Framework for Hierarchical Calibration of Multi-camera Systems for Teleimmersion

Gregorij Kurillo
University of California, Berkeley
Hearst Mining Bldg. #475
Berkeley, CA 94720-1764
tel: +1 510 642 7456
gregorij@eecs.berkeley.edu

Zeyu Li
University of California, Berkeley
Hearst Mining Bldg. #475
Berkeley, CA 94720-1764
tel: +1 510 642 7512
zeyuli@eecs.berkeley.edu

Ruzena Bajcsy
University of California, Berkeley
Hearst Mining Bldg. #475
Berkeley, CA 94720-1764
tel: +1 510 642 9423
bajcsy@eecs.berkeley.edu

ABSTRACT

In this paper we present framework for hierarchical calibration of multi-camera based teleimmersion systems. First, we calibrate the internal camera parameters and geometry of each stereo cluster using a checkerboard. Next, we present a robust and efficient method to externally calibrate the location of the stereo clusters using virtual calibration object created by two LED markers. Our novel algorithm does not require for all the cameras to share common workspace; only pairwise overlap is required. Finally, we address geometric correspondence between several remote locations by proposing a simple calibration method.

Keywords

teleimmersion, external camera calibration, multi-camera system

1. INTRODUCTION

For realistic experience of teleimmersion technology, the metrics and geometry of the interaction space among remote users have to be properly established. For example, a local participant should recognize the metrics and perceive spatial relationship to other remote users in a similar fashion as in real-life experience. In case of multi-camera based teleimmersion systems accurate and robust calibration of the cameras with respect to their intrinsic and extrinsic parameters is of high importance. Once the cameras are calibrated, the correspondence between the remote locations has to be defined.

In this paper we address both, multiple camera calibration and correspondence between remote locations. Our focus are systems for teleimmersion where multiple cameras are employed, either to reconstruct the depth using different stereo algorithms [3, 7] or to deliver different points of view to the users [13]. In systems with multiple cameras, two or more cameras may be arranged into (stereo) clusters which are used to perform the 3D reconstruction. Other systems may treat the multiple cameras as independent of each other until different algorithms are employed to reconstruct the full 3D scene. Our algorithm can be applied to both type of systems. We approach the calibration problem in a hierarchical fashion, starting with a single camera as the smallest unit. In the first step each stereo cluster (or camera) is independently calibrated for the internal and external parameters to define the geometry of the cluster. The intrinsic calibration of the cameras is performed by the well-known Tsai algorithm [16] using a checkerboard. Simultaneously, position and orientation of the cameras within the cluster are obtained.

Our external calibration approach [6] assumes that at least any two given clusters (cameras) overlap. The cameras are externally calibrated using a virtual calibration object generated by two LED markers. Global calibration is solved by constructing a 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.

In the last part of the paper, we address calibration across remote locations. The proposed camera calibration method on the local level, preserves the metric information on the scene captured by the cameras. The remote calibration on the other hand defines the relationship between the camera space and the physical space.

2. RELATED WORK

Camera calibration has been studied extensively in the past decades. Several methods exist for internal camera calibration [16, 17]. Due to length constraints of this paper, we limit our review to selected methods for external calibration of multiple cameras. Mantzel and colleagues [12] calibrated a set of cameras with sparse overlapping by acquiring planar checkerboard images. Similar approach was applied by Olsen and Hoover [14] who calibrated multiple cameras with small overlap in workspace using planar domino grid. In several studies, calibration targets have been substituted by the self-calibration approach [15]. 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.

Several researchers have shown high accuracy for geometric calibration using one dimensional objects [17, 11, 2]. Chen et al. [2] used iterative approach combined with ex-

IMMERSCOM 2009, May 27-29, Berkeley, USA

DOI 10.4108/immerscom.2009.11

tended Kalman filtering of object motion to calibrate unsynchronized cameras. Machacek et al. [11] proposed two-step calibration where cameras are fist calibrated internally, followed by the external calibration using a virtual calibration object.

Our algorithm combines the idea of graph theory to describe multi-camera topology and use of a virtual calibration object. In contrast to other methods [2, 5, 15] 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 [11]. Our main contribution is the application of weighted vision graph to determine the optimal transformation between the cameras. With our hierarchical approach, we are able to accurately calibrate large number of cameras in a short time. In the paper we present results of the calibration on 48-camera system for teleimmersion [7].

