MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Calibration of Non-Overlapping Cameras Using an External SLAM System

Ataer-Cansizoglu, E.; Taguchi, Y.; Ramalingam, S.; Miki, Y.

TR2014-106 December 2014

Abstract

We present a simple method for calibrating a set of cameras that may not have overlapping field of views. We reduce the problem of calibrating the non-overlapping cameras to the problem of localizing the cameras with respect to a global 3D model reconstructed with a simultaneous localization and mapping (SLAM) system. Specifically, we first reconstruct such a global 3D model by using a SLAM system using an RGB-D sensor. We then perform localization and intrinsic parameter estimation for each camera by using 2D-3D correspondences between the camera and the 3D model. Our method locates the cameras within the 3D model, which is useful for visually inspecting camera poses and provides a model-guided browsing interface of the images. We demonstrate the advantages of our method using several indoor scenes.

International Conference on 3D Vision (3DV)

Calibration of Non-Overlapping Cameras Using an External SLAM System

Esra Ataer-Cansizoglu*, Yuichi Taguchi†, Srikumar Ramalingam†, and Yohei Miki‡
**Northeastern University, Boston, MA, USA*
†*Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA*
‡*Mitsubishi Electric Corporation, Kamakura, Kanagawa, Japan*

Abstract—**We present a simple method for calibrating a set of cameras that may not have overlapping field of views. We reduce the problem of calibrating the non-overlapping cameras to the problem of localizing the cameras with respect to a global 3D model reconstructed with a simultaneous localization and mapping (SLAM) system. Specifically, we first reconstruct such a global 3D model by using a SLAM system using an RGB-D sensor. We then perform localization and intrinsic parameter estimation for each camera by using 2D-3D correspondences between the camera and the 3D model. Our method locates the cameras within the 3D model, which is useful for visually inspecting camera poses and provides a model-guided browsing interface of the images. We demonstrate the advantages of our method using several indoor scenes.**

***Keywords*-non-overlapping camera calibration; camera network; simultaneous localization and mapping (SLAM)**

I. Introduction

Camera calibration has been a long-standing research topic as many vision algorithms require accurate intrinsic and extrinsic parameters of cameras. Nowadays several calibration toolboxes are readily available [1], [2], [3], [4] for computing intrinsic parameters of perspective and omnidirectional cameras. Extrinsic parameters among multiple cameras can be easily computed as well, if the cameras share the field of views (FOVs). However, several applications, such as surveillance and car navigation, benefit more from cameras that do not have overlapping FOVs.

In this paper, we address the problem of calibrating cameras with non-overlapping FOVs. We present a simple and practical method by leveraging the recent advancement of SLAM systems using a Kinect-style sensor [5], [6], [7], [8], [9], [10], [11]. An overview of our method is shown in Figure 1. We first reconstruct a 3D model of the scene in which the non-overlapping cameras are located using an RGB-D SLAM system. Once the 3D model is reconstructed, the calibration can be done by localizing each camera with respect to the 3D model using 2D-3D correspondences between the camera and the 3D model. Note that the map reconstruction process can be done using any SLAM system and is completely independent of the calibration process of non-overlapping cameras; thus our use of the SLAM system is *external*, as opposed to the *internal* use of SLAM algorithms employed in previous work [12], [13], [14] for calibrating non-overlapping cameras attached on a mobile platform as described in Section I-B

A. Contributions

The main contributions of this paper are summarized as follows.

- We present a method for calibrating intrinsic and extrinsic parameters of non-overlapping cameras by exploiting an external SLAM system.
- We describe an efficient algorithm for localizing a 2D image with respect to the reconstructed 3D model.
- We demonstrate a model-guided browsing interface of the non-overlapping cameras as an application of our method.

B. Related Work

Here we review prior camera calibration methods that assume non-overlapping FOVs of the cameras. We categorize the methods into (1) those calibrating a multi-camera rig attached on a mobile platform and (2) those calibrating a set of stationary cameras.

Methods in the first category exploit the motion of a mobile platform for calibrating multiple cameras rigidly attached on the platform. Those methods capture image sequences synchronously using the multiple cameras while moving the platform, and then perform SLAM individually for each camera to compute its relative motions. Esquivel et al. [15] matched the relative motions of the multiple cameras to compute the extrinsic parameters between cameras, which is the same formulation as the hand-eye calibration problem [16], [17]. However, only matching the relative motions has degeneracies when specific motions (e.g., planar motions, rotations and screw motions about an axis) or special camera configurations (e.g., camera configurations where the centers lie on a straight line) are used [12]. Several methods have addressed the degeneracy by additionally matching scene points, fusing the maps reconstructed from individual cameras, and running bundle adjustment to jointly optimize the relative motions of a reference camera, the extrinsic parameters of the other cameras, and the scene points [12], [13], [14]. Note that the above methods use SLAM algorithms *internally*, i.e., the cameras to be calibrated are used for SLAM; thus they require the motion of cameras, which is not applicable if the cameras are stationary. In contrast, we leverage an *external* SLAM system independent of the cameras to be calibrated; thus our method is applicable

