2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

MonoRGBD-SLAM: Simultaneous Localization and Mapping Using Both Monocular and RGBD Cameras

Khalid Yousif[1], Yuichi Taguchi[2], and Srikumar Ramalingam[2]

***Abstract*— RGBD SLAM systems have shown impressive results, but the limited field of view (FOV) and depth range of typical RGBD cameras still cause problems for registering distant frames. Monocular SLAM systems, in contrast, can exploit wide-angle cameras and do not have the depth range limitation, but are unstable for textureless scenes. We present a SLAM system that uses both an RGBD camera and a wide-angle monocular camera for combining the advantages of the two sensors. Our system extracts 3D point features from RGBD frames and 2D point features from monocular frames, which are used to perform both RGBD-to-RGBD and RGBD-to-monocular registration. To compensate for different FOV and resolution of the cameras, we generate multiple virtual images for each wide-angle monocular image and use the feature descriptors computed on the virtual images to perform the RGBD-to-monocular matches. To compute the poses of the frames, we construct a graph where nodes represent RGBD and monocular frames and edges denote the pairwise registration results between the nodes. We compute the global poses of the nodes by first finding the minimum spanning trees (MSTs) of the graph and then pruning edges that have inconsistent poses due to possible mismatches using the MST result. We finally run bundle adjustment on the graph using all the consistent edges. Experimental results show that our system registers a larger number of frames than using only an RGBD camera, leading to larger-scale 3D reconstruction.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a method used for simultaneously estimating the pose of a camera and reconstructing a map of its surrounding environment. SLAM has been widely studied over the past decades and many methods have been proposed in robotics, computer vision, and augmented reality communities. Those methods have utilized various types of sensors such as laser scanners, monocular cameras, and stereo cameras. Recently, there has been a wealth of interest in using RGB-Depth (RGBD) sensors for solving the SLAM problem, mainly due to the appearance of Kinect. In addition to providing color information, the Kinect uses a structured light approach for finding the depth information in a scene (up to a certain depth limit). Providing color and depth information, in addition to the affordable price, made the Kinect an attractive and viable sensor.

Despite the advances in SLAM research, the problem still remains challenging. Some of those problems could be attributed to the type and quality of the sensor, whereas other problems may occur due to the environment that is being mapped such as containing limited texture or structure. We focus on the first of the two aforementioned problems: the type of sensor that is used to solve the SLAM problem. For instance, the Kinect has a number of positives as mentioned above, but the main disadvantages are the limited depth range, the relatively low resolution (640 × 480), and the narrow field of view (FOV) (60° in the horizontal axis). The narrow FOV makes registering frames more challenging, compared to laser scanners which can measure up to 360°. This is particularly evident when turning around corners, since an overlap between two frames is essential in order to extract and match the same features observed in those frames. Having a narrow FOV allows the required features to move out of the image faster, which may result in registration failure due to not having enough correspondences.

On the other hand, monocular cameras have an advantage over Kinect-style sensors in that it does not have a limited depth range. In addition, numerous types of monocular cameras with varying specifications are available and easily accessible. The main disadvantage of using monocular cameras for registration is that the generated maps are typically estimated up to scale, which can be determined by post or prior measurements. Unfortunately the scale consistency is hard to maintain and scale drift is unavoidable when mapping large scale environments, requiring careful handling [1]. Another disadvantage of monocular-based registration is the lack of 3D information. As such, registration is likely to fail when registering scenes containing limited textures. In contrast, RGBD sensors can fallback on methods that use the geometry of the scenes, such as the iterative closest point (ICP) algorithm [2].

In this paper, we propose a method that fuses the information provided by both an RGBD camera such as Kinect and a wide-angle monocular camera such as GoPro. We compensate for the weaknesses in each sensor by using the strengths from the other sensor. For instance, we overcome the scale ambiguity problem and the sparse nature of monocular SLAM using the metric depth information provided by the RGBD camera. In addition, we overcome the narrow FOV problem of the RGBD camera by making use of the wide-angle monocular camera (GoPro has a horizontal FOV of 120°). Our goal is to build a system that is able to handle challenging large-scale indoor sequences where a method using the aforementioned sensors individually would fail.

Figure 1 shows such an example, where RGBD-to-RGBD registration failed due to the limited textures in the scene and the narrow FOV of the RGBD camera. In contrast, RGBD-to-

[1]Khalid Yousif is with school of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, VIC 3083, Australia s3362555@student.rmit.edu.au

[2]Yuichi Taguchi and Srikumar Ramalingam are with Mitsubishi Electric Research Labs (MERL), Cambridge, MA 02139, USA {taguchi,ramalingam}@merl.com

