2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

SLAM-Based Automatic Extrinsic Calibration of a Multi-Camera Rig

Gerardo Carrera, Adrien Angeli and Andrew J. Davison
Department of Computing,
Imperial College London.
180 Queen's Gate , U.K
{gcarrera, aangeli, ajd}@doc.ic.ac.uk

Abstract— **Cameras are often a good choice as the primary outward-looking sensor for mobile robots, and a wide field of view is usually desirable for responsive and accurate navigation, SLAM and relocalisation. While this can potentially be provided by a single omnidirectional camera, it can also be flexibly achieved by multiple cameras with standard optics mounted around the robot. However, such setups are difficult to calibrate.**

Here we present a general method for fully automatic extrinsic auto-calibration of a fixed multi camera rig, with no requirement for calibration patterns or other infrastructure, which works even in the case where the cameras have completely non-overlapping views. The robot is placed in a natural environment and makes a set of programmed movements including a full horizontal rotation and captures a synchronized image sequence from each camera. These sequences are processed individually with a monocular visual SLAM algorithm. The resulting maps are matched and fused robustly based on corresponding invariant features, and then all estimates are optimised full joint bundle adjustment, where we constrain the relative poses of the cameras to be fixed. We present results showing accurate performance of the method for various two and four camera configurations.

I. Introduction

It is now widely accepted that cameras are highly suitable sensors for providing mobile robots with the situational awareness they need to navigate and operate autonomously in the real world. Although the geometrical shape of the scene around a robot does not immediately pop out of cameras in the same way as from a depth sensor such as a laser range-finder, it has been convincingly shown that with appropriate processing they can be used both to estimate motion accurately and recover detailed 3D geometric information about a scene (e.g. [17]). This seems to be only the beginning of the role cameras can play in robotics, however, since ongoing progress in computer vision indicates that the detailed photometric information they offer has almost unlimited potential for problems such as dense volumetric reconstruction and object recognition.

A clearly desirable characteristic of a sensing system for situational awareness is a wide field of view, and the advantages of wide-angle vision have been proven for instance both in geometric SLAM [10] and appearance-based place recognition. Wide angle vision can be achieved by single cameras with special optics such as fish-eye lenses or catadioptric systems, which can sometimes be elegant and practical. However, all such cameras will suffer from relatively poor angular resolution, since the pixels available must be spread thinly. Another alternative is an active camera system, with a mechatronic platform providing pan/tilt rotation of a narrow field of view camera; but this option suffers from the serious disadvantage that views of highly separated angles cannot be simultaneous and the robot loses the 'eyes in the back of its head' that true omnidirectional vision provides.

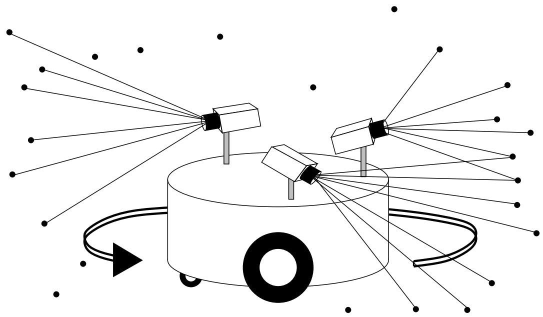

Fig. 1. Our goal is automatic extrinsic calibration of an arbitrary set of cameras mounted rigidly on a mobile robot and without external infrastructure. The cameras may or may not have overlapping fields of view, but will observe common parts of the surrounding scene as the robot makes a loopy motion including 360° rotation. We make monocular SLAM maps of the invariant feature points observed individually by each camera, match features robustly, perform 3D alignment to fuse these into a single joint scene map, and globally optimise enforcing rig rigidity.

Now that cameras and their supporting electronics are getting cheaper and smaller, mounting multiple cameras around a robot in a ring or other configuration is highly feasible, clearly offering the potential for both a wide field of view and high angular resolution. A multi-camera rig will in general not be fully equivalent to a single wide-angle camera in terms of imaging geometry since it will have multiple optical centres, but this is not a problem for applications such as navigation as long as the locations of those optical centres is known, and may even be an advantage in inferring depth information. Indeed, a multi-camera rig is fully reconfigurable and can be set up to provide the field of view needed, and if necessary regions where camera views overlap for stereo observation.

For a multi-camera rig to be useful, it is critical to know the relative poses of the cameras. The aim of this paper is therefore to propose an automatic procedure to achieve this

This work was supported by a CONACYT scholarship from the Mexican government to G. Carrera and European Research Council Starting Grant 210346

978-1-61284-385-8/11/$26.00 ©2011 IEEE

extrinsic calibration of the rig, by using SLAM techniques as a robot makes a small set of programmed motions through a natural scene. Multi-camera rigs, particularly those involving extruding cameras and mounts, are prone to moving as the result of vibrations or knocks. While it would be ultimately desirable to detect and correct this online during normal operation, the next best is to have an automatic calibration procedure which can easily be run whenever needed. Note that commercial multi-camera systems such as the Point Grey Ladybug, which are factory-calibrated, feature expensive, heavy and bulky rigid casings to ensure that the calibrated configuration does not change once the rig is out in the real world. In such a systems, a whole calibration room, with targets in all directions would be needed for pattern-based calibration. Our approach tackles the non-straight forward part of multi-camera calibration, the extrinsics of non-overlapping cameras.