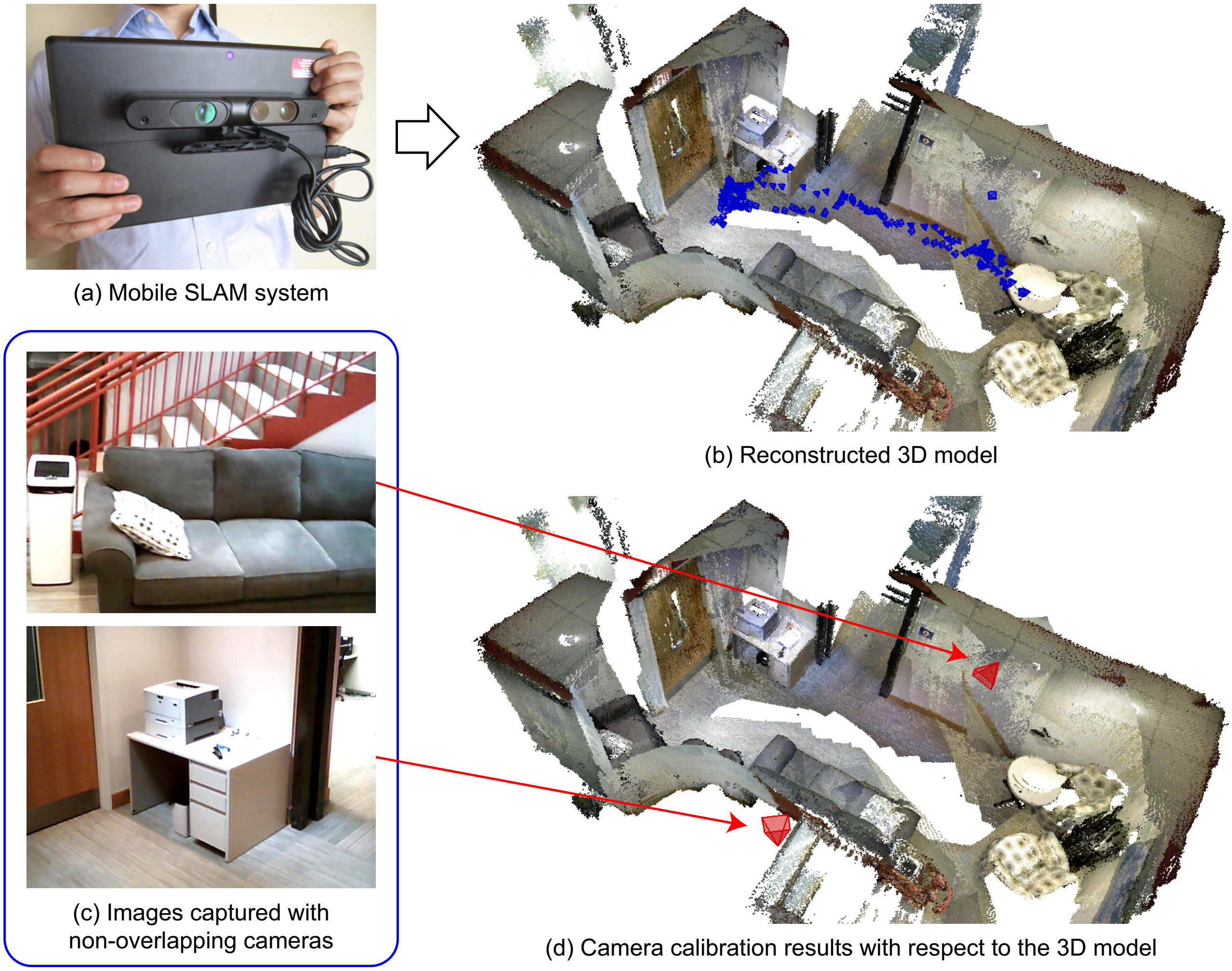
(a) Mobile SLAM system
(b) Reconstructed 3D model
(c) Images captured with non-overlapping cameras
(d) Camera calibration results with respect to the 3D model

Figure 1. Overview of our method for the *Lounge* scene. We use (a) an RGB-D SLAM system running in real time on a tablet to reconstruct (b) a 3D model of the scene where the non-overlapping cameras are located. The blue camera icons shown in (b) denote the poses of keyframes computed in the SLAM system. We then use the 3D model to localize (c) images captured with the non-overlapping cameras using 2D-3D correspondences. (d) Poses of the non-overlapping cameras can be obtained with respect to the 3D model, shown as the red camera icons.

to stationary cameras. Knorr et al. [18] also assumed the motion of cameras and presented an approach for refining the extrinsic parameters during online operations by using the homographies computed with respect to the ground plane in an extended Kalman filter framework.