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

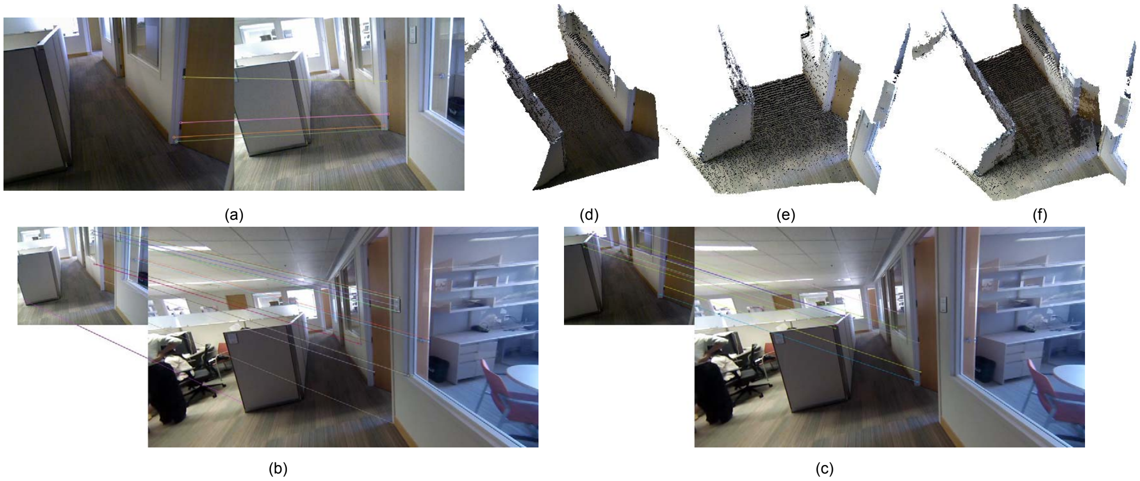


Fig. 1. An example scene demonstrating the advantage of using both RGBD-to-RGBD and RGBD-to-monocular registration. (a) For these two RGBD frames, RGBD-to-RGBD registration failed due to the limited textures and the narrow FOV, leading to an insufficient number of inliers visualized on the frames. (b, c) Each of the RGBD frames was successfully registered to a single monocular frame using RGBD-to-monocular registration by exploiting the wide FOV of the monocular camera, resulting in sufficient numbers of inliers. (d, e) The point clouds of the two RGBD frames, which are disconnected if only the RGBD-to-RGBD registration is used. (f) The two point clouds registered with each other due to the use of the monocular image.

monocular registration was still successful due to the wide FOV of the monocular camera. Our system exploits both of these RGBD-to-RGBD and RGBD-to-monocular matches by using 3D-to-3D and 3D-to-2D RANSAC registration. For obtaining accurate RGBD-to-monocular matches, we propose to generate multiple virtual images from each wide-angle monocular image by using the intrinsic parameters of the RGBD camera. Computing feature descriptors on the virtual images improves both feature matching and loop closure detection results. To compute the poses of the frames, we construct a graph where nodes represent the RGBD and monocular frames and edges denote the pairwise registration results. We compute minimum spanning trees (MSTs) to obtain initial pose estimates, which are used to prune incorrect edges due to mismatches. We then run bundle adjustment (BA) on the graph to refine the poses.

A. *Contributions*

The following list summarizes our main contributions.

- We present a SLAM system that fuses information from both a monocular camera and an RGBD camera.
- We propose to generate multiple virtual images from each wide-angle monocular image for improving feature matching and loop closure detection.
- We present an MST-based algorithm for connecting the frames and finding a good initial solution, which is later refined by BA.

B. *Related Work*

Monocular SLAM: A body of related work exists in the field of monocular SLAM, also known as structure from motion. Davison *et al.* [3] proposed one of the first extended Kalman filter (EKF) based monocular SLAM solutions. They constructed a map by extracting sparse features of the environment using Shi and Tomasi operator [4] and matched new features to those already observed using a normalized cross-correlation. Since an EKF was used for state estimation, only a limited number of features were extracted and tracked in order to manage the high computational cost of the EKF. PTAM is another well known method proposed by Klein and Murray [5], in which they pioneered the idea of running camera tracking and mapping in parallel threads. Unlike Davison *et al.*'s filtering based method, PTAM was optimization based and utilized BA for the estimation of its parameters. Despite its success, PTAM had several limitations, such as the restriction to map small environments, the lack of a large loop closure detection system, and the low invariance to viewpoint change since it is based on the correlation between low resolution images of the keyframes. Both of the aforementioned methods are feature based, as they rely on extracting and tracking a sparse set of salient image features. Most recently, due to the increase in computational capability, direct methods such as DTAM [6] and LSD-SLAM [1] have been proposed. The direct methods exploit every pixel in the image to produce an estimate of the camera pose relative to a 3D map, but are still unstable in scenes with limited textures, common in indoor environments.

RGBD SLAM: Henry *et al.* [7] developed one of the first methods in which an RGBD camera was employed to capture the scenes of an indoor environment and obtain a 3D map. In their implementation, FAST [8] features were extracted and matched between sequential frames using local image descriptors. RANSAC was then used to find the inliers and estimate the camera pose. This estimate was then refined using an ICP algorithm [2]. Loop closures were detected by matching the current frame to previously collected keyframes and BA was utilized in order to achieve global consistency. Audras *et al.* [10] presented an optical flow based direct RGBD SLAM method that does not rely on the feature extraction and matching steps. Newcombe *et*