Our method (outlined briefly in Figure 1):

- Is fully automatic, needing no human intervention between starting the robot motion, image capture and when the calibration result is returned.
- Requires no *a priori* known scene structure.
- Solves primarily for the relative poses (up to scale) of the cameras, which can be arbitrarily located and oriented in 3D on the robot's structure.
- Assumes known individual intrinsic parameters.
- Assumes synchronised capture across the multiple cameras, but they can have varying types and optics.

II. Related Work

To estimate intrinsic camera parameters, classic methods make use of calibration patterns to provide external metric information [22] and are well-tested and accurate. However, it is also possible to *auto-calibrate* camera intrinsics without a pattern if additional assumptions are made, most usually that several images of a rigid scene taken by a camera with unchanging parameters are available and between which correspondences can be obtained. Under such conditions, enough constraints are available from image feature measurements to uniquely determine the calibration constants. For instance, Pollefeys *et al.*[19] solved for intrinsic camera parameters as part of a complete off-line structure from motion algorithm. A more recent approach using SLAM-based probabilistic filtering allowed auto-calibration estimates of camera internal parameters to be refined sequentially [3].

Most work on extrinsic calibration has focused on the case of stereo rigs, designed such that two or sometimes more cameras have highly overlapping views and simultaneously observe mostly the same parts of a scene. Calibration methods based on external metric targets are relatively straightforward to formulate for this problem, allowing both intrinsic and extrinsic calibration to be unified (e.g. [6]), and are therefore prevalent practically. More relevant to the goal of our paper are methods which can perform extrinsic auto-calibration of stereo rigs presented with general, unknown scenes. For instance, extending the approach of [22], Luong and Faugeras [16] proposed to calibrate a stereo rig using point correspondences between the two cameras when the system undergoes a series of displacements in an *a priori* unknown rigid scene. The resulting estimated extrinsic parameters are determined up to a scale factor. Similar work on solving the stereo-camera calibration problem using correspondences information includes [11]. Another interesting approach is also presented in [20] where a central EKF-SLAM filter fuses the information coming from a stereo camera rig to obtain the extrinsic parameters.

There has been relatively little work on auto-calibration in the case we are considering in this paper, multi-camera rigs designed primarily to provide a wide field of view and which may therefore have little or even no image overlap. In fact, even using metric patterns to calibrate such rigs is challenging because the patterns must accurately surround the whole rig in the form of a 'calibration room' or similar. Different solutions have been proposed to achieve such behavior. For instance, in order to accurately calibrate an eight camera rig for SLAM purposes, Kaess and Dellaert [10] designed a semi-automatic procedure which involved placing the robot in a special scene with calibration targets on three walls and capturing a number of images manually before feeding these to automatic target matching and optimisation procedures. A complicated semi-automatic procedure was also proposed in [9], where a calibration grid is presented in front of each camera individually, at several positions, while a laser measurement system is used to determine the 3D coordinates of the grid's corners. From this geometrical information, and taking advantage of point correspondences over images of the grid in the different cameras, calibration parameters can be determined.

The two above approaches require significant user implication to obtain the calibration parameters. A simpler yet still semi-automatic solution was proposed by Li *et al.* [13], where calibration can be achieved in unprepared natural or man-made scenes. A three-step procedure sequentially determines the centre of distortion of the setup (assuming a single unique optic centre for all the cameras), the individual intrinsic parameters of each camera, and the relative camera orientations (under a unique optic centre assumption, where relative camera poses differ only by a rotation). The procedure however still requires user intervention, with the manual specification of point correspondences between the views of two neighboring cameras. In one more recent solution a planar mirror is used to make a single calibration object visible to all the cameras. However, even though the method allows for both intrinsic and extrinsic calibration, the approach would appear inconvenient in practise.

The work we have found which is most similar to our current approach is that by Esquivel *et al.* [5], who explicitly tackle auto-calibration of a non-overlapping rig and present some limited results obtained using real image sequences. They perform individual structure from motion computation for each camera separately, determining a trajectory for each as the rig moves. These trajectories are then locally aligned in 3D and the transformation determined estimate of the relative camera poses. A similar method that tackles the problem by tracking a moving object and matching trajectories is presented in [1]. Also, in [18] a system with two wearable forward and backward facing stereo is developed. The individual stereo rigs are calibrated in a standard way using a calibration pattern. The relative calibration between the rigs is achieved automatically via trajectory matching.

These techniques are more limited since they rely only on trajectory matching rather than aiming to build a globally consistent feature map within which the multi-camera rig is located as our method does. Our method, via its final global optimisation, implicitly takes advantage of the same trajectory matching behaviour, but is able to fully digest all of the other information available.

A final relevant paper is the recent work of Koch and Teller [12], who automatically learn about how feature motion in the images of a multi-camera rig relates to the rig's motion as it is carried by a person, but stop short of aiming to determine full geometric extrinsic calibration.