Methods in the second category relate multiple stationary cameras with respect to a single reference object. After the pose of each camera is computed with respect to the reference object, the poses of multiple non-overlapping cameras can be related through the reference object. One can use a large reference object (e.g., a calibration room with several known 3D locations) so that all the cameras can observe a part of the reference object, but building such a setup is often not practical. Several methods have used mirrors to image a standard-size reference object (e.g., checkerboard) that is not originally in the FOV of the camera. A planar mirror [19], [20], [21], multiple planar mirrors [22], and a spherical mirror [23] have been used. These techniques are simple and easy to use for small configurations that use fewer cameras. However in larger setups, the mirror-based techniques pose several challenges that are not always straightforward to resolve. First, the accuracy degrades as the distance between cameras becomes larger, since the image of the reference object becomes smaller. Second, there is always the under-emphasized, sometimes theoretically impossible, mirror-grid placement problem that requires the user to place the mirrors and grids such that the multiple cameras can observe either direct or reflected views of the calibration pattern simultaneously.

Another set of methods in the second category exploits the motions of objects in the scene (e.g., humans and cars), which is in particular used for surveillance camera networks [24]. Several methods have been proposed for determining the transition probabilities of one object observed in one camera to another camera [25], [26], which provide the topology of the camera network but not the geometric calibration. Rahimi et al. [27] modeled the object motions using the linear Gaussian Markov dynamics and estimated one rotation and two translation parameters between multiple cameras, assuming the ground plane is known in each camera. Using linear object motion models, Pflugfelder and Bischof [28] computed extrinsic parameters given the camera rotations and intrinsic parameters, while Micusik [29] computed them given only the gravity vector directions. Although those approaches showed promising results, they assume some calibration parameters to be known, and their accuracy is limited due to the assumptions on the object motions. In contrast, our method provides accurate intrinsic and extrinsic parameters for camera networks by localizing the cameras with respect to a global 3D model reconstructed with an external SLAM system. In addition, the reconstructed 3D model allows us to compute the transition probabilities in the camera network by simulating object motions in the 3D model if necessary.

To the best of our knowledge, a recent work of Heng et al. [30] is the closest to ours. Although their method is designed for calibrating multiple cameras attached on a moving vehicle and close to the first category, it separates the map reconstruction process from the camera localization process. For the map reconstruction, they used all the images captured with all the cameras in a visual SLAM system. Once the map is reconstructed, the cameras can be localized with respect to the map by using 2D-3D correspondences. Their focus was on computing camera extrinsic parameters as well as the rig pose jointly for moving platforms, whereas our focus is on estimating the intrinsic and extrinsic parameters of stationary cameras. Moreover, they used 2D cameras for both map reconstruction and localization, while we use different modalities, an RGB-D sensor and 2D cameras, for map reconstruction and localization respectively.

II. Non-Overlapping Camera Calibration

Figure 1 shows an overview of our method. We use a mobile SLAM platform consisting of an RGB-D sensor and a tablet to reconstruct a 3D model of the scene. We then perform localization of the non-overlapping cameras with respect to the 3D model using 2D-3D correspondences. We detail the map reconstruction and localization processes in the following subsections.

A. Map Reconstruction

Recently several SLAM systems using a Kinect-style sensor have demonstrated impressive 3D reconstruction results [5], [6], [7], [8], [9], [10], [11]. We leverage those SLAM systems and show a novel application of them to non-overlapping camera calibration.

We used an RGB-D SLAM system that uses both point and plane features as primitives [8]. Since planes are the dominant structure in man-made scenes, using plane features improves the registration accuracy as well as accelerates the processing speed due to the smaller number of feature matching candidates. The system is a keyframe-based SLAM system, where frames with representative poses are stored as keyframes in a map. For each new RGB-D frame, the system extracts point features using the SURF keypoint detector and plane features using a RANSAC-based plane fitting algorithm on the depth map. The frame is then registered with respect to the map by using a RANSAC-based registration algorithm that uses both the point and plane features. The frame is added to the map if its estimated pose is sufficiently different from any existing keyframe poses. The keyframe poses as well as point and plane features in the map are jointly optimized using bundle adjustment asynchronously from the frame-based registration.

In addition to the techniques presented in [8], we implemented a loop closing algorithm to improve the accuracy of SLAM when the camera comes back to locations visited previously. For this purpose, we describe the appearance of each frame by using a vector of locally aggregated descriptors (VLAD) [31] representation on the SURF descriptors of the point features. We compute VLAD for all the existing keyframes in the map, and check the appearance similarity with a new keyframe when we add it to the map. In addition to the appearance similarity, we check the pose similarity between the new keyframe and the existing keyframes. If both similarities are high for any existing keyframe, then we perform the geometric verification using the RANSAC-based registration between the frames; if there are enough number of inliers, we add the constraints between corresponding point/plane features appearing in the two keyframes in the bundle adjustment.