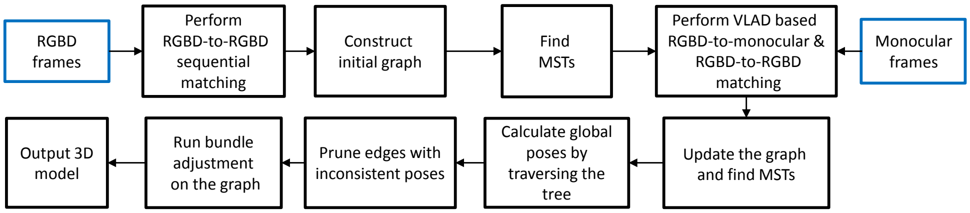

RGBD frames
Perform RGBD-to-RGBD sequential matching
Construct initial graph
Find MSTs
Perform VLAD based RGBD-to-monocular & RGBD-to-RGBD matching
Monocular frames
Update the graph and find MSTs
Calculate global poses by traversing the tree
Prune edges with inconsistent poses
Run bundle adjustment on the graph
Output 3D model

Fig. 2. Overview of our system.

al. [11] proposed KinectFusion, a GPU-based 3D mapping approach that employs a truncated signed distance function (TSDF) to represent the scene geometry. They used an ICP variant to match the current frame to a full growing surface model instead of matching sequential frames, resulting in more accurate registration. The original KinectFusion was limited to reconstruct a small fixed volume due to the limited GPU memory, which has been addressed by several recent extensions [12], [13], [14] to reconstruct large-scale scenes.

Monocular-RGBD SLAM: Hu *et al.* [16] addressed the problem of not having sufficient depth information in large areas due to the limitations of RGBD cameras. Their method heuristically chose between an RGBD SLAM approach and an 8-point RANSAC based monocular SLAM depending on the availability of depth information in the scene, and merged the two maps generated by the two individual SLAM approaches. Zhang *et al.* [17] addressed the issue of using a heuristic switch and proposed a single method to handle sparse depth information by combining both features with and without depth. In their method, depth was associated to the features in two ways, from a depth map provided by the RGBD camera and by triangulation using the previously estimated motion for features lacking depth information. One of the shortcomings of their method is that it is a visual odometry method, which lacks a loop closure system and would not achieve global consistency in large scale environments. Ataer-Cansizoglu *et al.* [18] used both features with and without depth in a SLAM framework as well as in postprocessing. As opposed to those methods using only an RGBD camera, we use a separate wide-angle monocular camera along with the RGBD camera for obtaining more constraints using RGBD-to-monocular registration.

RGBD-to-monocular registration was exploited in [19] for calibrating RGB cameras that might have non-overlapping FOVs using a map obtained with an RGBD SLAM system, but the map was assumed to be fixed for the RGBD-to-monocular registration. In contrast, we use RGBD-to-monocular registration to extend the mapped regions and to improve the registration accuracy.

II. MonoRGBD-SLAM

As mentioned in Section I, our MonoRGBD-SLAM system uses both RGBD-to-RGBD and RGBD-to-monocular registration to estimate the poses of the RGBD and monocular frames. The pose estimation is performed based on a graph, where nodes represent the RGBD and monocular frames and edges represent the pairwise registration between the nodes. One way to solve this problem would be performing the registration between all pairs of RGBD-RGBD and RGBD-monocular frames, adding all the registration results as edges to the graph, and running BA by assuming some initial solutions. However, this is computationally expensive and does not provide globally consistent poses because there might be several edges corresponding to incorrect pairwise registration results.

We propose an approach that is (1) computationally feasible by assuming the sequential capture of RGBD frames and by using appearance similarities and (2) robust to incorrect pairwise registration results by checking the pose consistency in the graph. To achieve the goal, we first add edges by using sequential RGBD-to-RGBD matches and then consider other edges obtained from RGBD-to-monocular matches and the other RGBD-to-RGBD matches proposed by a loop closing algorithm. An overview of the proposed system is shown in Figure 2. We detail each step of our SLAM procedure in the following subsections.

Note that we assume the sequential capture for the RGBD frames only; the monocular frames do not need to be ordered, and can even be a sparse set of images captured in regions where the RGBD-to-RGBD registration has difficulties.

A. Initial Graph Construction Using Sequential RGBD-to-RGBD Matches

In the first step, we use a sequence of RGBD frames to construct a graph consisting of nodes corresponding to the RGBD frames, and edges that connect them and contain information that is obtained from the matching process described here. We match each RGBD frame with its 5 neighbors. To match the frames, we first extract SIFT [20] keypoints and descriptors from each image. We then match two frames using a mutual consistency check, by finding the nearest neighbors in the descriptor vector space from the source frame to the target frame and vice versa. The result of this matching usually contains outliers (false matches). In order to find the inlier set, we perform geometric verification using the RANSAC based on 3D-to-3D registration between the two frames. One downside of using RANSAC is the assumption of a fixed error threshold (fixed error band) in which inliers are selected. Selecting a small threshold results in an accurate pose, although some inliers may not be detected and thus their constraints would not be included later in the BA refinement. On the other hand, selecting a large error threshold may result in the inclusion of undesired outliers. As such, we initially select a relatively

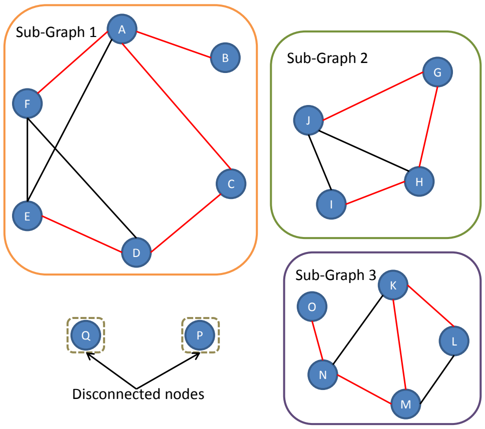


Fig. 3. An example of an initial graph constructed using RGBD-to-RGBD constraints. This graph is disconnected and consists of 3 segments. The MST is computed on each segment and shown in red. The MST is constructed by first sorting all the edges in decreasing order of their weight, followed by picking the edge with the highest weight (as long as it does not form a cycle) and repeating this process the number of edges = number of vertices - 1. This insures that all vertices in the graph are connected using the largest possible numbers of inliers.