3. SYSTEM OVERVIEW

Our teleimmersion apparatus consists of 48 Dragonfly cameras (Point Grey Research Inc, Vancouver, Canada) with the resolution of 640 × 480 pixels which are arranged in 12 stereo clusters covering 360° view of the user(s). Two of the clusters (#7 and #11) have cameras with 4mm lenses while the remaining clusters use 6mm lenses. Each cluster consists of three grayscale cameras intended for stereo reconstruction and a color camera for texture acquisition. Real-time stereo reconstruction algorithm is performed on each camera cluster simultaneously to obtain partial images of the 3D scene. 3D mesh obtained from this algorithm is sent through the network and displayed in the virtual environment to image local and remote users. Interested reader is referred to [7] for more information on the hardware setup and the 3D reconstruction algorithms.

4. CAMERA CLUSTER CALIBRATION

In the first step of the calibration, all cameras are internally calibrated using well-known Tsai algorithm [16]. 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 is optimized using LM algorithm [8]. Simultaneously we obtain the geometric relationship of the cameras within the cluster.

4.1 Camera Model

We use the standard pinhole camera model while considering radial and tangential distortion models [10]:

$$\mathbf{x}_i = \mathbf{K}_f \mathbf{\Pi}_0 \mathbf{G} \mathbf{X}_i \tag{1}$$

The model in equation 1 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 are estimated in the first step of calibration.

4.2 Intrinsic Calibration

The algorithm for camera calibration uses a set of known points $\mathbf{X}_i$ (i = 1, 2,... N) defined by the corner features of the checkerboard. The checkerboard is placed in different positions and orientations while the camera captures the projection $\mathbf{x}_i$ of the points onto the image plane. Linear solution of the camera parameters can be obtained by writing a set of equations from the camera model equation 1 for the projection of known grid coordinates $\mathbf{X}_i$ onto detected image points $\mathbf{x}_i$. The set of linear equations can be solved by singular value decomposition (SVD) to obtain the linear solution of the 8 internal parameters and 6 external parameters for each frame. LM algorithm is applied to further refine the parameters and reduce the reprojection error. The error function is defined as the reprojection error between M image points $\mathbf{x}_i$ obtained in P positions of the calibration board and the points projected through the camera model with the linear parameters as the initial guess:

$$e_i = \sum_{p=1}^{P} \sum_{j=1}^{M} F(\mathbf{K}_i, \mathbf{R}_i^p, \mathbf{t}_i^p) \tag{2}$$

After each camera is calibrated independently, the relative orientation and position of the cameras within the cluster (R_{i0}, t_{i0}) is obtained. The relative relationship between an arbitrary camera C_i and selected reference camera C_0 can be expressed as follows:

$$\mathbf{R}_{i0} = \mathbf{R}_i \mathbf{R}_0^{-1}, \quad \mathbf{t}_{i0} = \mathbf{t}_i - \mathbf{R}_i \mathbf{R}_0^{-1} \mathbf{t}_0 \tag{3}$$

In equation 3 the parameters ($\mathbf{R}_0$, $\mathbf{t}_0$) denote the orientation and position of the reference camera and ($\mathbf{R}_i$, $\mathbf{t}_i$) denote the orientation and position of the arbitrary camera with regard to the current checkerboard position p. Due to noise, ($\mathbf{R}_{i0}$, $\mathbf{t}_{i0}$) parameters will slightly vary for different checkerboard positions. Average value of the parameters is used in further computations.

To replace ($\mathbf{R}_i$, $\mathbf{t}_i$) in Eq. 1, the transformation in Eq. 3 is rearranged:

$$\mathbf{R}_i = \mathbf{R}_{i0} \mathbf{R}_0, \quad \mathbf{t}_i = \mathbf{t}_{i0} + \mathbf{R}_{i0} \mathbf{t}_0 \tag{4}$$

Next, we rewrite equation 1 to only consider the orientation and position of the reference camera and the relative orientation and position between cameras:

$$x_i = \mathbf{K}_{fi} \mathbf{\Pi}_0 \begin{bmatrix} \mathbf{R}_{i0}\mathbf{R}_0 & \mathbf{t}_{i0} + \mathbf{R}_{i0}\mathbf{t}_0 \\ 0 & 1 \end{bmatrix} \mathbf{X}_0 \tag{5}$$

