WIDE-AREA EXTERNAL MULTI-CAMERA CALIBRATION USING VISION GRAPHS AND VIRTUAL CALIBRATION OBJECT

Gregorij Kurillo, Zeyu Li, Ruzena Bajcsy

University of California, Berkeley

ABSTRACT

In this paper we address external calibration of distributed multi-camera system intended for tracking and observing. We present a robust and efficient method for wide area calibration using virtual calibration object created by two LED markers. Our algorithm does not require for all the cameras to share common volume; only pairwise overlap is required. We assume the cameras are internally calibrated prior to deployment. Calibration is performed by waiving the calibration bar over the camera coverage area. The initial pose of the cameras is calculated using essential matrix decompositions. Global calibration is solved by automatically constructing weighted vision graph and finding optimal transformation paths between the cameras. In the optimization process, we introduce novel parametrization for two-point calibration using direction normal. The results are increased accuracy and robustness of the method under the presence of noise. In the paper, we present experimental results on a synthetic and real camera setup. We have performed image noise analysis on a synthetic wide-area setup of 5 cameras. Finally, we present the results obtained on a real setup with 12 cameras. The results obtained on the real camera setup show that our approach compensates for error propagation when the path transformation includes two to three nodes. No significant difference in reprojection error was found between the cameras on non-direct and direct path of the vision graph. The mean reprojection error for the real cameras was below 0.4 pixels.

***Index Terms*—** External camera calibration, vision graph, multi-camera system, epipolar geometry

1. INTRODUCTION

Important step in deployment of multiple cameras is their localization. For many applications, only approximate position and orientation of the camera are needed. Many researchers have addressed the issue of localization using different techniques, such as use of acoustic delays, radio frequency intensity, image based localization and others. Required accuracy of the camera localization depends on the application requirements.

However, many applications of tracking and observing dynamic targets require multiple cameras to be accurately calibrated to a global coordinate system. Accurate calibration is especially important in tracking of passive or active markers using vision based methods for applications of virtual (VR) or augmented reality (AR). Multiple cameras are used to track small set of markers to calculate user's body or head position and/or orientation in real space and map it into the virtual space.

Existing firewire cameras often used for such applications can capture medium to high resolution images with 30-100 FPS, requiring high bandwidth for image transfer. The bandwidth limitations restrict number of cameras used or lower the resolution and frame rate of the transferred image data. With the increasing power of microprocessors of smart cameras, much of the processing needed for tracking applications can be implemented on board of the cameras. Detected marker locations can be thus sent through the network to the processing server. Such small packets allow high-speed transfer of data from the smart camera also over wireless protocols allowing more flexibility in camera arrangement.

To achieve proper mapping of 3D location from the physical space to the virtual space, accurate position and orientation of the cameras have to be known in addition to camera internal parameters (i.e. focal length, image center, distortion). The internal calibration can be performed in advance, before the camera is deployed, while the external calibration requires on-site localization.

In this paper we propose geometric calibration of multiple cameras, which can be arranged over a wide area without imposed constraints for all cameras to share a common volume. Our approach assumes that at least any two given cameras overlap and that the cameras have been internally calibrated prior to deployment. In our algorithm the cameras are externally calibrated using two moving LED markers positioned at a fixed distance. The moving markers generate a virtual calibration object in 3D space which is captured as a time sequence. The use of virtual calibration object improves robustness of point detection and point correspondence between cameras while reducing computational load on the capturing side.

Global calibration is solved by constructing vision graph and determining the optimal transformation paths from each camera to the reference camera. Finally, the parameters are optimized using sparse bundle adjustment implementation.

978-1-4244-2665-2/08/$25.00 ©2008 IEEE

We demonstrate the effectiveness of the proposed algorithm on a set of regular cameras while considering implementation in the smart cameras. This paper is organized as follows. In Section 2 we present some of the previous research in the area of camera network calibration. Section 3 describes our calibration approach. In Section 4 we present the results of the calibration on a synthetic and real multi-camera setup. Finally, Section 5 concludes the paper with discussion on advantages and drawbacks of the presented calibration method.

2. RELATED WORK

Multiple camera calibration has been studied extensively in the past decades. In this paper we review selected number of studies related to our work. Calibration of distributed cameras using image-based localization has been described by Mantzel and colleagues [1], who calibrated a network of cameras with sparse overlapping by acquiring planar checkerboard images. They combined cameras with overlap into microclusters and refined the localization error using planar metrics. Their approach required checkerboard image to be fully visible simultaneously in several cameras. Similar approach was applied by Olsen and Hoover [2] who calibrated a camera network with small overlap in workspace using planar domino grid.

In several studies, calibration targets have been substituted by the methods of self-calibration [3]. The calibration is preformed by extracting feature points and matching corresponding features between the cameras. In the case of self-calibration, the internal parameters are optimized simultaneously with the external ones. The method, however, assumes simplifications regarding the internal parameters, such as image center location is set to the center of image, distortion of the camera lens is omitted, etc. Cheng et al. [4] applied feature extraction from captured images of the environment to recover the location of the cameras. The method detects natural features in a scene observed by several cameras and tries to find matching features, i.e. same points seen by multiple cameras. The main drawbacks of this algorithm are variability of natural features in quantity and number and required proximity of cameras to allow detection of the same features in the scene. The latter restriction prevents calibration between two cameras facing opposite directions. Additionally, variability of photometric parameters of the cameras can significantly affect the accuracy and robustness of the feature detection algorithms.