The SLAM system was implemented on a Surface Pro tablet with an Asus Xtion PRO LIVE sensor as shown in Figure 1(a). The system runs about 3 frames per second on the tablet, enabling interactive 3D reconstruction; the operator can get the feedback on whether the frames are successfully registered or not and determine where to scan next in real time. Figure 1(b) shows a 3D model as well as keyframe poses generated by our system.

B. Camera Localization

Given the 3D model of the scene, our goal is to compute the pose of each camera with respect to the 3D model. Since the reconstructed 3D model acts as a single large-size 3D reference object, extrinsic parameters between multiple non-overlapping cameras can be obtained once each of the cameras is localized with respect to the 3D model. Our localization works for each camera in the following two stages: (1) finding 2D-3D point correspondences between the image and the 3D model; and (2) estimating the camera pose by using a Perspective-n-Point (PnP) algorithm.

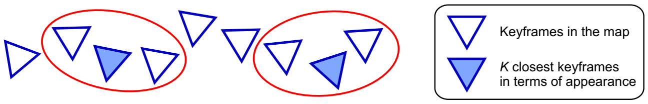

Figure 2. Selecting candidate keyframes for point descriptor matching. We first select K ($= 2$ in this figure) keyframes that are closest to the query image in terms of the appearance using the VLAD descriptor. Then for each of the K candidates, we add $N - 1$ ($= 2$) keyframes that are closest in terms of their poses to form a cluster of $N (= 3)$ keyframes. The descriptor matching is done for each of the clusters of keyframes. After the geometric verification using RANSAC, we select the best cluster that produces the largest number of inliers.

Due to repetitive patterns and textureless regions in many indoor scenes, finding point correspondences between a query image and the entire 3D model is not straightforward. Furthermore, such an all-to-all matching approach would be time-consuming. To handle these problems, we use appearance-based keyframe matching and geometric verification to find the correspondences. Figure 2 shows the keyframe-matching technique. We first find a set of candidate keyframes that are close to the query image in terms of the appearance using the VLAD, similar to the loop closing process described in Section II-A. The VLAD descriptors for the keyframes in the map can be pre-computed. Given a query image, we compute the VLAD descriptor on the query image, and then match it with those of the keyframes in the map. We consider the K closest keyframes as candidates. Then, for each of the K candidates, we form a cluster of N keyframes by adding $N - 1$ closely located keyframes. The closely located ones can be identified by finding the similarity in the 6 degrees-of-freedom (DOF) pose space. In practice, we found that the similarity computed using just 3 DOF translation is sufficient. Then, we perform the point descriptor matching between the query image and the N keyframes in each cluster. The parameter K, denoting the number of clusters, depends on the nature of the scene. For example, if the scene consists of R large repetitive patterns, then using $K \leq R$ may lead to an incorrect pose. The parameter N, denoting the size of each cluster, can be chosen based on the difference between the FOV of the camera used in SLAM and that of the camera used for obtaining the query image. If the query image observes a large portion of the scene, we can use a large value for N for robustness. For the scenes shown in Figures 1, 3, and 4, we set in experiments $(K, N) = (1, 3)$, $(1, 1)$, and $(1, 3)$, respectively.

In the second stage, we geometrically verify the candidate point correspondences using RANSAC. Here we have two different cases. If the camera intrinsic parameters are known, we use the standard P3P algorithm [32]. Otherwise, we use the P5Pfr algorithm [33] to compute the intrinsic parameters (focal length, distortion parameters) along with the 6 DOF pose. In experiments, we computed only one distortion parameter, which makes the P5Pfr algorithm over-determined. We select the best solution out of the K candidate clusters of the keyframes that produces the largest number of inliers. The initial estimates for the intrinsic parameters and the pose are refined using the nonlinear least squares that minimizes the sum of reprojection errors for all the inliers.

III. Experiments

We performed experiments in several indoor scenes shown in Figures 1, 3, and 4, which we refer to as *Lounge*, *Reception*, and *Garage* scenes, respectively. The scenes were reconstructed by using the mobile SLAM system as described in Section II-A. For the *Lounge* and *Reception* scenes, 2D images were captured by using a single USB web camera (640×480 pixel resolution) placed at different locations, and their poses were estimated by using the P5Pfr algorithm followed by the nonlinear least squares. On the other hand, for the *Garage* scene, 2D images were captured by using a GoPro camera (1280×720 pixel resolution) mounted at different locations on a car. We calibrated the GoPro camera offline using a checkerboard [3] and corrected the distortions of the captured images using the calibration result. We then estimated the poses of the images using the P3P algorithm followed by the nonlinear least squares.