Finally, non-linear optimization of the external camera parameters within the cluster is performed using LM algorithm. The error function is defined as the total reprojection error defined as the sum of reprojection errors of all the grid points M as seen by N cameras in P checkerboard positions:

$$e_{total} = \sum_{p=1}^{P} \sum_{i=1}^{N} \sum_{j=1}^{M} F(\mathbf{K}_i, \mathbf{R}_{i0}, \mathbf{t}_{i0}, \mathbf{R}_0^p, \mathbf{t}_0^p) \tag{6}$$

Since the internal parameters of the cameras are independent and have already been optimized, only external

Digital Object Identifier: 10.4108/ICST.IMMERSCOM2009.6119
http://dx.doi.org/10.4108/ICST.IMMERSCOM2009.6119

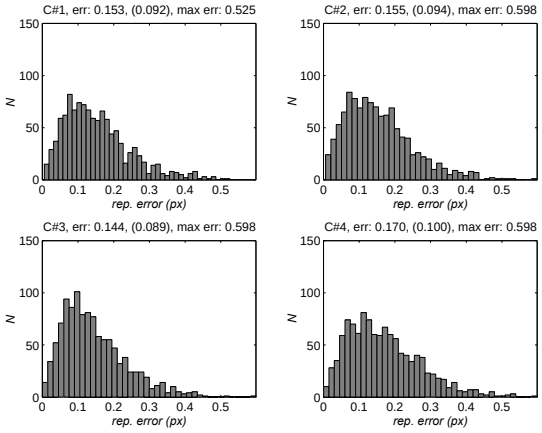

Figure 1: Distribution of the reprojection error for four cameras after cluster calibration with 10 checkerboard images. The total mean reprojection error was 0.153 (0.091) pixels.

parameters are considered in this optimization. In total $6 \times (N-1) + 6 \times P$ parameters are optimized.

4.3 Results

For the calibration of the stereo clusters with 4 cameras, we used a black and white checkerboard with 15 x 10 squares (square size was 40 mm). Number of points on each grid was 126. About 15-20 images were collected on each cluster with the checkerboard placed in different orientations and positions. The calibration software was written in C++ using OpenCV [1] and levmar LM algorithm [8] libraries. Numerical approximation of the Jacobian matrix was used for the optimization process.

Figure 1 shows typical error distribution as obtained on the four cameras within the cluster. After the global optimization on all four cameras, the combined error distribution resembles Gaussian distribution with the mean value of 0.153 pixels and standard deviation of 0.091 pixels. Note that the reprojection error on the color camera (#4) is higher than on the grayscale cameras. The maximal error for this set of images was 0.8 though only a small number of points had errors in that range.

5. EXTERNAL CALIBRATION OF MULTIPLE STEREO CLUSTERS

Once all the cameras have been internally calibrated and the geometry of the stereo clusters has been defined, we need to determine the location of the clusters with regard to the selected reference cluster. For the external calibration we use virtual calibration object defined by two moving LED markers located at a fixed distance. Our algorithm requires for the cameras to at least pairwise share common volume. The 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 weighted vision graph describing interconnections between the cameras (e.g. number of common points)
(c) computation of fundamental $\mathbf{F}$ and essential matrix $\mathbf{E}$ with RANSAC
(d) essential matrix decomposition into rotation and translation parameters defined up to scale factor λ
(e) determination of the scale factor λ through triangulation and LM optimization
(f) optimal path search using Dijkstra algorithm
(g) global optimization of the parameters using sparse bundle adjustment (SBA) [8]

In the remainder of this section we described each of the calibration steps and present the results obtained on the real teleimmersion setup. Additional details on the calibration algorithm and further error analysis can be found in [6].

5.1 Marker Detection

Marker detection has to be reliable and robust in different environmental conditions. In our algorithm LED markers were detected in real time by thresholding captured image with low shutter setting. Ellipse fitting algorithm was applied to eliminate any large or oddly shaped objects. To calculate sub-pixel marker center we used squared gray scale centroid method where the marker center is determined by a centroid of the intensities of the detected marker.