small threshold (10 mm) to find the initial inlier set and an accurate rigid body transformation between the two frames. This is followed by performing an MSSE [21] segmentation step, which is an extension of the robust least K-th order statistical estimator, to find the final inlier set. An edge is added between two frames if the number of inliers exceeds a predefined threshold. All the necessary information obtained by this matching procedure is stored to the edge, such as the relative pose between two frames and the inlier set. In addition, we assign a weight to each edge based on the number of inliers multiplied by -1.

B. *Minimum Spanning Tree*

The initial graph contains the relative pose information between frames. To relate the frames with respect to a single reference coordinate system, we construct a sub-graph as the MST using Kruskal's algorithm. This idea is illustrated in a simple example shown in Figure 3. MST provides a simple, yet effective way of connecting all the nodes using the lowest possible weights (i.e., the largest possible numbers of inliers), while assuring that no loops are induced in the graph. Thus no transformation averaging is required when traversing the tree in order to calculate the global poses of the frames. Note that some nodes in the original graph may have been disconnected, since we use a minimum inlier threshold for accepting an edge between two frames. This may result in a graph with multiple disconnected components. Thus we compute an MST for each connected component of the graph.

C. *Addition of RGBD-to-Monocular Constraints*

In the next step, we add more edges to the graph by matching RGBD frames to wide-angle monocular frames. Our aim to add the edges using the monocular frames is twofold:

- In the case of having several MSTs, connect them via RGBD-to-monocular edges.
- Add more constraints to the graph in order to improve the camera pose and map estimates using BA.

In this paper, we used a GoPro Hero 3 camera to capture the monocular frames. Note that the GoPro camera uses a wide-angle fisheye lens and the images are distorted. In addition, the GoPro camera has a resolution of 1920×1080 pixels, while the RGBD camera has a resolution of 640×480 pixels. To achieve accurate feature matching between the RGBD and monocular frames, we compensate for those differences by generating multiple virtual images from each monocular frame as described below.

Virtual Image Generation: The resolution and FOV differences between the RGBD and monocular cameras can reduce the matching accuracy as mentioned above. If we were to generate a single undistorted monocular image (as shown in Figure 4 (b)) from the wide-angle image so that it can cover the entire FOV, then the peripheral regions have perspective distortions[1] and the features in those regions do not match well with those in the RGBD images due to having different descriptors. We propose to generate multiple virtual images from a monocular frame, each of which has the same camera intrinsic parameters as the RGBD camera. This idea is illustrated in Figures 4 and 5. We define multiple virtual cameras, each of which is placed at the camera center of the original monocular frame but has different viewing directions to cover the entire FOV of the original monocular frame (shown in Figure 4 (a)). In this paper we used 9 such virtual cameras (shown in Figure 4 (c)). Using the same intrinsic parameters generates similar appearances between RGBD and monocular images and improves their matching accuracy, as we will show in Section III-C.

Finding RGBD-to-Monocular Match Candidates: In the next step, we extract SIFT keypoints and calculate SIFT descriptors from each virtual image. We then describe the appearance of each virtual image using VLAD [22]. For every virtual image, we find n most similar RGBD frames by finding the nearest neighbors in the VLAD descriptor vector space. Those matches are candidates for potential constraints that may be added to the graph. For each candidate, we perform geometric verification using 3D-to-2D (P3P) RANSAC registration followed by MSSE segmentation. Note that although the keypoint locations and descriptors are computed on the virtual images, the corresponding 2D rays used in the 3D-to-2D registration are defined in the original coordinate system of the monocular frame. Note that it is possible that a matched keypoint is viewed by multiple virtual images. In this case, we only use a single keypoint to avoid duplicates. If the RANSAC is successful, then a new node representing the monocular frame and an edge between the monocular frame and RGBD frame are added to the graph.

[1] Perspective distortions occur when the perspective projection is used for generating a wide FOV image. Although the straight lines remain straight (i.e., lens distortions are corrected), the peripheral regions are stretched and occupy more pixels than the central regions.