A. Qualitative Results

Figures 1, 3, and 4 demonstrate the results of our calibration method. One of the advantages of our method is that it allows us to visually inspect the estimated camera poses with respect to the reconstructed 3D model; it can be seen that the poses obtained from our method visually match with those we used for capturing the images. We also developed a visualization interface for browsing the 2D images with the aid of the reconstructed 3D model. Please refer to the supplementary video demonstrating the interface. The interface was inspired by the Photo Tourism system [34], where the sparse point clouds and camera poses reconstructed using structure from motion were used to browse images from geometrically correct locations.

B. Quantitative Analysis

To perform quantitative analysis, we estimated the intrinsic parameters of the USB web camera using a checkerboard [2] as the ground truth and compared the parameters with those obtained with the P5Pfr algorithm in our method. Table I illustrates the results. Note that the camera model used in the P5Pfr algorithm [33] and in [2] are different (in terms of the focal length and the lens distortion model), which implies that we cannot compare the exact values of these parameters. Nevertheless, the intrinsic parameters obtained by our method are close to those obtained with [2]; in particular, the focal length, which is typically the most important intrinsic parameter for camera localization, has an average error of 4% with respect to the ground truth (computed as the mean for the x and y axes).

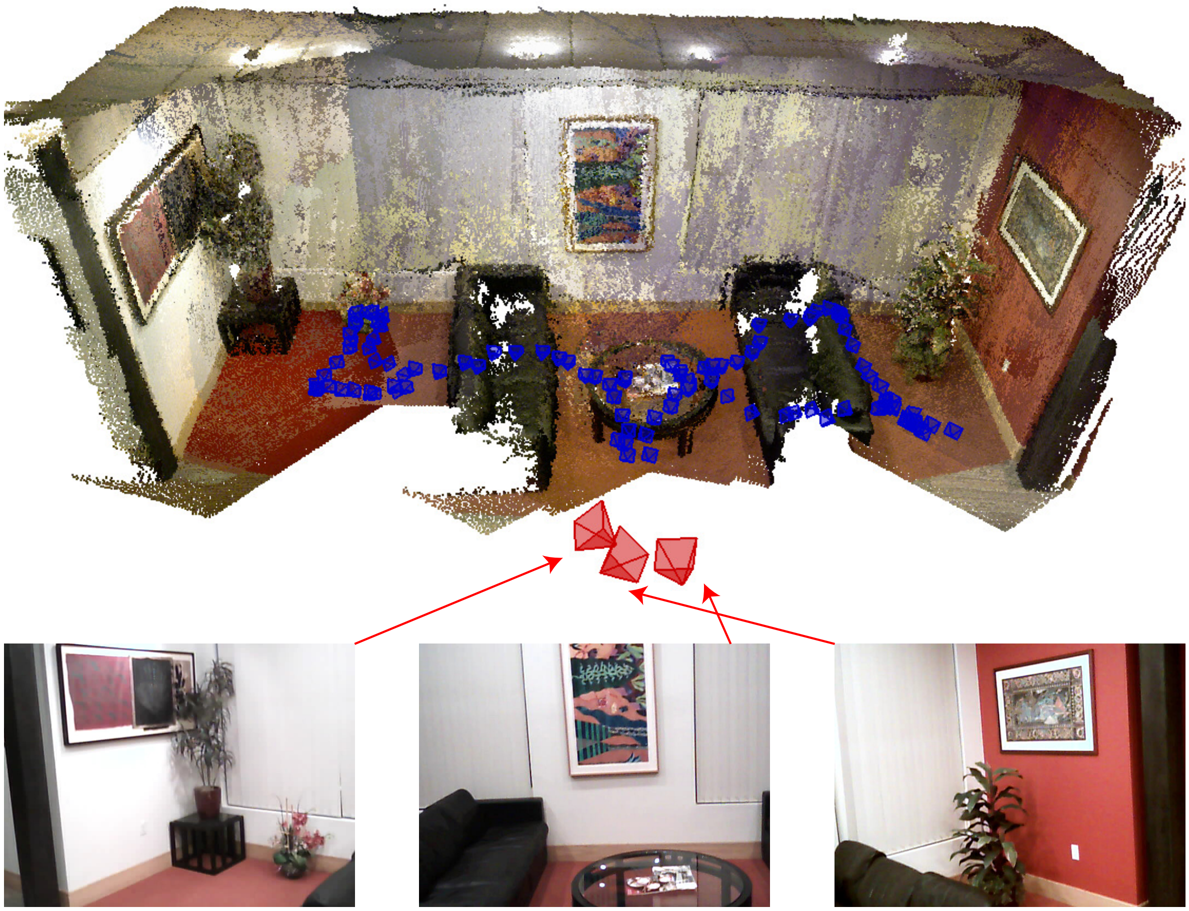

Figure 3. Results for the *Reception* scene. The reconstructed 3D model is depicted with the poses of keyframes (blue camera icons) as well as those of the non-overlapping images (red camera icons). The images were captured with a USB web camera and their poses were computed using the P5Pfr algorithm.

