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.
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.

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 cam-
era 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 recon-
structed 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 param-
eters 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 re-
sults [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 algo-

rithm 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 imple-
mented 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 descrip-
tors (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.
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 date keyframes that are close to the query image in terms of appearance using the VLAD descriptor. Then for each of the $K$ candidates, we add $N − 1$ ($= 2$) keyframes that are closest in terms of the keyframe 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.

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).

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 $\theta = \| \log(\hat{R}_1^T \hat{R}_2) \|_F / \sqrt{2}$, which is the angle of the rotation matrix required to transform one rotation matrix $\hat{R}_1$ to the other $\hat{R}_2$. 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
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.
REFERENCES