III. Method

Our method in detail consists of the following steps:

1) Cameras are attached to a mobile robot in arbitrary positions and orientations, and locked into place. After this, all the following operations are fully automatic.
2) The robot makes short pre-programmed movement, such as turning on the spot through 360° rotation or driving in a small circle, capturing synchronised video streams from the cameras as it moves through an unprepared environment.
3) The video sequence from each camera is fed to a modified version of the MonoSLAM algorithm [4]. Individually for each camera, MonoSLAM estimates camera motion and builds a 3D map of visual feature locations up to scale. Each feature is characterized by a SURF descriptor [2].
4) Each full video sequence is decimated into regularly spaced keyframes. Each camera's map and motion estimates are then refined individually using bundle adjustment over these keyframes.
5) Candidate feature correspondences between each pair of individual maps are obtained via thresholded matching between their SURF descriptors.
6) Initial alignment of the maps' relative 3D pose and scale is achieved using 3D similarity alignment and RANSAC [7], with sets of three correspondences between maps derived from the SURF matches used to generate hypotheses. A reliable set of correspondences between each pair of maps, satisfying both descriptor matching and geometrical correspondence, is deduced, and the monocular maps are fused into a single joint map each of whose features has been mapped by one or more cameras.
7) This initial joint map is used as the starting point for full bundle adjustment optimisation to estimate the relative poses of the cameras, 3D positions of scene points and motion of the robot.

We will now explain each of these steps in detail.

A. Robot Motion and Sequence Capture

The set of cameras whose extrinsic calibration is required are fixed to the robot in the desired configuration, and it commences a set of pre-programed movements, controlled by odometry, of duration around 1–2 minutes. The robot captures a synchronised sequence of images from each camera, typically at a fixed frame-rate.

When the cameras used are of the same type, it is often possible to synchronise capture using capabilities provided by the manufacturer — many IEEE 1394 cameras for instance will sychronise by default when attached to the same bus. In the case that cameras of different types or otherwise unsynchronisable cameras were required to be used, a somewhat tedious but still fully automatic procedure to achieve the same effect would be to program the robot to move in a stop-start fashion, making many small motions instead of a single continuous one and stopping to request one image from each of the cameras every time it stopped.

There are no absolute requirements for the type of robot motion used; there are many different types of robot (wheeled, legged, flying…) with different type of movement, and our method is in principle applicable to any of these. However, there are certain characteristics which are desirable. To be practical and straightforward the motions used should be simple and short, but from an estimation standpoint there is a clear advantage to 'loop closing' motions where the robot makes a full rotation while moving. As is well understood in SLAM research, this will remove drift and permit tightly converged individual camera maps. Also, it maximises the chances of many feature correspondences between the multiple cameras since cameras pointing at different horizontal angles will come to view all of the same scene elements (occlusion notwithstanding). Even if there is no overlap between the images simultaneously captured from the different cameras, there will be overlap between the monocular *maps* built.

Similarly, our algorithm places no specific requirements on the scene around the robot such as the presence of calibration objects, but there are some favourable characteristics which will lead to easier and more accurate operation. A highly textured scene which offers many trackable features across the whole field of view of the cameras is best, and it will also be helpful if this texture is not overly repetitive to aid unambiguous matching. The depth of the main elements of the scene should ideally also be relatively small compared to the size of the robot's motion in order that parallax can be observed and 3D locations of the features determined.

B. Single Camera Mapping Using MonoSLAM

The first processing step of our method consists of estimating the structure of the scene and the robot's motion from each camera individually. For this we use a slightly modified version of MonoSLAM [4], a sequential 3D visual SLAM algorithm for monocular cameras based on the EKF. MonoSLAM is able to routinely build maps within room-sized domains, including the capability to automatically close loops induced by rotation (Figure 2 (top row)).

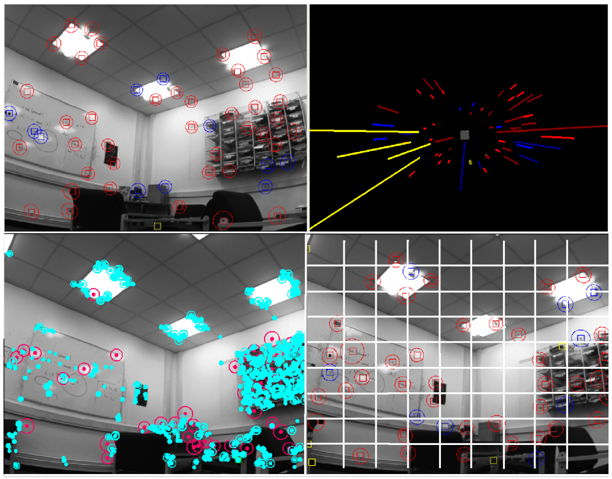

Fig. 2. Top: MonoSLAM in operation, showing processing of an image stream from a single camera on a robot which has made nearly a full 360° rotation, highlighting feature tracking and 3D map creation. Bottom: adding SURF features to a MonoSLAM map: from all the SURF features detected in a frame (left), up to one new feature from each cell of a regular grid is added to the map (right).