Several researchers have have shown high accuracy for geometric calibration using one dimensional objects [5, 6, 7, 8]. Chen et al. [7] used iterative approach combined with extended Kalman filtering of object motion to calibrate unsynchronized cameras. Machacek et al. [6] suggested two-step calibration of a stereo camera system in a large volume where the internal parameters are first obtained with a calibration board, followed by the external calibration using a virtual calibration object. The study showed that small adjustments of lens focus, when cameras were deployed after the internal calibration, did not significantly affect the accuracy of this two-step calibration.

In our work we combine the idea of vision graphs for wide area camera network with small working volume overlap and calibration methods using virtual calibration object. Our algorithm requires cameras to share workspace volume at least pairwise. In contrast to other methods [7, 9] our approach resolves Euclidean reconstruction (preserving metric information) and introduces novel parameters reduction in the case of two-point bar calibration for multiple cameras as compared to [6]. Our main contribution is the application of weighted vision graph to determine the optimal transformation between the cameras when using pairwise calibration. The weight of the graph can comprise of number of common points between camera pairs, distribution of image points, closeness to the reference camera or a combination of several parameters.

3. PROPOSED METHOD

In our approach we use virtual calibration object defined by two moving markers with fixed distance. The algorithm requires for the cameras to at least pairwise share common volume. We assume that the cameras are synchronized. In case of unsynchronized cameras, the approach described by Chen et al. [7] can be applied. The calibration is two-step. First, the cameras are internally calibrated using checkerboard and well-known Tsai algorithm [10, 11]. In the second step, external calibration is performed to determine 6 external parameters describing orientation and position of each camera with regard to selected reference camera.

Our external calibration algorithm can be summarized as follows:

(a) image acquisition and sub-pixel marker detection on multiple cameras

(b) composition of adjacency matrix for vision graph describing interconnections between the cameras (e.g. number of common points)

(c) computation of fundamental **F** and essential matrix **E** with RANSAC

(d) essential matrix decomposition into rotation and translation parameters defined up to a scale factor λ

(e) determination of the scale factor λ through triangulation and LM optimization

(f) optimal path search using Dijkstra algorithm [12]

(g) global optimization of the parameters using sparse bundle adjustment [13]

In the remainder of this section we described in detail each of the calibration steps. The algorithms for intrinsic and extrinsic calibration were both implemented using C++ and OpenCV [14] computer vision library.

3.1. Camera Model and Intrinsic Calibration

In the first step of calibration, the cameras are internally calibrated using Tsai algorithm [10, 11]. A planar checkerboard target is placed in different positions and orientations to generate a set of points for homography calculation. Initial guess of the internal parameters (i.e. focal length, optical center and distortion) is optimized using Levenberg-Marquardt algorithm [13].

We use the standard pinhole camera model while considering radial and tangential distortion models [15]:

$$\mathbf{x}_i = \begin{bmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{T}_i \\ & \\ 0 & 1 \end{bmatrix} \mathbf{X}_i \tag{1}$$

$$\mathbf{x}_i = \mathbf{K}_f \mathbf{\Pi}_0 \mathbf{G} \mathbf{X}_i \tag{2}$$

The model in Eq. 2 represents the transformation from a homogeneous 3D point $\mathbf{X}_i \in \mathbb{R}^4$ seen by camera to the corresponding image pixel coordinate $\mathbf{x}_i$ defined on the image plane. Matrix $\mathbf{K}_f \in \mathbb{R}^{3\times3}$ represents camera matrix, consisting of the focal length (f_x, f_y), optical center (c_x, c_y) and skew angle parameter α. In most cases α can be set to 1. The matrix $\mathbf{\Pi}_0 \in \mathbb{R}^{3\times4}$ is the standard projection matrix. The matrix $\mathbf{G} \in \mathbb{R}^{4\times4}$ contains rotational matrix and position of the camera center from the object coordinate system origin. The lens distortion is modeled by two parameters of radial distortion (k_1, k_2) and two parameters of tangential distortion (p_1, p_2). In total, the camera model used in this paper consists of 8 internal parameters. All of the internal parameters have been estimates in our first step of calibration.

3.2. Marker Detection

The accuracy of marker detection can significantly influence calibration quality. Marker detection has to be reliable and robust. Detection can be influenced by CCD noise, uniformity of the background, other sources of illumination in the scene and poor thresholding of the captured images [16].

For the experiments on the real cameras, we have selected several cameras of our multi-camera system consisting of 48 cameras and 12 computers (clusters) [17]. The marker detection was implemented in real-time on each camera cluster. The camera parameters for shutter and gain were reduced to 5 ms and 3 dB respectively. LED markers were detected by thresholding the image and using ellipse fitting algorithm to eliminate any large or oddly shaped objects. To calculate sub-pixel marker center we used squared gray scale centroid method [16] where the sub-pixel marker center is determined by a centroid of the intensities of the detected marker. The input for the algorithm was the approximate center of the marker and the bounding box determined by the ellipse fitting algorithm. The sub-pixel marker center was calculated as follows:

$$\bar{\mathbf{x}} = \sum_{j=1}^{m} \sum_{i=1}^{n} i \cdot I_{i,j}^2 / \sum_{j=1}^{m} \sum_{i=1}^{n} I_{i,j}^2 \tag{3}$$

where $I_{i,j}$ denotes intensity value of the i,j-th pixel location; and m and n denote the dimensions of the bounding box around the marker. The marker detection algorithm allowed robust tracking in illuminated conditions.