5.2 Pairwise Calibration

Given two images from calibrated cameras, camera pose and the position of the points in space can be obtained through epipolar geometry and essential matrix decompositions. 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 [4]. The geometric relationship between the two cameras can be described by the fundamental matrix $\mathbf{F}$ for the image coordinates $\mathbf{x}$ and by the essential matrix $\mathbf{E}$ for the normalized image coordinates ($\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x}$):

$$\mathbf{x}_{i2}^T \mathbf{F} \mathbf{x}_{i1} = 0 \quad and \quad \hat{\mathbf{x}}_{i2}^T \mathbf{E} \hat{\mathbf{x}}_{i1} = 0 \tag{7}$$

The relationship described in equation 7 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 in our algorithm, the fundamental matrix is obtained using normalized 8-point algorithm implemented in OpenCV [1]. The essential matrix is defined as follows:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \hat{\mathbf{T}}\mathbf{R} \tag{8}$$

where $\hat{\mathbf{T}}$ represents antisymmetric matrix of position vector $\mathbf{t}$ describing the relative position between the left and right camera coordinate system. Using singular value decomposition (SVD), the matrices $\mathbf{T}$ and $\mathbf{R}$ can be obtained by examining the four solutions for the one that yields the positive depth reconstructed for all the image points [4]. Results obtained from essential matrix decomposition are further optimized using LM algorithm for bundle adjustment to comply with the properties of the essential matrix [6].

Due to the nature of the essential matrix, the vector $\mathbf{t}$ can only be obtained up to a scale factor λ. However, λ can be obtained from 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

Digital Object Identifier: 10.4108/ICST.IMMERSCOM2009.6119
http://dx.doi.org/10.4108/ICST.IMMERSCOM2009.6119

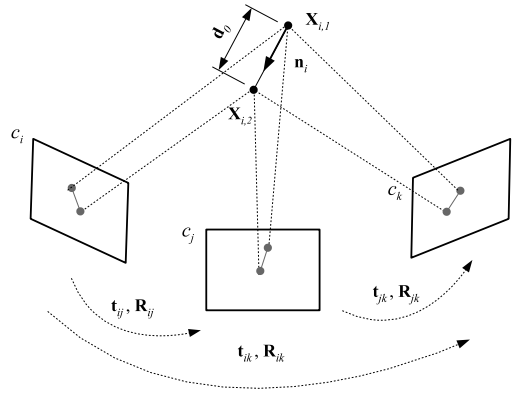

Figure 2: Projection of i-th frame onto three image planes. The marker coordinates are parametrized using initial point $\mathbf{X}_{i1}$ and normalized direction $\mathbf{n}_i$.

determined from their distance in normalized space $\hat{d}$ and the actual length of the calibration bar d_0 as follows:

$$(\mathbf{X}_1 - \mathbf{X}_2) = \lambda(\hat{\mathbf{X}}_1 - \hat{\mathbf{X}}_2) \Rightarrow \lambda = \frac{d_0}{\hat{d}} \tag{9}$$

Accuracy of the obtained scale factor is improved by calculating the average scale factor $\bar{\lambda}$ over N frames and further non-linear optimization using LM algorithm [8] to minimize the error between calculated distance d and actual bar length d_0.

5.3 Vision Graphs

We represent the structure of the multi-camera system using tools of graph theory. The layout of M cameras is represented by graph G consisting of M vertices V_i which represent individual cameras (clusters). In order for the global calibration to succeed, the vision graph has to be connected. We describe the overlap between different camera pairs by assigning weights ω_{ij} to the graph edges as $\frac{1}{N_{ij}}$, where N_{ij} represents number of common points seen by the two cameras. If there are no common points between two cameras, value 0 is assigned to the weight.

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. The optimal transformation path is determined using Dijkstra's shortest path algorithm which solves the single-source shortest path problem for a graph with non negative weights.

Using the shortest path from the reference camera to each camera, we can calculate the absolute position of each camera (Figure 2). 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:

$$\mathbf{t}_{ik} = \mathbf{t}_{ij} + \mathbf{R}_{ij}\mathbf{t}_{jk} \quad \text{and} \quad \mathbf{R}_{ik} = \mathbf{R}_{ij}\mathbf{R}_{jk} \tag{10}$$

5.4 Global Optimization

Once we obtained the initial solution of the relative position and orientation of the cameras, the results are globally optimized using nonlinear optimization to reduce the errors. Sparse nature of the optimization problem (i.e. all cameras cannot see all the points), the solution can be obtained using sparse bundle adjustment (SBA) [9]. The algorithm simultaneously refines the 3D structure and the external camera parameters. The internal camera parameters are kept fixed since they were already optimized during the cluster calibration.

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. 1. 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:

$$\min_{\mathbf{a}_j, \mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} d(\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i), \mathbf{x}_{ij})^2 \tag{11}$$

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. Initial position of 3D points in the coordinate system of the reference camera needed for the bundle adjustment are obtained using pair-wise stereo triangulation on each camera pair where points are visible. Average position is calculated if more than one pair sees the point.