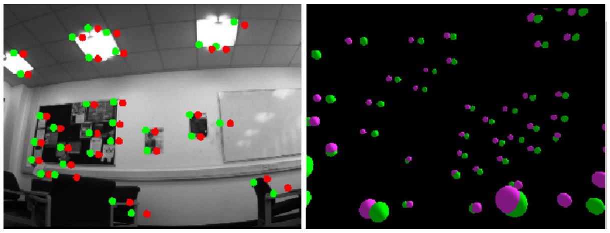

Fig. 3. Bundle adjustment of an individual camera map generated by MonoSLAM. Here the 3D map generated by MonoSLAM and that after bundle adjustment is shown reprojected into one of the keyframes (left) and in 3D (right). In each view, the green points are after bundle adjustment and red/purple are before. Reprojection accuracy across the full keyframe set clearly improves after bundle adjustment.

The first modification is the use of SURF features [2] rather than the Shi-Tomasi image patches MonoSLAM normally relies on. The motivation of this modification is to allow for invariant feature description, which will be crucial for the later step of getting map-to-map correspondences. We choose SURF because of its proven recognition performances and reasonable computational demand.

The second modification is to MonoSLAM's map management policy, also with the aim of improving inter-map matching performance. Instead of the random feature initialisation strategy originally used in MonoSLAM, the image is divided into a regular grid as shown in Figure 2 (bottom row). At each frame of the sequence, if one of the cells of this grid is unoccupied by a feature in the existing map, the most salient SURF feature detected in that region is initialised into the map (i.e., the one with the highest Hessian score, see [2] for details). By ensuring that features are initialised right across the field of view of each camera, we maximise the chances of detecting a decent number of corresponding features between the individual maps. In order to ensure that the feature scale do not affect significatly in the measurement accuracy, we only use SURF features from the first 2 octaves of the image pyramid, which are therefore well located in the images

C. Bundle Adjustment of Single Camera Maps

MonoSLAM uses EKF processing and outputs filtered maps which may include linearisation errors. Therefore, we look to refine the maps for each camera individually to get as much accuracy as we can in the structure and camera poses of each map before attempting 3D alignment. We use the free software library SBA for this purpose [14].

To save on computational load at the expense of a relatively small loss in accuracy, at this stage we decimate the full image sequences processed by MonoSLAM into *keyframes* regularly spaced in time, and in the rest of the method we use only these. By controlling robot's motion at a constant speed we ensure that the keyframes are spatially well distributed. It has been well proven in small scale structure from motion that a set of well chosen keyframes is quite sufficient for accurate camera motion estimation and scene reconstruction so we are confident that regularly decimating the sequence is a very sound procedure with regard to accuracy. This has most recently been confirmed in [21].

Each keyframe image is saved along with the 2D feature measurements made in that image by MonoSLAM and the camera position estimate from MonoSLAM at that time instant. After the robot completes its trajectory, the vector of 3D features position estimates $\hat{Y}$, the estimated camera poses $\hat{C}$, and the vector of 2D keyframe measured feature positions X are used in a bundle adjustment (BA) procedure for map refinement. The above quantities, which are input to the BA procedure, can be written in expanded vector form as follows:

$$\begin{aligned} \hat{C} &= [\hat{c_0}, \dots, \hat{c_m}] \\ \hat{Y} &= [\hat{y_0}, \dots, \hat{y_n}] \\ X &= [x_{00}, \dots, x_{0m}, \dots, x_{10}, \dots, x_{1m}, \dots, x_{nm}] \end{aligned} \quad , \qquad (1)$$

where $\hat{c_i}$ is the estimated 6DoF camera pose of keyframe i, $\hat{y_j}$ is the estimated 3D position of feature j, and x_{ij} is the measurement corresponding to feature j in the image of keyframe i. The prediction $\hat{x_{ij}}$ of this measurement can be obtained through a standard camera projection function $h(\hat{c_i}, \hat{y_j})$, while the noise associated with this measurement, modeled by the Gaussian distribution $N(0, \sigma_x^2)$, is represented by the 2×2 matrix $\Sigma_{x_{ij}} = \text{diag}(\sigma_x^2, \sigma_x^2)$. Concatenating all predicted measurements in a single vector leads to the estimated measurement vector $\hat{X}$, with corresponding measurement noise encoded by the diagonal matrix Σ_X formed by concatenating all individual $\Sigma_{x_{ij}}$ matrices. Then, the vector of parameters to be optimized is given by $\hat{P} = [\hat{C}, \hat{Y}]$. In BA we look to minimize the Mahalanobis distance $\epsilon^T \Sigma_X^{-1} \epsilon$ (i.e., the weighted re-projection error), where $\epsilon = X - \hat{X}$ [14].

The results of individual map bundle adjustment can be confirmed if necessary from a console within our application whereby the keyframe sequences from each camera are easily browsable with a slider and the reprojections of the optimised point locations can be checked for stability (see Figure 3).