C. Comparison between P5Pfr and P3P

In our setup where non-overlapping cameras are placed in large-scale scenes, obtaining the ground truth poses of the cameras is challenging; therefore we evaluated the results of extrinsic camera calibration by comparing the camera poses estimated using the P5Pfr algorithm (with unknown intrinsic parameters) and the P3P algorithm (with intrinsic parameters given by [2]) for the *Lounge* and *Reception* scenes. Figure 5 visually compares the camera poses, while Table II shows the difference of the poses in translation and rotation. The translation difference was computed as the Euclidean distance between two camera centers, while the rotation difference was computed as $\theta = \| \log(R_1^T R_2)\|_F/\sqrt{2}$, which is the angle of the rotation matrix required to transform one rotation matrix R_1 to the other R_2. Note that inliers selected by P5Pfr and P3P may not be the same due to the differences in the camera models. Nevertheless, the poses computed by the two algorithms are close, which indicates that the computed poses are close to the ground truth. The translation differences are small compared to the size of the scene (approximately $7 \times 4 \times 3$ m for the *Lounge* scene and $7 \times 2 \times 3$ m for the *Reception* scene), except for the image 3 in the *Reception* scene, where the number of inliers was small and the inliers were distributed only around the center of the image. We also observed that the average reprojection errors were less than 2 pixels.

D. Processing Time and Statistics

In our experiments the SLAM pipeline for reconstructing 3D models was completely done on the tablet in real time. Scanning the entire scenes in Figures 1, 3, and 4 took about 5 minutes, and those models contained 175, 110, and 205 keyframes, respectively. The localization process took about 0.2 seconds for each image, demonstrating that the non-overlapping camera calibration can be efficiently done once the 3D models are obtained.

IV. Conclusions and Discussion

RGB-D sensors such as Kinect have made breakthroughs in many vision problems such as 3D reconstruction and

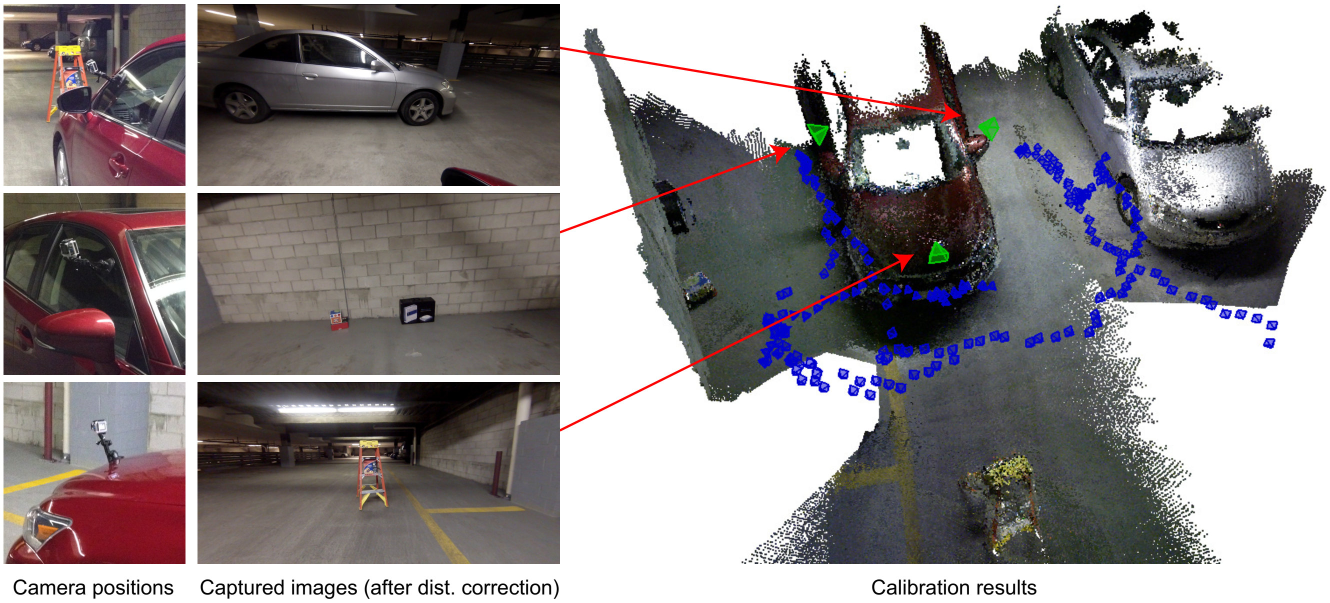

Figure 4. Results for the *Garage* scene. The reconstructed 3D model is depicted with the poses of keyframes (blue camera icons) as well as those of the non-overlapping images (green camera icons). The images were captured with a GoPro camera, whose distortions were corrected by using the calibration result obtained with [3]. The poses of the images were then computed using the P3P algorithm.