3.3. Pairwise Calibration

Given two images from calibrated cameras, camera pose and position of the points in space can be obtained through epipolar geometry. First, the effects of the camera parameters are compensated by undistorting and normalizing the image points on all cameras. Essential matrix is then calculated from epipolar geometry constraints [15]. Through the essential matrix decomposition, pose of the camera can be obtained (up to a scale factor). Finally, the scale factor can be determined by the constraint between the two markers on the calibration bar.

3.3.1. Epipolar Geometry and Essential Matrix

The epipolar geometry is based on the fact that each 3D point $\mathbf{X}_i$ observed by two cameras and its two image projections $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ lie on the same plane [18]. The geometric relationship between the two cameras can be described by the fundamental matrix $\mathbf{F}$ with the following relation:

$$\mathbf{x}_{i2}^T \mathbf{F} \mathbf{x}_{i1} = 0 \tag{4}$$

The fundamental matrix depends on the internal parameters of the cameras ($\mathbf{K}_1$, $\mathbf{K}_2$) and the pose between the two cameras ($\mathbf{R}$, $\mathbf{T}$). For the calibration of a camera pair, the fundamental matrix in our algorithm is obtained using normalized 8-point algorithm implemented in OpenCV [14].

When describing the relationship between normalized image coordinates ($\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x}$), essential matrix ($\mathbf{E}$) can be obtained from the fundamental matrix:

$$\mathbf{F} = \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \quad \text{and} \quad \mathbf{E} = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1 \tag{5}$$

The following relationships between the image points can be defined for the essential matrix:

$$\hat{\mathbf{x}}_{i2}^T \mathbf{E} \hat{\mathbf{x}}_{i1} = 0 \tag{6}$$

In the subsequent text we omit symbol ' ˆ ' and assume that the image coordinates have been normalized, unless otherwise stated.

The essential matrix is defined as follows [18]:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \hat{\mathbf{T}} \mathbf{R} \tag{7}$$

where $\hat{\mathbf{T}}$ represents antisymmetric matrix of position vector $\mathbf{t}$ describing the relative position between the left and right camera coordinate system. Unlike the fundamental matrix with 7 degrees of freedom, the essential matrix has only five degrees of freedom. Important property of the essential matrix is that the singular value decomposition (SVD) results in two equal singular values and the third one is zero. This property is used for decomposition of essential matrix where four possible solutions for ($\mathbf{R}$, $\mathbf{t}$) are obtained [18]:

$$(\text{SVD})\mathbf{E} = \mathbf{U\Sigma V} \quad \text{with} \quad \mathbf{\Sigma} = diag\{\sigma, \sigma, 0\} \tag{8}$$

The four solutions obtained by the essential matrix decompositions are as follows:

$$\begin{array}{l} (\hat{\mathbf{T}}_1, \mathbf{R}_1) = (\mathbf{U}\mathbf{R}_Z(+\frac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \mathbf{U}\mathbf{R}_Z^T(+\frac{\pi}{2})\mathbf{V}^T) \\ (\hat{\mathbf{T}}_2, \mathbf{R}_2) = (\mathbf{U}\mathbf{R}_Z(-\frac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \mathbf{U}\mathbf{R}_Z^T(-\frac{\pi}{2})\mathbf{V}^T) \end{array} \tag{9}$$

where $\mathbf{R}_Z(.)$ represents 3 by 3 matrix defining the rotation around the z-axis for $\pm\frac{\pi}{2}$. The two solutions are referred to as "twisted pair" while their geometric interpretation results in the first two solutions are obtained by reversing the translation vector and the other two solutions are obtained by rotation through 180° about the line joining the two camera centers. In only one of the solutions a reconstructed point $\mathbf{X}_i$ will be in front of the both cameras (i.e. it has positive depth coordinate). Although testing with a single point should be sufficient, in practice, it turns out that due to noise testing with all points gives most reliable results in any position between the two cameras. Due to the nature of the essential matrix, the vector $\mathbf{t}$ can only be obtained up to a scale factor.

To optimize the results for $\mathbf{R}$ and $\mathbf{t}$, we apply LM algorithm for bundle adjustment [19] with three rotational and three position parameters as input. The following function is minimized [15]:

$$\Phi(\mathbf{R}, \mathbf{T}) = \sum_{j=1}^{N} \frac{(\tilde{\mathbf{x}}_2^{jT} \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_1^{jT})^2}{\|\hat{\mathbf{e}}_3 \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_i^{j}\|^2} + \frac{(\tilde{\mathbf{x}}_2^{jT} \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_1^{jT})^2}{\|\tilde{\mathbf{x}}_2^{jT} \hat{\mathbf{T}} \mathbf{R} \hat{\mathbf{e}}_3^{T}\|^2} \tag{10}$$

where $\hat{\mathbf{e}}_3$ is the anti-symmetric matrix of vector $\mathbf{e}_3 = [0, 0, 1]^T$. The expression (10) is based on the properties of the epipolar geometry: $\mathbf{x}_1^{jT}\mathbf{e}_3 = 1$, $\mathbf{x}_2^{jT}\mathbf{e}_3 = 1$ and $\mathbf{x}_2^T\mathbf{E}\mathbf{x}_1 = 0$.