D. Inter-Map Correspondences from SURF Descriptors

Now that we have built individual feature maps for each camera, we must start to align them in order to produce a single joint map of the environment around the robot. First we find candidate correspondences between the features in each pair of individual maps. Exhaustively, the SURF descriptors of all features f_a of map A are compared with those of all features f_b of map B. During this procedure, the closest match in map B for a given feature f_a in map A is determined by the *closest-to-second-closest* ratio proposed in [15], by comparing (using the L2 norm) the SURF descriptors associated to the features.

E. Confirming Correspondences and Finding Initial 3D Map Alignment using RANSAC

The goal of this step is to determine which of the inter-map correspondences between two maps suggested by SURF matching are geometrically consistent and therefore highly likely to be correct, and to compute an initial estimate of the cameras' relative poses. From a set of at least three feature correspondences between any two maps, the rigid body similarity transformation between those maps (and therefore between the cameras which made them) can be hypothesized. We therefore run a RANSAC procedure where at each iteration three candidate SURF correspondences are chosen, a similarity transformation is calculated, and then the rest of the candidate correspondences are checked for consistency with this (via a Euclidean distance threshold) when the transformation is applied. Note that the potential transformation between the maps is a similarity (we must solve for scale as well as rotation and translation since) they have been built by monocular cameras.

While there are various techniques for aligning 3D point sets, here we use generic non-linear optimisation via the Levenberg-Marquardt algorithm as in [8] to minimise the distance between transformed points, as computed by the following similarity transformation:

$$(y_i)_A = s\mathtt{R}_A^B (y_i)_B + t_A^B \ , \tag{2}$$

relating a feature's coordinates y_i in maps A and B via rotation matrix $\mathtt{R}_A^B$, translation vector t_A^B and scale change s.

This alignment calculation is carried out for each random choice of three correspondences, and inlier scores for each candidate correspondence are counted up until a maximum number of RANSAC iterations. The final set of correspondences with the highest inlier counts are then used to make a final similarity estimate between the pair of maps. In experiments we normally obtain between 10 and 20 accurate correspondences between each pair of maps.

F. Global Bundle Adjustment

Now that the individual maps from each camera have been aligned in 3D, an initial estimate is available of the relative pose of the cameras on the robot. However, this estimate relies heavily on the correspondences which have been obtained between the maps using SURF and RANSAC, and the number of these correspondences may be quite low. As a consequence, the resulting transformation may not be very accurate. There is still potential for improvement, by applying global bundle adjustment of robot motion, scene structure and relative camera poses together. Importantly, in this optimisation we enforce the constraint that the relative camera poses are constant, this being enabled since we know that our keyframes were captured in a synchronised manner.

After alignment and matching of the individual camera maps, we can work with the concept of a single joint map which is the union of the individual maps. Many of the features in this map will have only ever been observed by one of the cameras (features with no inter-map correspondences), while others (those successfully matched in the previous map alignment step) will have been observed by two or more cameras. Nevertheless, *all* of the features and *all* of the measurements in keyframes are useful in joint optimisation.

We formulate global bundle adjustment with the following state and measurement vector elements:

1) The set of estimated camera poses $\hat{C} = [\hat{c_0}, \ldots, \hat{c_m}]$ associated to the keyframes of one camera, called the *reference camera*[1]. In particular, the pose $\hat{c_0}$ will define the origin of the absolute reference frame for all the estimated quantities.
2) The set of estimated relative camera poses $\hat{T} = \{\hat{T^i}\}$, where $\hat{T^i}$ is the estimated rigid body transformation from the reference camera to the i^{th} camera on the robot. Note that each transformation is a 6DoF transformation made up of rotation $\hat{R^i}$, translation $\hat{t^i}$ components. Because all the features are expressed in global coordinates, the scale between maps is not taking into account this time. Note also that the estimated transformations do not depend on time, modeling the fact that cameras are rigidly fixed on the rig.
3) The set of 3D coordinates estimates for the features of the joint map $\hat{Y} = [\hat{y_0}, \ldots, \hat{y_n}]$.
4) The set of measurements $X = \{(x_{ij})_k\}$, where $(x_{ij})_k$ is the measurement corresponding to feature j in the i^{th} keyframe of camera k. The prediction $(\hat{x_{ij}})_k$ of a measurement can be obtained through the composition of a similarity transformation and a standard camera projection $h_k(\hat{c_i}, \hat{y_j}, \hat{T^k})$: the feature $\hat{y_j}$ must first be transformed into the coordinate frame of the i^{th} keyframe of camera k, i.e. by the means of $\hat{c_i}$ and T^k, before being projected into the corresponding image.

The vector of parameters to be optimised is given by $\hat{P} = [\hat{C}, \hat{T}, \hat{Y}]$. Again, we look to minimise the Mahalanobis distance $\epsilon^T \Sigma_X^{-1} \epsilon$ (i.e., the weighted re-projection error), where $\epsilon = X - \hat{X}$. Correctly formulating bundle adjustment in this way means that all of the useful information can be sucked out of the data available. The relative poses of the robot's cameras are estimated not just based on the structure of the scene, but also based on the motion of the robot, by

[1] Note that any of the cameras mounted on the robot can be selected here.