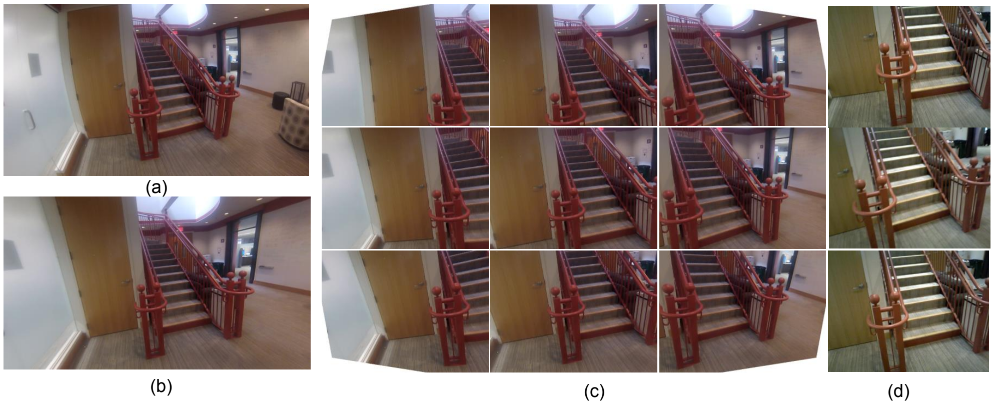


Fig. 4. (a) The original wide-angle monocular image. Straight lines in the scene are distorted due to the fisheye lens distortions. (b) A single undistorted monocular image generated to cover the entire FOV of the original image. Although the straight lines remain straight, the peripheral regions are stretched compared to the central regions due to the perspective distortions, resulting in different feature descriptors in different regions. (c) Nine virtual monocular images generated to cover the entire FOV of the original image. (d) Example RGBD images of the same scene, whose appearance is similar to the virtual images (c) compared to the single undistorted image (b).

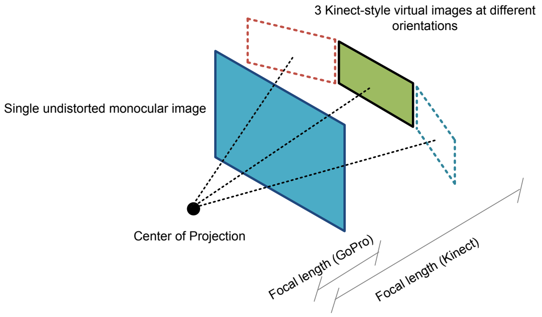

3 Kinect-style virtual images at different orientations
Single undistorted monocular image
Center of Projection
Focal length (GoPro)
Focal length (Kinect)

Fig. 5. An illustration of the virtual image generation. In this example, 3 Kinect-style virtual images at different viewing directions are generated.

D. Addition of Loop Closure RGBD-to-RGBD Constraints

In addition to the constraints added by the sequential RGBD-to-RGBD matching and VLAD-based RGBD-to-monocular matching described in the previous sections, we also use VLAD to find the *n* most similar RGBD-to-RGBD frames that are not included in the sequential matching step. This accounts for large loop closures. To accept an edge, we apply the same RANSAC inlier threshold verification step described previously.

E. Updating the MSTs

Once the additional edges are added, we can update the MSTs by performing another MST that takes into account the additional edges. For instance, let us assume that there are two disconnected segments *Sub_Graph_1* and *Sub_Graph_2* in the original MSTs as shown in Figure 3. If an edge that connects a monocular frame node G_i and *Sub_Graph_1* exists, and another edge connecting the same monocular frame node and *Sub_Graph_2* is also available, then the two segments can be connected in the updated MSTs. Using the updated MSTs, we compute the global poses of all the nodes in each MST by traversing the tree and concatenating the relative pose assigned to each edge.

F. Edge Consistency Check

The graph includes many edges other than the edges included in the MSTs due to all of the sequential RGBD-to-RGBD, VLAD-based RGBD-to-monocular, and VLAD-based RGBD-to-RGBD matches. Those edges provide additional constraints in the pose estimation, but some of them might be incorrect due to incorrect pairwise registration. We prune the incorrect edges by comparing the relative pose assigned to the edge and that computed based on the MSTs. Specifically, we compute

$$\mathbf{T}_{\text{diff}} = (\mathbf{T}_{\text{A}}^{-1}\mathbf{T}_{\text{B}})\mathbf{T}_{\text{relative}}^{-1}, \qquad (1)$$

where $\mathbf{T}_{\text{A}}$ and $\mathbf{T}_{\text{B}}$ are the global poses of frames A and B obtained from the MSTs, and the term $\mathbf{T}_{\text{A}}^{-1}\mathbf{T}_{\text{B}}$ is the predicted relative pose between frames A and B. $\mathbf{T}_{\text{relative}}$ is the measured relative pose between the two frames and was estimated using RANSAC. We threshold the translation component of $\mathbf{T}_{\text{diff}}$ to prune the inconsistent edges.

G. Bundle Adjustment

Bundle adjustment (BA) jointly optimizes the camera pose and the 3D structure parameters that are viewed and matched over multiple frames by minimizing a cost function. BA can be performed by using measurements obtained as 2D pixels (minimizing reprojection errors) or 3D points (minimizing 3D point-to-point distance errors). We found that the results obtained by the 2D-based BA were underwhelming, constantly converging at incorrect local minima. We therefore employed a 3D-based BA. However, the monocular frames do not provide 3D measurements. In order to associate each inlier point in the monocular frame with 3D information, we propose the following method. Each monocular frame is matched with an RGBD frame, and their relative pose is available. Thus, in order to associate 3D information to all inlier points in monocular frames, we simply transfer the corresponding 3D points from the RGBD frame to the monocular frame. In cases where an inlier point from the monocular frame has multiple matches from different

RGBD frames, we transfer all associated 3D points to the monocular frame and then average their coordinates. The inlier point from the monocular frame is then assigned to the averaged 3D point. In our experiments, we found that the extra dimension provides valuable information that helps the convergence of BA. The cost function to be minimized can be formulated as

$$\arg\min_{\mathbf{X}^i,\mathbf{C}_k} \sum_k \sum_i v_k^i ||\hat{\mathbf{X}}_k^i - \mathbf{C}_k^{-1}\mathbf{X}^i||^2, \tag{2}$$

where v_k^i is either 1 if the i-th 3D landmark point $\mathbf{X}^i$ is observed by the k-th frame or 0 otherwise. $\hat{\mathbf{X}}_k^i$ is its corresponding 3D measurement point observed by the k-th frame and $\mathbf{C}_k$ is the global pose of the k-th frame. We used the Ceres Solver [23] for the optimization.

