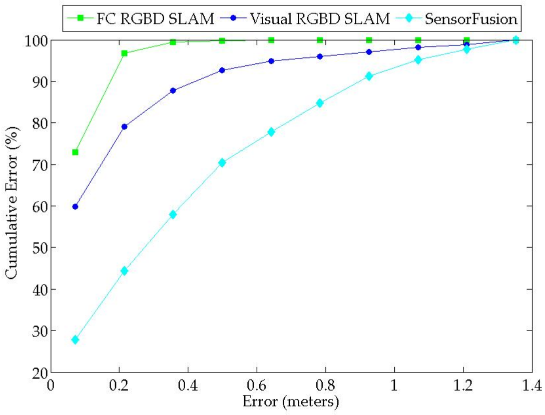


Figure 4. Average cumulative error histogram for all captured experiments

5. CONCLUSIONS AND FUTURE WORK

The use of RGB-D cameras as low-cost depth sensors to reconstruct indoor 3D models has attracted great attention in various research areas (e.g., surveying, robotics, computer vision). Two major research problems face the use of this fruitful technology in precise surveying applications (i.e., centimeter-level precision applications). The first problem is the quality of the depth information produced by RGB-D cameras, and the second is the lost tracking, which can produce the full 3D model from the captured RGB-D frames. This research focuses on the second research problem; thus, a new framework is proposed to achieve 3D models with centimeter-level precision from captured RGB-D frames. The proposed framework adopts both 2D and 3D features to obtain the RGB-D camera pose as the first stage. In the second stage, the existing constraints between the 3D features are used to refine the reconstructed 3D model. The proposed framework was compared to visual SLAM and SensorFusion systems to evaluate its performance. The results demonstrate the usefulness of the proposed framework.

The nomination of matched point features from RGB images can be applied based on a suitable stochastic analysis of matched SIFT points to enhance the SLAM performance, especially if 3D features are missing from some frames. An extended SLAM may include an object extracted from RGB-D images to enhance both modeling and tracking performance. This could be a new way to integrate both computer vision applications (i.e., RGB-D object recognition) and feature-based RGB-D SLAM. The revolution of machine learning applications could facilitate the 3D feature extraction and matching process, as the neural network can effectively predict the descriptor of the 3D feature from a noisy point cloud.

ACKNOWLEDGEMENTS

The work described in this paper was substantially supported by a grant from The National Key Research and Development Program of China (No. 2016YFB0502101).

REFERENCES

Bae, Kwang-Ho, Lichti, Derek D, 2008. A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63, 36–54.

Bay, Herbert, 2006. From wide-baseline point and line correspondences to 3D. PhD thesis, ETH Zurich.

Bay, Herbert, Ess, Andreas, Tuytelaars, Tinne, Gool, Luc Van, 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110, 346 - 359. Similarity Matching in Computer Vision and Multimedia.

Besl, Paul J, McKay, Neil D, 1992. Method for registration of 3-d shapes. *Sensor Fusion IV: Control Paradigms and Data Structures*, 1611, International Society for Optics and Photonics, 586–607.

Bose, Laurie, Richards, Arthur, 2016. Fast depth edge detection and edge based rgb-d slam. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1323–1330.

Chow, Jacky CK, Lichti, Derek D, 2013. Photogrammetric bundle adjustment with self-calibration of the PrimeSense 3D camera technology: Microsoft Kinect. *IEEE Access*, 1, 465–474.

Concha, Alejo, Civera, Javier, 2017. Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 6756–6763.

Cornelis, N., Van Gool, L., 2008. Fast scale invariant feature detection and matching on programmable graphics hardware. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8.

Dai, Angela, Nießner, Matthias, Zollhöfer, Michael, Izadi, Shahram, Theobalt, Christian, 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36, 76a.

Darwish, Walid, Tang, Shenjun, Li, Wenbin, Chen, Wu, 2017. A new calibration method for commercial RGB-D sensors. *Sensors*, 17, 1204.

Diez, Yago, Roure, Ferran, Lladó, Xavier, Salvi, Joaquim, 2015. A qualitative review on 3D coarse registration methods. *ACM Computing Surveys (CSUR)*, 47, 45.

Dos Santos, Daniel R, Basso, Marcos A, Khoshelham, Kourosh, de Oliveira, Elizeu, Pavan, Nadisson L, Vosselman, George, 2016. Mapping indoor spaces by adaptive coarse-to-fine registration of RGB-D data. *IEEE geoscience and remote sensing letters*, 13, 262–266.

Endres, Felix, Hess, Jürgen, Engelhard, Nikolas, Sturm, Jürgen, Cremers, Daniel, Burgard, Wolfram, 2012. An evaluation of the rgb-d slam system. *Icra*, 1691–1696.