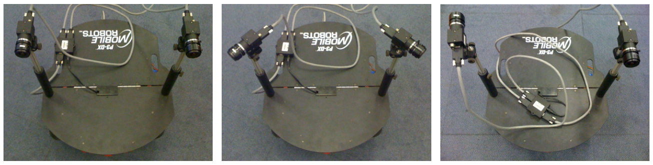

Fig. 4. The three experimental configurations of a two-camera rig, with relative horizontal angles of 0°, 90° and 180°.

enforcing the rigidity of the relative camera transformations over the trajectory of the robot.

IV. Experiments

As our method is valid for any number of cameras mounted rigidly on a robot, we have currently been able to test it with different configurations of two cameras, and a four camera set up. We ran the experiments in two normal university rooms with no modification and each with size around 5×5 metres. The robot's pre-programmed motion was as a turn on the spot controlled in a saw-tooth manner, such that the robot would rotate by left 90°, right 45°, left 90° again and so on until it had completed somewhat more than a full rotation.

In most experiments, this motion was completed in around 1 minute and image capture was at 15Hz. After execution of MonoSLAM, the sequences were decimated by a factor of 40 to leave only keyframes to be used in the rest of the algorithm. We used a set of Point Grey Flea2 cameras, which are able to synchronise automatically across the IEEE 1394b bus. The cameras were fitted with identical wide angle lenses giving a field of view of around 80°, having had their intrinsic parameters (including substantial radial distortion) calibrated individually using a standard calibration pattern-based method.

To perform verification of our experimental calibrations against ground truth, we used the commercial photogrammetry software Photomodeler (Eos Systems Inc.) to manually make a surveyed model of the rooms in which experiments were carried out from a set of high resolution digital photographs, with additional scale information provided by manual measurements. Images taken from our camera rig were then matched into this model to obtain estimates of the camera positions within the room and therefore their relative pose.

A. *Different Configurations of Two Cameras*

The camera calibrations used in our three experiments with two cameras are shown in Figure 4, and we present results from each of these in the following sections.

1) Parallel Cameras Facing Forwards: The first experiment, to confirm operation in a standard binocular stereo camera configuration, featured approximately parallel cameras facing forwards mounted on the left and right of the robot. All steps of the algorithm were carried out as explained in the previous sections, and we concentrate on the results of the final global bundle adjustment.

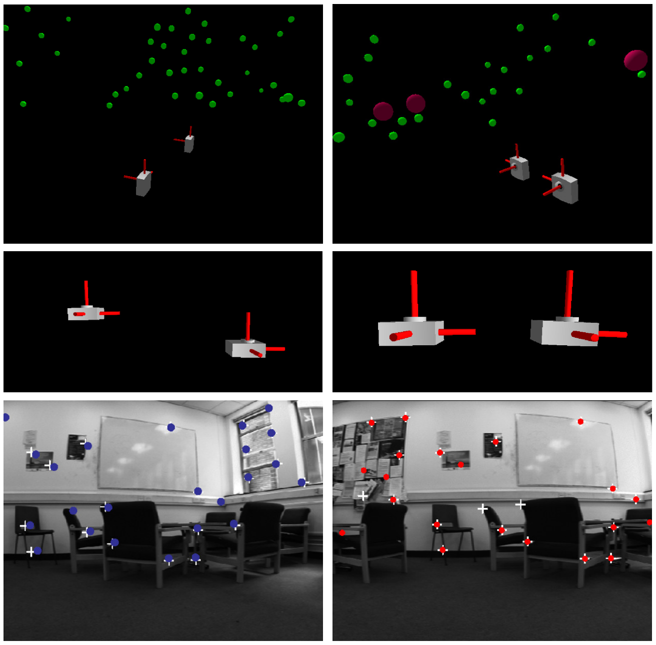

Fig. 5. Effects of the final BA procedure, parallel cameras sequence: the images of the first and second row show the improvement in the estimated transformation between the two cameras (left column: before BA; right column: after BA), while the images of the second row show the reduction in reprojection error (see text for details).

Figure 5 demonstrates the effects of the final BA procedure. The first row of the figure shows a 3D view of the camera rig before (left) and after (right) the final BA. As it can be seen, the transformation between the two cameras is improved after the final BA: in the right image, the cameras look coplanar, whereas in the left image, one camera is slightly in front of the other. The improvement is proven quantitatively in the second row of the figure, which shows the re-projection errors of features into the maps, taking into account all of the estimated transformations, in one video frame before and after the final global BA. After BA (right), the projected map points (circles) match more accurately the image positions of the measurements (crosses) than in the left image. The mean squared reprojection error achieved with this configuration was 0.90 square pixels.

2) Non-Overlapping Cameras Separated by 90°.: In the second experiment, the cameras faced horizontally but at around 45° to the left and right of the forward direction of the robot respectively. Figure 6 shows the improvement in reprojection error during global BA in this experiment, achieving an mean squared error of 1.4 square pixels.