H. Obtaining 3D Model

The 3D landmarks that BA optimizes are a sparse set of keypoints which results in a sparse 3D reconstruction. In order to obtain a dense model of the environment, we simply transfer a sub-sampled set of the 3D points provided by each RGBD frame into a global frame using the optimized camera poses.

III. Experiments

We evaluated the performance of the proposed SLAM method by mapping an entire floor of a typical office building. We used an Asus Xtion (640 × 480 pixel resolution) for capturing RGBD data and a wide-angle GoPro Hero 3 (1920 × 1080 pixel resolution) for capturing monocular images. The two cameras were placed side by side on a tablet PC, although we assume neither synchronization nor fixed relative pose between the two cameras in our method. The sequence involved moving the hand-held cameras around the large office and eventually returning back to the starting location. The sequence consisted of 3222 RGBD frames and 2656 monocular images.

A. Disconnected Graph vs. Connected Graph vs. Optimized Graph

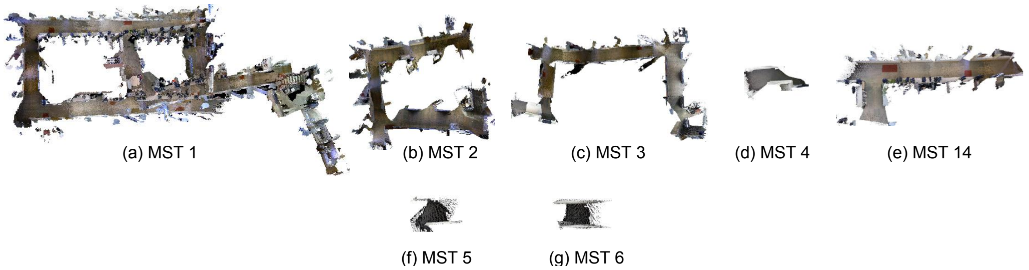
As we described in Section II, we first use RGBD-to-RGBD sequential registration to construct the initial graph. This graph may be disconnected due to rejected constraints. We then find an MST for each disconnected segment. Figure 6 shows an example of the disconnected MSTs which were initially obtained by performing the proposed method on the sequence. In this experiment, the map was disconnected into 14 segments, although the frames in 5 of those segments (segments 1, 2, 3, 4 and 14) accounted for ≈ 95% of all frames and are the ones shown in Figures 6 (a) to (e). In the next step, we calculated a new set of MSTs by adding the RGBD-to-monocular constraints that were found by VLAD and RANSAC geometric verification. The largest MST is shown in Figure 7 (a), demonstrating that the five main segments are now connected as a single MST. Note that although the MST contains a little drift, it provides a good initial solution which will be later refined using BA. Figure 7 (b) shows the 3D reconstructed map after taking into account all RGBD-to-RGBD constraints and applying 3D-3D BA optimization. It can be seen that the drift has been significantly reduced and the result is a globally consistent map. We also note that the remaining 9 segments that were not included, contained very limited texture. As such, neither RGBD-to-RGBD nor RGBD-to-monocular constraints were able to connect them. Two examples of those segments are shown in Figures 6 (f) and (g).

B. Proposed Method vs. RGBD Only Method

We compared the proposed method with a method that only uses RGBD frames for registration and mapping. The process is identical to our method except that only RGBD-to-RGBD constraints are used to connect the MSTs. The qualitative results can be seen in Figure 7 (d). The statistical results of this experiment are summarized in Table I. The RGBD only approach connected two segments (1 and 14). These two segments accounted for 1689 RGBD frames out of a total 3222 frames. In comparison, the proposed method connected 5 segments (as mentioned above) containing a total of 3060 RGBD frames. In addition, 26762 RGBD-to-monocular constraints were added to the graph, which resulted in a better optimized graph as can be seen in Figures 7 (b) and 7 (d). The RGBD-to-monocular matching also recorded a higher average number of inliers (almost double) when compared to RGBD-to-RGBD matching.

C. Multiple Virtual Monocular Images vs. Single Undistorted Monocular Image

In this experiment, we compared the mapping results of the proposed method with an identical method except that it uses a single undistorted monocular image instead of the proposed multiple virtual images. The results of this experiment are summarized in Table I. The method using a single undistorted monocular image added only 6213 constraints to the graph, compared with 26762 constraints that were added by the proposed method. In addition, this method was only able to connect 2 segments (1 and 14), compared to the 5 segments connected via the proposed method. We also note that at a first glance, it might seem that the matching performance of both methods is even, since the average number of RGBD-to-monocular inliers is roughly the same. However, the average number alone does not reveal the full truth and after analysis, we found that the matching performance was not even. The reason is that the proposed method adds many more constraints that are not added by the method using a single undistorted monocular image, and many of those constraints contain a relatively low number of inliers. As such, the average number of inliers is reduced. The same constraints that resulted in reducing the average are also responsible for connecting more segments (5) in comparison to a single monocular image (2). To fairly compare the matching performance, we calculated the average number of inliers on the edges connecting the same frames. There were a total of 4726 of those frames, out of a possible 6213. The average number of inliers using the proposed method