We further reduce the number of parameters by parameterizing the coordinates of the two LED markers using the starting point $\mathbf{X}_{i,1}$, the normalized direction vector $\mathbf{n}_i$ between the two points, and their distance d_0 which is a priori known (Figure 2). The normalized direction vector n_{iy} component is expressed by the remaining 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.

$$\mathbf{n}_i = \frac{\mathbf{X}_{i,2} - \mathbf{X}_{i,1}}{\|\mathbf{X}_{i,2} - \mathbf{X}_{i,1}\|} = \begin{bmatrix} n_{ix} \\ \sqrt{1 - n_{ix}^2 - n_{iy}^2} \\ n_{iz}^2 \end{bmatrix}. \tag{12}$$

The parameter space is thus reduced from $M = n \times 3 + 6 \times (m-1)$ to $M = \frac{n}{2} \times 5 + 6 \times (m-1)$ where n is number of 3D points in the scene and m is the number of cameras. For example of 500 3D points observed by 5 cameras, the number of parameters is reduced from 1524 to 1274.

5.5 Results

For external calibration we used a rigid metal bar with two LEDs attached on each end. We chose Luxeon I LED (Phillips Lumileds Lighting Company, San Jose, CA), with brightness 30.6 lm and 160° emitting angle. The distance between the markers was measured at 317 mm using a tape measure prior to calibration. Data analyzed in this paper consisted of 2534 collected 3D points which were collected in real time with frequency of 15 Hz on each cluster and stored locally. The calibration algorithm was implemented using C++ and OpenCV computer vision library to allow fully automatic and fast calibration. The complete external calibration of 12 cameras took 6 seconds on a personal

Digital Object Identifier: 10.4108/ICST.IMMERSCOM2009.6119
http://dx.doi.org/10.4108/ICST.IMMERSCOM2009.6119

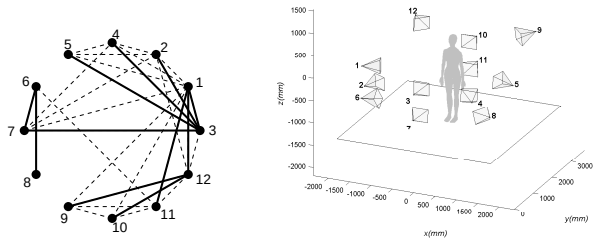

Figure 3: Vision graph generated for 12 stereo clusters with the cluster #3 selected as the reference cluster.

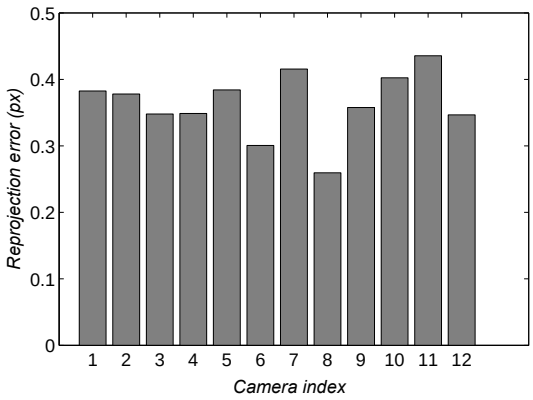

Figure 4: Reprojection error in pixels on each reference camera plane after global bundle adjustment. The reprojection error shows no significant difference between the cameras calibrated by the indirect and direct transformation paths.

computer with Intel Xeon 3.60 GHz processor and 2 GB of memory.

Figure 3 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 due to its central position. The results of the external calibration are shown in Figure 3.

Figure 4 shows the reprojection error on all cameras. 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 0.3633 pixels with the standard deviation of 0.0486 pixels.

6. CALIBRATION BETWEEN REMOTE LOCATIONS

In this section we address the transformations between the coordinate systems attached to different teleimmersive locations and propose a simple calibration method to calibrate the geometry between the locations.

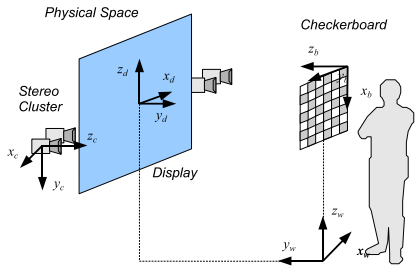

Figure 5: Coordinate systems for teleimmersion.