From the essential matrix decomposition, the position vector $\mathbf{t}$ is obtained up to a scale factor λ. Next, we obtain the value of λ from known geometry of the LED markers [8].

3.3.2. Scale Factor Determination

The unknown scale factor λ is obtained from the dimension of the distance between the two LED markers. Pair of points $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ in the normalized 3D space can be reconstructed from their respective images using stereo triangulation while their coordinates in the absolute 3D space ($\mathbf{X}_1$ and $\mathbf{X}_2$) remain unknown. The scale factor λ can be determined from their distance in normalized space $\hat{d}$ and the actual length of the calibration bar d_0 as follows:

$$(\mathbf{X}_1 - \mathbf{X}_2) = \lambda(\hat{\mathbf{X}}_1 - \bar{\mathbf{X}}_2) \Rightarrow \lambda = \frac{d_0}{\hat{d}} \tag{11}$$

Due to presence of noise, the 3D reconstruction of the point pairs will not be precise. To improve the accuracy we calculate the mean value of the scale factor $\bar{\lambda}$ over N frames:

$$\bar{\lambda} = \frac{d_0}{N} \sum_{i=1}^{N} \frac{1}{d_i} \tag{12}$$

Finally, we implement non-linear optimization of the obtained solution using LM algorithm [13] where we minimize the error between calculated distance d and actual bar length d_0:

$$\delta(\lambda) = \sum_{i=1}^{N} d_0 - \|\mathbf{X}_{1i}(\lambda) - \mathbf{X}_{2i}(\lambda)\| \tag{13}$$

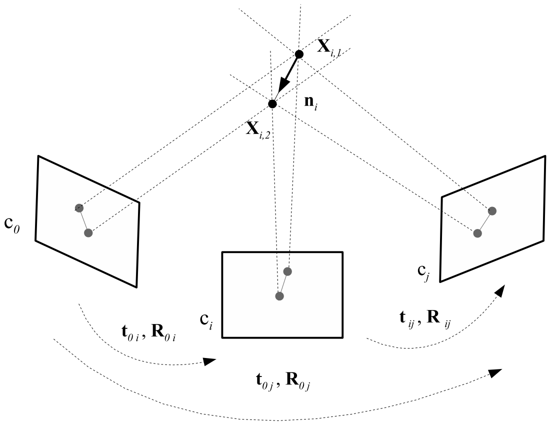


Fig. 1. Projection of i-th frame onto three image planes. The marker coordinates are parametrized using initial point $\mathbf{X}_{i1}$ and normalized direction $\mathbf{n}_i$.

3.4. Vision Graph

We represent the structure of the multi-camera system using tools of graph theory [12] and vision graphs [1]. The layout of M cameras is represented by graph G consisting of M vertices V_i which represent individual cameras. In order for the global calibration to succeed, the vision graph has to be connected. In terms of graph theory, this means that all pairs (i, j) of vertices are connected by paths (i.e. there is no isolated vertices inside the graph) [20]. We describe the overlap between different camera pairs by assigning weights to the graph edges. The weights ω_{ij} correspond to $\frac{1}{N_{ij}}$ where N_{ij} represents number of common points between the two cameras. If there are no common points between two cameras, value 0 is assigned to the weight. To describe the graph structure, adjacency matrix $\mathbf{A}(G)$ with weights ω_{ij} is defined. The weights ω_{ij} can be further modified to prioritize certain features of the cameras, such as closeness to the reference camera, accuracy of the internal calibration, or distribution of the captured image points. The adjacency matrix is updated after performing pairwise calibration and eliminating some of the points by RANSAC algorithm.

After the relative pose between all the camera pairs has been calculated, the location of any camera with regard to arbitrary selected reference camera can be computed as long as the graph remains connected. When calculating the transformations between the cameras, we try to find the optimal path to reduce the propagation of error. Two criteria should be considered for optimal transformation: (1) the calibration of a camera pair is more accurate with more common points between the cameras and (2) the number of transformations between different camera coordinate systems should be minimal. To find the optimal path for transformation from the reference camera to all the other cameras, we employ Dijkstra's shortest path algorithm [12] on the vision graph. The algorithm solves the single-source shortest path problem for a graph with non negative weights. The algorithm succeeds as long as the graph is connected.

Using the shortest path from the reference camera to each camera, we can calculate the absolute position of each camera (1). Let i, j, and k be indices of consecutive cameras on the path found in graph G. From pairwise calibration, the transformations from i to j and from j to k are denoted as ($\mathbf{R}_{ij}$, $\mathbf{t}_{ij}$) and ($\mathbf{R}_{jk}$, $\mathbf{t}_{jk}$). The transformation from i to k can be calculated as follows:

$$\mathbf{t}_{ik} = \mathbf{t}_{ij} + \mathbf{R}_{ij}\mathbf{t}_{jk} \quad \text{and} \quad \mathbf{R}_{ik} = \mathbf{R}_{ij}\mathbf{R}_{jk} \tag{14}$$

If a path from the reference camera has a length longer than two, the equation (14) is applied sequentially to cover the entire path.