(a) MST 1

(b) MST 2

(c) MST 3

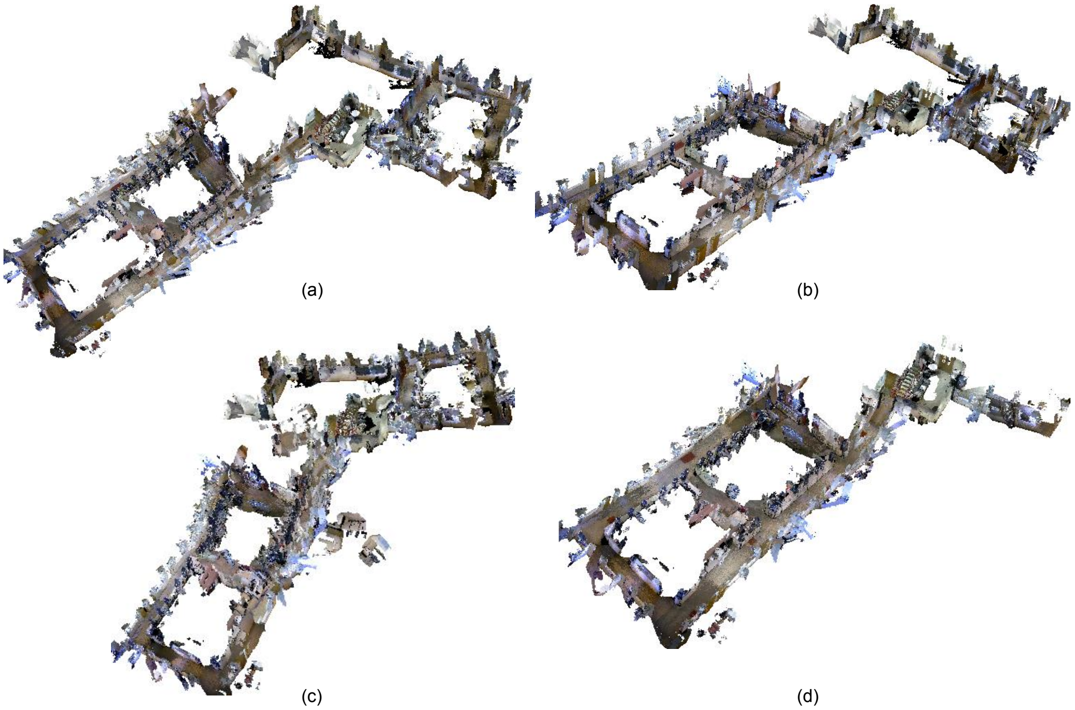
(d) MST 4

(e) MST 14

(f) MST 5

(g) MST 6

Fig. 6. Examples of disconnected MSTs obtained using sequential RGBD-to-RGBD registration. The segments (a) to (e) are later connected using RGBD-to-monocular matches, while (f) and (g) are not due to the limited textures.


(a)

(b)

(c)

(d)

Fig. 7. (a) The updated MST after connecting the segments using the virtual monocular images. (b, c) The reconstructed maps after adding all RGBD-to-monocular and RGBD-to-RGBD constraints and optimizing using (b) 3D-3D BA and (c) 3D-2D BA. (d) The reconstructed map using the RGBD only approach.

TABLE I

STATISTICAL COMPARISON BETWEEN THE RESULTS OBTAINED WITH THE PROPOSED METHOD, A METHOD THAT ONLY USES RGBD FRAMES, AND A METHOD THAT USES SINGLE UNDISTORTED MONOCULAR IMAGES WITHOUT GENERATING MULTIPLE VIRTUAL IMAGES.

Method	No. of RGBD-RGBD edges	No. of RGBD-mono. edges	No. of segments in largest MST	No. of frames in largest MST	Avg. no. of RGBD-RGBD inliers	Avg. no. of RGBD-mono. inliers
RGBD-mono. (virtual images)	27950	26762	5	3060	33.0	60.7(93)
RGBD only	27143	N/A	2	1689	32.9	N/A
RGBD-mono. (single image)	27950	6213	2	1689	33.0	60.2(64)

was 93.2 whereas the method using the single undistorted monocular image recorded an average of 64. In addition, we selected a number of similar RGBD and monocular images, and compared the matching between RGBD and monocular images using virtual images vs. using a single monocular image. The matching results can be seen in Figure 8, demonstrating that there are significantly more inliers using the proposed method than using a single monocular image. The number of inliers for the two examples using the proposed method were 64 and 112 respectively, whereas using a single monocular image resulted in 16 and 33 inliers only.