3) Non-Overlapping Cameras Separated by 180°.: In the third experiment the cameras were mounted horizontally on the left and right of the robot, but with the right camera pointing approximately forwards and the left camera backwards. In Figure 7 we can see that after global BA the estimate of the cameras' relative pose was accurate, with a mean squared reprojection error of 0.807 square pixels.

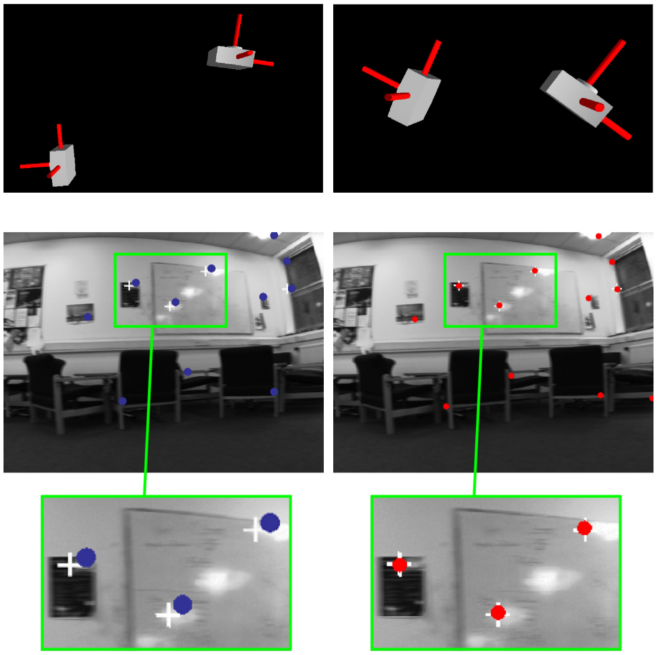

Fig. 6. Effects of the final BA procedure, 90° cameras sequence in terms of camera pose and reprojection error (left: before BA; right: after BA.)

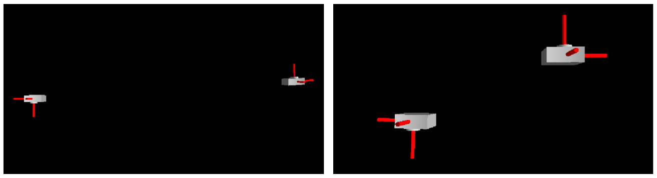

Fig. 7. Effects of the final BA procedure, 180° cameras sequence in terms of camera pose and reprojection error (left: before BA; right: after BA.)

B. Omnidirectional Four Camera Configuration

In this experiment we show that our method is applicable to any number of cameras rigidly attached to the robot. We set up two slightly divergent stereo pairs with no overlap between the front pair and the back pair, as illustrated in Figure 8. As in our previous experiments, we performed all the stages mentioned in section III including a full optimisation in which the global map, the motion and the transformations between each camera and the reference frame are globally optimised.

	Rig Configuration with Two Cameras		
	Parallel	**90°**	**180°**
Photomod	2.88° ± 0.5°	94.72 ° ± 0.5°	176.14° ± 0.5 °
Our System	1.38° ± 0.22°	94.10° ± 0.83°	174.69° ± 0.43°

TABLE I

ANGLES BETWEEN CAMERAS ESTIMATED BY OUR APPROACH AND PHOTOMODELER FOR THE THREE TWO-CAMERA EXPERIMENTS.

	Rig Configuration with Four Cameras		
	Camera$_{1\_2}$	Camera$_{1\_3}$	Camera$_{1\_4}$
Photomodeler	25.20°	173.76°	156.96°
Our System	25.30°	174.71°	157.20°

TABLE II

ESTIMATED ANGLES BETWEEN CAMERA 1 (REFERENCE CAMERA) AND THE OTHER THREE CAMERAS FOR THE FOUR CAMERA EXPERIMENT.

C. Ground Truth Comparison

Our system computes the full relative 3D pose (up to scale) between the cameras in each experiment, but for ground truth comparison the most straightforward parameter to analyse is the total relative rotation angle between the two camera frames. For each of the experimental scenarios presented above, we computed this angle and compared it with the relative angle determined by registering several pairs of camera images within the 3D model of the room obtained using Photomodeler. Table I presents for comparison the angles obtained with our system and Photomodeler for the two camera configurations, including an assessment of the standard deviation we observed over multiple runs of our whole system on the same image data.

We certainly see gross agreement, but we are inclined to believe that the not insignificant different is due to limitations of our use of Photomodeler rather than a weakness of our method. The Photomodeler estimates were achieved from matching in just a few image pairs from our robot against a relatively sparse 3D point model, whereas the estimates from our method were achieved from global optimisation of long sequences with all the correct constraints applied. We are therefore inclined to put much more trust in the reprojection measures we obtained in experiments, all of which were in the range 0.8–1.5 pixels RMS. Considered as an angle in our 640 × 480 resolution images over an 80° field of view, this corresponds to an angular error of around 0.1° and we believe that this is much more indicative of the accuracy of our method.