3.5. Global Optimization

The solution described in the previous section was obtained using pairwise calculations of camera pose and is therefore prone to errors. The final goal of the calibration is to obtain the pose of each camera relative to the reference camera. The captured 3D points from the calibration bar can be seen as a 3D structure viewed by multiple cameras. Given 3D point coordinates in the reference camera frame and the initial pose of the cameras, one can optimize the reprojection error using bundle adjustment (BA) algorithm which simultaneously refines the 3D structure and the camera parameters. The algorithm for each 3D point calculates reprojection error to all camera images and adjusts parameters to minimize the error between the reprojected and captured image point. The optimization can be effectively solved using Levenberg-Marquardt (LM) nonlinear optimization. Due to sparse nature of the problem, where there is lack of interaction between different 3D points and cameras, sparse bundle adjustment (SBA) can be applied [19].

The SBA algorithm assumes we have n 3D points which are seen by m cameras. Projection of i-th point on camera plane j is denoted as $\mathbf{x}_{ij}$. Each camera can be parametrized by vector $\mathbf{a}_j$ and each 3D point i by vector $\mathbf{b}_i$. Function $\mathbf{Q}()$ defines projection of the 3D point onto camera image plane using the camera model from Eq. 2. Function $d(\mathbf{x}, \mathbf{y})$ denotes Euclidean distance between image points represented by $\mathbf{x}$ and $\mathbf{y}$. Bundle adjustment minimizes the following reprojection error:

$$\min_{\mathbf{a}_j, \mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} d(\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i), \mathbf{x}_{ij})^2 \tag{15}$$

The non-linear minimization problem is defined by the parameter vector $\mathbf{P} \in \mathbb{R}^M$, consisting of all camera pose parameters, and the measurement vector $\mathbf{X} \in \mathbb{R}^N$, consisting of the measured image points across all cameras.

To obtain the initial position of each 3D point in the reference camera coordinate system, we use pair-wise stereo triangulation to obtain the position of points visible to a particular camera pair. Next, we calculate average position of the 3D points over the pairs that capture the point since the projections from different camera may not coincide into the same point. To remove outliers, we check the distance between the two 3D points of the calibration bar. The threshold for distance error was set at 1%

For n freely distributed 3D points in the scene and m cameras, the dimensions of the parameter space are $M = n \times 3 + 6 \times (m-1)$ (3 coordinates for each 3D point, and 3 rotational and 3 translational parameters) and the dimensions of the measurement space are $N \leq n \times 2$ since all cameras may not see each 3D point. For example, 500 3D points observed by 5 cameras will be defined by 1524 parameters while the image projection space will be ≤ 5000.

The number of the input parameters can be reduced if we take into account the rigid connection between each 3D point pair on the calibration bar. The position of the two 3D points can be described by the starting point $\mathbf{X}_{i,1}$ while the location of the second point $\mathbf{X}_{i,2}$ is defined by the normalized direction vector $\mathbf{n}_i$ between the two points and their distance d_0 which is a priori known (Figure 1). The normalized direction vector can be parameterized as follows:

$$\mathbf{n}_i = \frac{\mathbf{X}_{i,2} - \mathbf{X}_{i,1}}{\|\mathbf{X}_{i,2} - \mathbf{X}_{i,1}\|} = \begin{bmatrix} n_{ix} \\ \sqrt{1 - n_{ix}^2 - n_{iy}^2} \\ n_{iz}^2 \end{bmatrix}. \tag{16}$$

The coordinate n_{iy} is expressed by the other two coordinates since the calibration bar is kept close to vertical direction and its value will therefore be close to 1 and n_{ix} and n_{iz} will be balanced numbers. Inside the LM loop we enforce the condition $n_{ix}^2 + n_{iz}^2 \leq 1$ to keep the direction vector normalized. Finally, we can calculate the second coordinate of the point bar as follows:

$$\mathbf{X}_{i,2} = \mathbf{X}_{i,1} + d_0\mathbf{n}_i \tag{17}$$

Above parametrization will decrease the parameter space size to $M = \frac{n}{2} \times 5 + 6 \times (m-1)$. In case of the numerical example given above, the number of parameters would decrease from 1524 to 1274. The dimensions of the measurement vector remains the same. The equation (15) is applied for reprojection minimization. The parameterization additionally constraints the LM optimization to keep the distance between the two 3D points constant. In subsequent text we refer to the use of above parametrization as the constrained SBA.

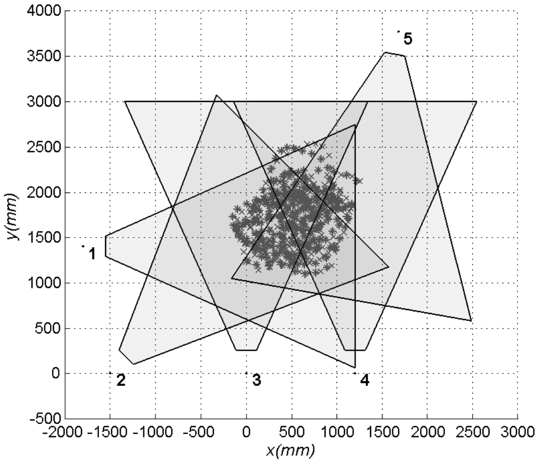


Fig. 2. Simulated setup of five cameras and generated 3D points used for the calibration.