Table I

INTRINSIC PARAMETERS COMPUTED USING THE P5PFR ALGORITHM FOLLOWED BY NONLINEAR LEAST SQUARES IN OUR METHOD FOR DIFFERENT SCENES AND IMAGE LOCATIONS. FOR THE *Lounge* SCENE IN FIGURE 1, THE IMAGES 1 AND 2 CORRESPOND TO THE TOP AND BOTTOM IMAGES IN (C). FOR THE *Reception* SCENE IN FIGURE 3, THE IMAGES 1 TO 3 ARE FROM THE LEFT TO RIGHT IN THE BOTTOM ROW. THE GROUND TRUTH VALUES OBTAINED USING A CHECKERBOARD [2] ARE ALSO SHOWN. WE COMPUTED A SINGLE FOCAL LENGTH FOR OUR METHOD, WHILE [2] PROVIDES TWO FOCAL LENGTHS IN THE x AND y AXES. NOTE ALSO THAT P5PFR AND THE CHECKERBOARD METHOD [2] USE DIFFERENT LENS DISTORTION MODELS. THE UNIT IS IN PIXELS AND THE USB WEB CAMERA HAS A RESOLUTION OF 640×480 PIXELS.

	Lounge		*Reception*			Ground Truth
	Image 1	Image 2	Image 1	Image 2	Image 3	
Focal Length	801.1	914.6	869.0	838.8	827.7	(851.2, 865.2)
Principal Point (x)	312.2	282.1	346.2	346.5	342.5	361.9
Principal Point (y)	157.1	219.6	226.4	239.8	345.8	216.4

Table II

DIFFERENCE BETWEEN THE CAMERA POSES COMPUTED USING THE P5PFR AND P3P ALGORITHMS. BOTH METHODS USED NONLINEAR LEAST SQUARES REFINEMENT AFTER THE INITIAL RANSAC SOLUTIONS.

	Lounge		*Reception*		
	Image 1	Image 2	Image 1	Image 2	Image 3
Translation (cm)	16.1	12.4	10.3	6.0	67.6
Rotation (°)	4.3	3.9	1.5	2.2	5.4

human pose estimation. Despite several algorithms for non-overlapping camera calibration, this problem has always remained challenging due to many practical constraints. In this paper we addressed the problem of non-overlapping camera calibration by reducing the problem to localizing each camera with respect to the reconstructed 3D model obtained using an RGB-D SLAM system. This enables us to provide a model-guided browsing interface for visualizing the images obtained from the non-overlapping cameras.

Although the proposed method has obvious simplicity and practical advantages, it still suffers from a few limitations. First, the accuracy of our calibration method is bounded by the accuracy of the external RGB-D SLAM system. However, we believe that the accuracy of recent SLAM systems has reached a sufficient level to be used for the calibration purpose as demonstrated in this paper. Second, if the descriptor matching fails to identify the closest images to the query image, our method fails to estimate the correct pose. Placing some discriminative reference object in the FOV of the camera would resolve such cases.

ACKNOWLEDGMENT

We thank Jay Thornton for valuable discussion. This work was supported by and done at MERL.

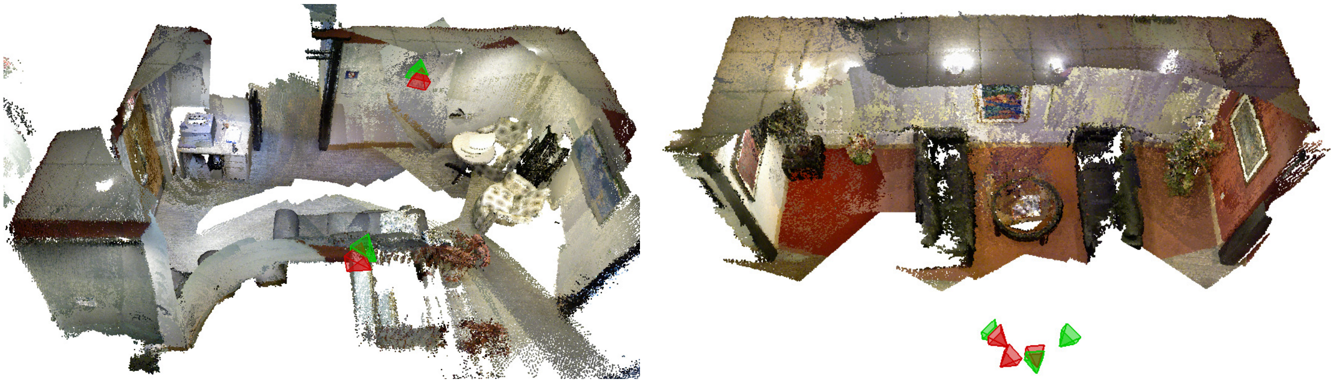

Figure 5. Comparison of the camera poses computed using P5Pfr (red) and P3P (green) for the *Lounge* (left) and *Reception* (right) scenes.