For the four camera configuration we present results comparing angles between our reference frame (camera$_1$) and the other cameras (see Table II). We can see that the differences between the angles obtained by Photomodeler and our system are smaller than for the two camera configuration, indicating as expected that using more cameras adds more constraints to the final global optimisation.

V. CONCLUSIONS

We have presented a fully automatic algorithm for general multi-camera rig calibration which does not require calibration targets or other infrastructure. Well known SLAM techniques are used to build monocular feature maps as the robot makes controlled movements. These maps are matched and aligned in 3D using invariant descriptors and RANSAC to determine the correct correspondences. Final joint bundle adjustment is then used to refine estimates and take account of all feature data. We have demonstrated the accurate ability

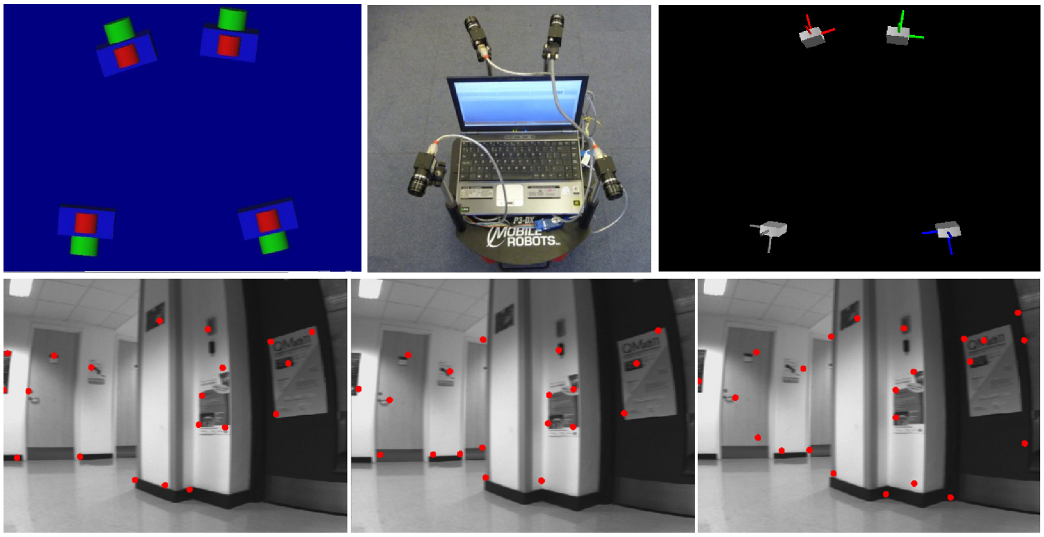

Fig. 8. Experiment with an omnidirectional four camera configuration (top-left: manual Photomodeler, top-middle: photograph of robot, top-right: automatic result from our system). The bottom images show the reprojection of features in the maps of cameras 2, 3 and 4 respectively from the global map into keyframes of camera 1. Note the high quality alignment between these points and scene corners, indicating excellent registration accuracy.

of the method to recover the configuration of a camera rig with two and four cameras in a variety of configurations.

It would be interesting and beneficial to determine the intrinsic parameters of the individual cameras as part of the full calibration procedure, rather than requiring them to be known *a priori*. It would be straightforward to include and optimise these parameters in the final bundle adjustment step, but the problem is that the construction of individual camera maps using MonoSLAM would be inaccurate without well known intrinsics for each camera. This could be tackled using the approach recently proposed by Civera *et al.* [3] for sequential auto-calibration of a single camera.

References

[1] R. Angst and M. Pollefeys. Static Multi-Camera Factorization Using Rigid Motion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, 2009.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[3] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel. Camera self-calibration for sequential bayesian structure from motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[4] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007.

[5] S. Esquivel, F. Woelk, and R. Koch. Calibration of a multi-camera rig from non-overlapping views. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2007.

[6] O. D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1986.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2001.

[9] S. Ikeda, T. Sato, and N. Yokoya. A calibration method for an omnidirectional multi-camera system. In *Proceedings of SPIE Electronic Imaging*, 2003.

[10] M. Kaess and F. Dellaert. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding (CVIU)*, 2009.

[11] J. Knight and I. Reid. Binocular self-alignment and calibration from planar scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2000.

[12] O. Koch and S. Teller. Body-relative navigation guidance using uncalibrated cameras. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[13] H. Li, R. Hartley, and L. Wang. Auto-calibration of a compound-type omnidirectional camera. In *Proceedings of the Digital Image Computing on Techniques and Applications (DICTA)*, 2005.

[14] M. I. A. Lourakis and A. A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009.

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

[16] Quang-Tuan Luong and Olivier Faugeras. Self-calibration of a stereo rig from unknown camera motions and point correspondences. Technical Report INRIA Tech. Report RR-2014, INRIA Sophia-Antipolis, 1993.

[17] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo SLAM system. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[18] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[19] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1998.

[20] Joan Solà, Andr Monin, Michel Devy, and Teresa Vidal-Calleja. Fusing monocular information in multicamera slam. *IEEE Transactions on Robotics*, 24(5):958–968, 2008.

[21] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[22] R. Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1986.