4. RESULTS

4.1. Simulated Data

The performance of the algorithm was analyzed on simulated setup of five cameras. The cameras were arranged as shown in Figure 2. For clarity and due to limitation of space in this paper, all the cameras were positioned in the same plane. The internal parameters of the cameras were randomly chosen from five actual cameras of our multi-camera setup which were calibrated using checkerboard. The internal parameters included four parameters of camera matrix and four parameters for radial and tangential distortion.

For the experiment we have generated 310 positions of the calibration bar with two markers positioned at the distance of 314 mm from one another. Figure 3 shows the vision graph generated from the calibration with camera #3 chosen as the reference camera. Due to a small overlap between camera #3 and camera #4, the optimal transformation path for this camera was 3-2-4. Too few points were also found between cameras #3 and #5, therefore alternative path 3-2-5 was found by the path-searching algorithm.

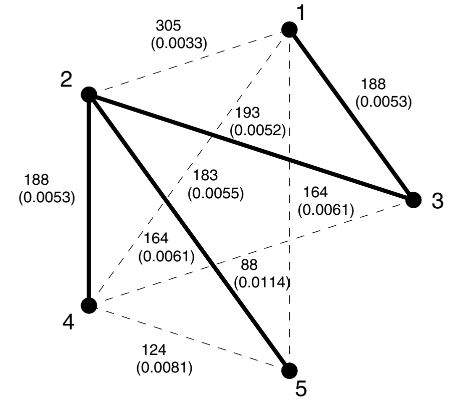
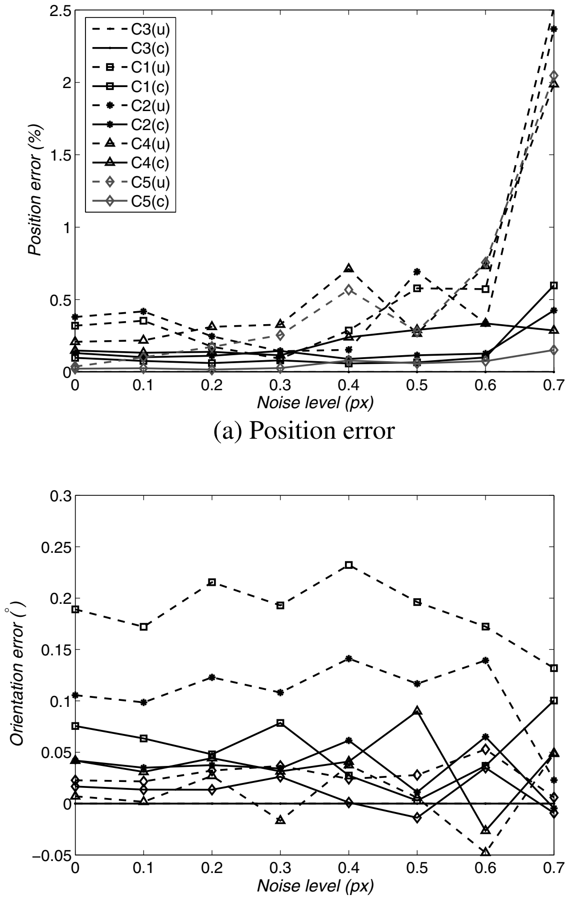


Fig. 3. Vision graph and the optimal path from reference camera #3 obtained from the above setup. Numbers indicate number of common points between cameras and their corresponding weights.

Next, Gaussian noise with mean value 0 and standard deviation σ was added to the image points. The noise level on image size 640 × 480 was varied from 0 to 0.7 pixels in steps of 0.1 pixels. At each noise level, cameras were calibrated under two conditions, with unconstrained and constrained SBA. We analyzed the effect of noise on position and orientation of the cameras obtained from the proposed calibration algorithm.

The results of the noise analysis in Figure 4(a) show the relative error in calibrated position of the cameras. The error was calculated as percentage of the root mean square error between calibrated camera position and its true position in 3D space. Parameterization of the two markers with 3D point and normalized direction improved the robustness and accuracy of the algorithm. The errors are below 0.2% for noise levels of 0.6 pixels and under. With increasing noise levels, the RANSAC algorithm, whose threshold was set at 1 pixel, removes too many points to successfully perform the calibration. Figure 4(b) shows the absolute error in the orientation with regard to the vertical axis of the reference camera for different noise levels. The difference between unconstrained and constrained SBA algorithm is evident mainly for cameras #1 and #2, which are oriented at 45 and 90 degrees, respectively, with regard to the reference camera. The corresponding relative error for these two cameras is between 0.05 and 0.1%. The corresponding image reprojection errors for different levels of noise are listed in Table 1.



(a) Position error

(b) Orientation error

Fig. 4. Camera position and orientation errors for different noise levels as compared between unconstrained (u) and constrained (c) sparse bundle adjustment.