Endres, Felix, Hess, Jürgen, Sturm, Jürgen, Cremers, Daniel, Burgard, Wolfram, 2014. 3-D mapping with an RGB-D camera. *IEEE transactions on robotics*, 30, 177–187.

Fioraio, Nicola, Konolige, Kurt, 2011. Realtime visual and point cloud slam. *Proc. of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems Conf.(RSS)*, 27.

Fischler, Martin A, Bolles, Robert C, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 381–395.

Halber, Maciej, Funkhouser, Thomas, 2017. Fine-to-coarse global registration of rgb-d scans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1755–1764.

Henry, Peter, Krainin, Michael, Herbst, Evan, Ren, Xiaofeng, Fox, Dieter, 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31, 647–663.

Herrera, Daniel, Kannala, Juho, Heikkilä, Janne, 2011. Accurate and practical calibration of a depth and color camera pair. *International Conference on Computer analysis of images and patterns*, Springer, 437–445.

Herrera, Daniel, Kannala, Juho, Heikkilä, Janne, 2012. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 2058–2064.

Hsiao, Ming, Westman, Eric, Zhang, Guofeng, Kaess, Michael, 2017. Keyframe-based dense planar slam. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5110–5117.

Huang, Albert S, Bachrach, Abraham, Henry, Peter, Krainin, Michael, Maturana, Daniel, Fox, Dieter, Roy, Nicholas, 2017. Visual odometry and mapping for autonomous flight using an rgb-d camera. *Robotics Research*, Springer, 235–252.

Kerl, Christian, Sturm, Jürgen, Cremers, Daniel, 2013. Dense visual slam for rgb-d cameras. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2100–2106.

Koenderink, Jan J, Van Doorn, Andrea J, 1991. Affine structure from motion. *JOSA A*, 8, 377–385.

Kümmerle, Rainer, Grisetti, Giorgio, Strasdat, Hauke, Konolige, Kurt, Burgard, Wolfram, 2011. g 2 o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation*, IEEE, 3607–3613.

Lachat, Elise, Macher, Hlne, Landes, Tania, Grussenmeyer, Pierre, 2015. Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling. *Remote Sensing*, 7, 13070–13097. http://www.mdpi.com/2072-4292/7/10/13070.

Lowe, David G, 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110.

Lowe, David G et al., 1999. Object recognition from local scale-invariant features. *iccv*, 99number 2, 1150–1157.

Mallick, T., Das, P. P., Majumdar, A. K., 2014. Characterizations of Noise in Kinect Depth Images: A Review. *IEEE Sensors Journal*, 14, 1731-1740.

Melbouci, Kathia, Collette, Sylvie Naudet, Gay-Bellile, Vincent, Ait-Aider, Omar, Carrier, Mathieu, Dhome, Michel, 2015. Bundle adjustment revisited for slam with rgbd sensors. *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, IEEE, 166–169.

Newcombe, Richard A, Izadi, Shahram, Hilliges, Otmar, Molyneaux, David, Kim, David, Davison, Andrew J, Kohi, Pushmeet, Shotton, Jamie, Hodges, Steve, Fitzgibbon, Andrew, 2011. Kinectfusion: Real-time dense surface mapping and tracking. *2011 IEEE International Symposium on Mixed and Augmented Reality*, IEEE, 127–136.

Pagliari, Diana, Pinto, Livio, 2015. Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors. *Sensors*, 15, 27569–27589.

Raposo, Carolina, Barreto, Joao Pedro, Nunes, Urbano, 2013. Fast and accurate calibration of a kinect sensor. *2013 International Conference on 3D Vision-3DV 2013*, IEEE, 342–349.

Rusinkiewicz, Szymon, Levoy, Marc, 2001. Efficient variants of the icp algorithm. *3dim*, IEEE, 145.

Stachniss, Cyrill, Frese, Udo, Grisetti, Giorgio, 2007. OpenSLAM. *URL: http://www. openslam. org*.

Stückler, Jörg, Behnke, Sven, 2012. Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras. *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, IEEE, 162–167.

Tang, Shengjun, Zhu, Qing, Chen, Wu, Darwish, Walid, Wu, Bo, Hu, Han, Chen, Min, 2016. Enhanced RGB-D mapping method for detailed 3D indoor and outdoor modeling. *Sensors*, 16, 1589.

Tsai, J, Chiang, KW, Chu, CH, Chen, YL, El-Sheimy, N, Habib, A, 2015. THE PERFORMANCE ANALYSIS OF AN INDOOR MOBILE MAPPING SYSTEM WITH RGB-D SENSOR. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 40.

Zhang, Cha, Zhang, Zhengyou, 2014. Calibration between depth and color sensors for commodity depth cameras. *Computer Vision and Machine Learning with RGB-D Sensors*, Springer, 47–64.