D. 3D-3D Bundle Adjustment vs. 3D-2D Bundle Adjustment

We proposed assigning 3D information to the monocular image keypoints using the corresponding RGBD images in order to perform 3D-3D BA. We found that BA using a 3D-3D cost function performs significantly better than BA using a 3D-2D cost function. This is illustrated in Figures 7 (b) and

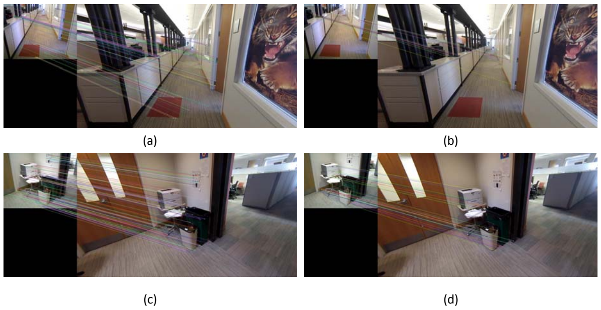


Fig. 8. (a) and (c) show the inlier correspondences between a monocular image and an RGBD image using the virtual images. (b) and (d) show the inliers of the same images using a single undistorted monocular image.

(c). The figure shows that the global consistency of the map is significantly worse when using 3D-2D BA in comparison to 3D-3D BA.

IV. Conclusions

In this paper, we presented a SLAM system that employs both an RGBD camera and a wide-angle monocular camera for combining the advantages of the two types of cameras. The proposed system is able to handle large-scale indoor environments by using both RGBD-to-RGBD and RGBD-to-monocular registration. We generate multiple virtual images for each wide-angle monocular image in order to compensate for the difference of FOV and resolution of the cameras. We construct a graph consisting of nodes which represent the camera frames, and edges which denote the pairwise registration results between the frames. We then compute MSTs and traverse them to calculate the initial global poses of the cameras, which are used to prune edges from the original graph that have inconsistent poses. We finally run bundle adjustment on the graph using consistent edges. We showed in our experiments that the proposed SLAM method performs well in reconstructing large-scale indoor environments. The experiments showed that the RGBD only approach struggled to reconstruct the whole sequence, mainly due to its limited FOV. We also showed that our method using multiple virtual images performs better than using a single undistorted monocular image, both in terms of the mapping results and the average number of RGBD-to-monocular inliers. Currently, the main limitation of our system is the time it takes to fully complete the process. For instance, the computational time required to map the sequence described in Section III is approximately 4 hours on a standard desktop PC. Improving the efficiency of this method is possible by using keyframes instead of every frame. Additionally, it is possible to use faster feature extraction techniques, which could vastly improve the speed of the matching algorithm.

Acknowledgments

We thank Jay Thornton, Adina Stoica, Gim Hee Lee, Esra Ataer-Cansizoglu, and Chen Feng for valuable discussion. This work was supported by MERL.

References

[1] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in *Proc. European Conf. Computer Vision (ECCV)*, Sep. 2014.

[2] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[4] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1994, pp. 593–600.

[5] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. IEEE Int’l Symp. Mixed and Augmented Reality (ISMAR)*, Nov. 2007, pp. 1–10.

[6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Proc. IEEE Int’l Conf. Computer Vision (ICCV)*, Nov. 2011, pp. 2320–2327.

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments,” in *Proc. Int’l Symp. Experimental Robotics (ISER)*, Dec. 2010.

[8] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *Proc. European Conference on Computer Vision (ECCV)*, pp. 430–443, 2006.

[9] H. Du, P. Henry, X. Ren, M. Cheng, D. Goldman, S. Seitz, and D. Fox, “Interactive 3d modeling of indoor environments with a consumer depth camera,” in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 75–84.

[10] C. Audras, A. Comport, M. Meilland, and P. Rives, “Real-time dense appearance-based slam for rgb-d sensors,” in *Proc. Australasian Conference on Robotics and Automation (ACRA)*, 2011.

[11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” in *Proc. IEEE Int’l Symp. Mixed and Augmented Reality (ISMAR)*, Oct. 2011, pp. 127–136.

[12] H. Roth and M. Vona, “Moving volume KinectFusion,” in *Proc. British Machine Vision Conf. (BMVC)*, Sep. 2012.

[13] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, “Robust real-time visual odometry for dense RGB-D mapping,” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5724–5731.

[14] J. Chen, D. Bautembach, and S. Izadi, “Scalable real-time volumetric surface reconstruction,” *ACM Trans. Graphics*, vol. 32, no. 4, pp. 113:1–113:16, Jul. 2013.

[15] C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for RGB-D cameras,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 2100–2106.

[16] G. Hu, S. Huang, L. Zhao, A. Alempijevic, and G. Dissanayake, “A robust RGB-D SLAM algorithm,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012.

[17] J. Zhang, M. Kaess, and S. Singh, “Real-time depth enhanced monocular odometry,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Sep. 2014.

[18] E. Ataer-Cansizoglu, Y. Taguchi, and S. Ramalingam, “Pinpoint SLAM: A hybrid of 2D and 3D simultaneous localization and mapping for RGB-D sensors,” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, May 2016.

[19] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and Y. Miki, “Calibration of non-overlapping cameras using an external SLAM system,” in *Proc. Int’l Conf. 3D Vision (3DV)*, Dec. 2014.

[20] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int’l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] A. Bab-Hadiashar and D. Suter, “Robust segmentation of visual data using ranked unbiased scale estimate,” *Robotica*, vol. 17, no. 6, pp. 649–660, 1999.

[22] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, “Aggregating local image descriptors into compact codes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[23] S. Agarwal, K. Mierle, and Others, “Ceres solver,” http://ceres-solver.org.