4.2. Real Data

The experiment with the real data was performed on 12 Dragonfly firewire cameras with the image resolution of 640 × 480 pixels. Two of the cameras (#7 and #11) had 4 mm lens installed while the rest of the cameras had 6 mm lenses. The size of the setup was about 4.0 m × 4.0 m × 2.5 m, with the cameras positioned at different vertical levels and various orientations. Due to small or no overlap between some of the cameras, it would be difficult to deploy a physical calibration object to calibrate all the cameras. The cameras were synchronized using external trigger to capture images with 15 frames per second. The cameras were first internally calibrated using the checkerboard with 10 × 15 number of squares with 40 mm in size. The checkerboard was placed in 20 different positions and orientations. Tsai calibration algorithm was used to obtain the intrinsic parameters for each camera.

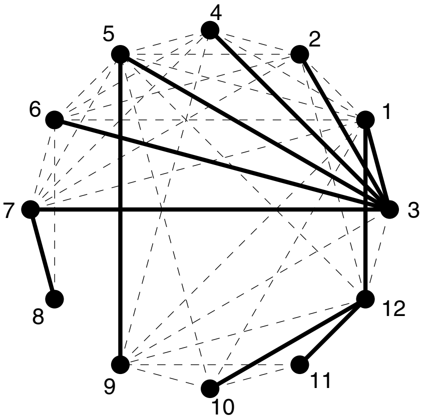


Fig. 5. Vision graph obtained for 12 real cameras with the optimal transformation path shown with thick line.

For external calibration we used a rigid metal bar with two LED markers attached on each end. The distance between the markers was measured at 317 mm using a tape measure before the calibration. The LEDs were emitting in visible and infra-red spectrum. The marker locations were extracted in real time and stored locally. Data analyzed in this paper consisted of 4738 collected 3D points. As we have demonstrated on the synthetic camera setup, the calibration can be accurately and robustly performed with smaller number of points. Implementation using C++ and OpenCV library and the use of sparse bundle adjustment makes our calibration algorithm fast and efficient. For the data set presented in this paper, the complete external calibration of 12 cameras, took 14 seconds on a personal computer with Intel Xeon 3.20 GHz processor and 1 GB of memory.

Figure 5 shows the vision graph and the corresponding optimal transformation path as obtained from the collected data points. The vision graph weights were calculated based on the number of overlapping points between the camera pairs. Camera #3 was chosen as the reference camera since its position and orientation correspond with the floor level. The

Noise level σ	Image reprojection error
0.0	0.0417
0.1	0.1150
0.2	0.2083
0.3	0.2968
0.4	0.3937
0.5	0.4958
0.6	0.6172
0.7	0.6750

Table 1. Mean image reprojection error in pixels across five cameras for different noise levels σ.

results of the external calibration are shown in Figure 6.

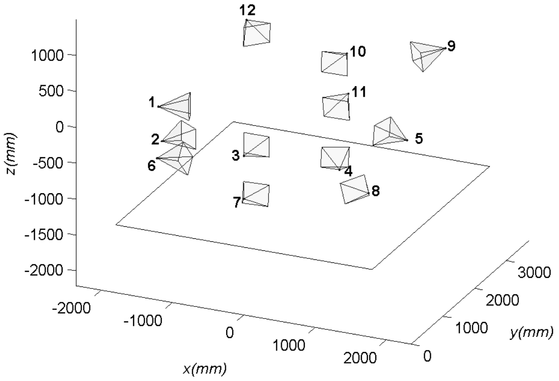


Fig. 6. A three-dimensional layout of 12 real cameras after the calibration.

To assess the accuracy of the calibration, we used mean reprojection error obtained on each camera (Figure 7). The reprojection error on all cameras was below 0.4 pixels. The cameras whose position and orientation were obtained by indirect transformation path with the reference camera had no significantly different reprojection errors as compared to the cameras calibrated directly with the reference camera. The mean reprojection error between all the cameras was 0.3391 pixels with the standard deviation of 0.0365 pixels. Compared to the results obtained for the synthetic camera setup, this reprojection error would correspond to the image noise of about $\sigma = 0.4$ pixels.

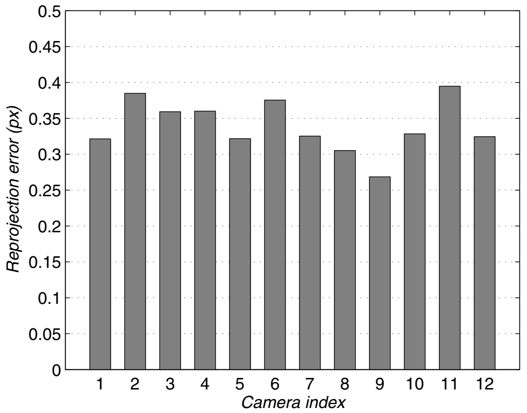


Fig. 7. Reprojection error in pixels on each camera plane as obtained after global optimization. The reprojection errors between the cameras show no significant difference between the cameras calibrated by indirect and direct transformation path.

The accuracy of the external calibration depends on several factors: (a) accuracy of internal camera calibration, (b) accuracy of marker detection algorithm, (c) number of common points and their distribution on image plane, and (d) distance between the two LED markers. For metric accuracy, the measured length of the calibration bar is critical. The length of the calibration bar should be as large as possible for the length to be reconstructed with adequate accuracy, however, the projection of the marker points should cover most of the image plane (i.e. center and edge regions) to obtain reliable fundamental matrix. The length of the bar was therefore chosen in such a way that its projected length represented about 1/3 of the image height.