REFERENCES

[1] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[2] J.-Y. Bouguet, “Camera calibration toolbox for Matlab,” http://www.vision.caltech.edu/bouguetj/calib_doc/.

[3] D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easily calibrating omnidirectional cameras,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 5695–5701.

[4] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, May 2012, pp. 3936–3943.

[5] 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.

[6] 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.

[7] 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.

[8] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, “Point-plane SLAM for hand-held 3D sensors,” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5182–5189.

[9] 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.

[10] 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.

[11] M. Meilland and A. I. Comport, “On unifying key-frame and voxel-based dense visual SLAM at large scales,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 3677–3683.

[12] P. Lébraly, E. Royer, O. Ait-Aider, and M. Dhome, “Calibration of non-overlapping cameras - Application to vision-based robotics,” in *Proc. British Machine Vision Conf. (BMVC)*, Sep. 2010, pp. 10.1–10.12.

[13] G. Carrera, A. Angeli, and A. J. Davison, “SLAM-based automatic extrinsic calibration of a multi-camera rig,” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, May 2011, pp. 2652–2659.

[14] L. Heng, B. Li, and M. Pollefeys, “CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 1793–1800.

[15] S. Esquivel, F. Woelk, and R. Koch, “Calibration of a multi-camera rig from non-overlapping views,” in *Proc. DAGM Conf. Pattern Recognition*, 2007, pp. 82–91.

[16] R. Horaud and F. Dornaika, “Hand-eye calibration,” *Int’l J. Robotics Research*, vol. 14, no. 3, pp. 195–210, Jun. 1995.

[17] K. Daniilidis, “Hand-eye calibration using dual quaternions,” *Int’l J. Robotics Research*, vol. 18, no. 3, pp. 286–298, Mar. 1999.

[18] M. Knorr, W. Niehsen, and C. Stiller, “Online extrinsic multi-camera calibration using ground plane induced homographies,” in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, Jun. 2013, pp. 236–241.

[19] P. Sturm and T. Bonfort, “How to compute the pose of an object without a direct view?” in *Proc. Asian Conf. Computer Vision (ACCV)*, vol. II, Jan. 2006, pp. 21–31.

[20] R. K. Kumar, A. Ilie, J.-M. Frahm, and M. Pollefeys, “Simple calibration of non-overlapping cameras with a mirror,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2008.

[21] K. Takahashi, S. Nobuhara, and T. Matsuyama, "A new mirror-based extrinsic camera calibration using an orthogonality constraint," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012, pp. 1051–1058.

[22] J. A. Hesch, A. I. Mourikis, and S. I. Roumeliotis, "Extrinsic camera calibration using multiple reflections," in *Proc. European Conf. Computer Vision (ECCV)*, vol. IV, Sep. 2010, pp. 311–325.

[23] A. Agrawal, "Extrinsic camera calibration without a direct view using spherical mirror," in *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, Dec. 2013, pp. 2368–2375.

[24] R. J. Radke, "A survey of distributed computer vision algorithms," in *Handbook of Ambient Intelligence and Smart Environments*, H. Nakashima, H. Aghajan, and J. C. Augusto, Eds. Springer, 2010, pp. 35–55.

[25] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, Jun. 2004, pp. II–205–II–210.

[26] K. Tieu, G. Dalley, and W. E. L. Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," in *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, vol. 2, Oct. 2005, pp. 1842–1849.

[27] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun. 2004, pp. I–187–I–194.

[28] R. Pflugfelder and H. Bischof, "Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 709–721, Apr. 2010.

[29] B. Micusik, "Relative pose problem for non-overlapping surveillance cameras with known gravity vector," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2011, pp. 3105–3112.

[30] L. Heng, M. Bürki, G. H. Lee, P. Furgale, R. Siegwart, and M. Pollefeys, "Infrastructure-based calibration of a multi-camera rig," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2014.

[31] 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.

[32] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int'l J. Computer Vision*, vol. 13, no. 3, pp. 331–356, Dec. 1994.

[33] Z. Kukelova, M. Bujnak, and T. Pajdla, "Real-time solution to the absolute pose problem with unknown radial distortion and focal length," in *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, Dec. 2013, pp. 2816–2823.

[34] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 835–846, Jul. 2006.