5. CONCLUSIONS

In this paper we have presented a robust and efficient camera calibration method which can be applied to distributed smart cameras. The calibration is based on virtual calibration object created by LED markers. The marker detection can be efficiently implemented on smart cameras since it requires minimal image processing. The marker location data can be stored on-board the cameras or sent through the network to a server. The method does not require the scene to be dark. Color information can be included in the marker tracking to increase the robustness in case of environment with different light sources.

The proposed vision graph analysis allows calibration of camera setups in which all the cameras do not share common working volume. The only requirement is for the cameras to have pairwise overlap. We apply weights to graph edges to describe relationship between camera pairs and find optimal path transformation to minimize the error. In our future work we plan to explore how different parameters (e.g. distribution of points, closeness to reference camera) used for weight calculation affect accuracy of the calibration and error propagation. The major advantage of using the vision graph is that the algorithm does not need any prior knowledge of approximate camera locations, allowing for fast and robust calibration of distributed camera systems. Our algorithm, in contrast to [3], also reconstructs metric information on camera positions.

Our novel parameterization of the two-marker approach adds robustness to the algorithm allowing more accurate camera calibration with two-point target in presence of noise as we have demonstrated in our experiment with the synthetic data. The results obtained on the real camera setup show that our approach compensates for error propagation when the path transformation includes two to three nodes. No significant difference in reprojection error was found between the two groups of cameras.

Vision graph based approach described in this paper could also be applied to address video segmentation in structure from motion problems [21] or mobile robot localization [22] where robot position can be determined from two pairs of views of the same scenery.

6. REFERENCES

[1] W.E. Mantzel, H. Choi, and R.G. Baraniuk, "Distributed camera network localization," *38th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1381–1386 Vol.2, 7-10 Nov. 2004.

[2] B.D. Olsen and A. Hoover, "Calibrating camera network using domino grid," *Pattern Recognition*, vol. 34, pp. 1105–1117, 2001.

[3] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," *Presence*, vol. 14, no. 4, pp. 407–422, 2005.

[4] Z. Cheng, D. Devarajan, and R.J. Radke, "Determining vision graphs for distributed camera networks using feature digests," *EURASIP Journal on Advances in Signal Processing: Special Issue on Visual Sensor Networks*, p. 11, 2007.

[5] Z. Zhang, "Camera calibration with one-dimensional objects," Tech. Rep. MSR-TR-2001-120, Microsoft Research, August 2002.

[6] M. Machacek, M. Sauter, and T. Rsgen, "Two-step calibration of a stereo camera system for measurement in large volumes," *Measurement Science and Technology*, vol. 14, pp. 1631–1639, 2003.

[7] X. Cheng, J. Davis, and P. Slusallek, "Wide area camera calibration using virtual calibration objects," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, 2000.

[8] N.A. Borghese and P. Cerveri, "Calibrating a video camera pair with a rigid bar," *Pattern Recognition*, vol. 33, no. 1, pp. 81–95, 2000.

[9] I. Ihrke, L. Ahrenberg, and Marcus M. Magnor, "External camera calibration for synchronized multi-video systems," in *Proceedings of 12th International Conference on Computer Graphics, Visualization and Computer Vision 2004*, Plzen, Czech Republic, February 2004, vol. 12, pp. 537–544.

[10] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA3, no. 4, pp. 323–344, 1987.

[11] D. Zhang, Y. Nomura, and S. Fujii, "Error analysis and optimization of camera calibration," in *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS 91), Osaka, Japan*, 1991, pp. 292–296.

[12] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Elsevier Science Publishing Co., Inc., New York, 5th edition, 1982.

[13] M.I.A. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] http://www.ics.forth.gr/ lourakis/levmar, July 2004.

[14] "Opencv: Open computer vision library," [web page] http://sourceforge.net/projects/opencvlibrary, November 2006.

[15] Y. Ma, S. Soatto, J Košecká, and S.S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, vol. 26, Springer-Verlag, New York, NY, 2004.

[16] M.R. Shortis, T.A. Clarke, and T. Short, "A comparison of some techniques for the subpixel location of discrete target images," in *Proceedings of SPIE Conference*, Sabry F. El-Hakim, Ed., 1994, vol. 2350, pp. 239–250.

[17] S.H. Jung and R. Bajcsy, "A framework for constructing real-time immersive environments for training physical activities," *Journal of Multimedia*, vol. 1, no. 7, pp. 9–17, 2006.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, New York, NY, second edition, 2004.

[19] M.I.A. Lourakis and A.A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004, [web page] http://www.ics.forth.gr/ lourakis/sba.

[20] A. K. Hartman and M. Weigt, *Phase transition in combinatorial optimization problems: Basics, Algorithms and Statistical Mechanics*, Wiley-VCH Verlag, 2006.

[21] M. Pollefeys, L. Van Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops, "Image-based 3d acquisition of archaeological heritage and applications," in *Proceedings of Virtual Reality, Archaeology, and Cultural Heritage Conference (VAST 2001)*, 2001, pp. 255–262.

[22] Li Maohai, Hong Bingrong, and Luo Ronghua, "Novel method for monocular vision based mobile robot localization," in *Proceedings of International Conference on Computational Intelligence and Security*, PGuangzhou, November 2006, vol. 12, pp. 949–954.