6.1 Coordinate Systems

Within the teleimmersive environment we deal with several different coordinate systems that are related to stereo reconstruction, display, and the virtual environment (see Figure 5 for details). The metric consistency between different coordinate systems is preserved as long as all the calibration procedures are performed in Euclidean space using the same units. 3D reconstruction from multiple cameras yields data expressed in the coordinate system of the reference camera (C_c) determined through the calibration described above. To achieve geometric consistency between multiple sites, data should be transformed into a physical world coordinate system (C_w) which is defined by a common plane (e.g. floor or table top) that all the locations share. For the calibration of the workspace, we use a checkerboard placed in horizontal orientation. The checkerboard coordinate system (C_b) is defined in the top-left corner (black square corner in OpenCV grid detection), while the x-axis points along the horizontal side and the y-axis along the vertical side of the board.

6.2 Calibration between Camera Space and Physical Space

We propose a simple method for calibrating the reference cluster (i.e. camera space) to the physical space. By consistently defining the physical coordinates across the remote locations, the geometric correspondence between the sites will be well-defined.

To define the transformation between the camera and physical space, we must calibrate the reference camera to the physical coordinate system (Figure 5). The calibration is performed by acquiring one image of the checkerboard placed in the vertical position. Since the camera has already been internally calibrated, it is possible to determine the location of the camera with respect to the checkerboard coordinate system.

The checkerboard is mounted on a tripod to properly fix it in this position. From the captured grid image, we can determine the orientation (R_b^c) and position (T_b^c) of C_b coordinate system with regard to C_c coordinate system. If the origin of the physical coordinate system is defined on the floor, the offset between the systems is $T_w^b = (h, w/2, 0)^T$ from the origin, where h is the vertical distance of the origin from the floor and w is the width of the board in metric units. The orientation of the physical coordinate system can be arbitrary but to simplify the measurements and calculations, the two systems are aligned.

Once we define orientation matrix R_w^b, the combined transformation of the physical coordinate system with regard to

Digital Object Identifier: 10.4108/ICST.IMMERSCOM2009.6119
http://dx.doi.org/10.4108/ICST.IMMERSCOM2009.6119

the camera coordinate system is calculated as follows:

$$R_w^c = R_b^c \cdot R_w^b$$
$$T_w^c = R_b^c T_w^b + T_b^c \quad (13)$$

The transformation above defines the orientation and position of the physical coordinate system in the coordinate system of the reference camera. When transforming a point from the camera space into the physical world space, the following equation is used:

$$X_i^w = {R_w^c}^T \cdot (X_i^c - T_w^c) \quad (14)$$

7. CONCLUSIONS

In this paper we have presented calibration framework which can be applied to multi-camera systems. The calibration is approached in a hierarchical fashion where cameras arranged in (stereo) clusters are calibrated internally and externally with a checkerboard while the external calibration is performed with a virtual calibration object generated by LED markers. Such approach allows easy deployment of portable cameras for stereo reconstruction and teleimmersion.

Our contribution is mainly in the external calibration approach using LED markers. We use vision graph analysis to calibrate camera setups where all the cameras do not share common workspace. Our novel parameterization of the two-marker approach adds robustness to the algorithm allowing more accurate camera calibration in presence of noise as we have demonstrated in our past experiments with the synthetic data [6]. The results obtained on the real camera setup presented in this paper show that our approach compensates for error propagation when the path transformation includes two to three nodes. 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. Our algorithm, in contrast to [15], also reconstructs metric information on camera positions. 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.

In the paper, we have also addressed geometric correspondence between several remote locations by proposing a simple calibration procedure to preserve metric space between the locations.

8. REFERENCES

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

[2] 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.

[3] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Trans. Graph.*, 22(3):819–827, 2003.

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

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

[6] G. Kurillo, Z. Li, and R. Bajcsy. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. In *Proceedings of 2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '08)*, Stanford, CA, September 7-11 2008. IEEE.

[7] G. Kurillo, R. Vasudevan, E. Lobaton, and R. Bajcsy. A framework for collaborative real-time 3d teleimmersion in a geographically distributed environment. In *Proceedings of IEEE International Symposium on Multimedia (ISM 2008)*, Berkeley, CA, December 15-17 2008.

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

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

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

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

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

[13] D. Nguyen and J. Canny. Multiview: spatially faithful group video conferencing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 799–808, Portland, Oregon, April 2-7 2005. ACM.

[14] B. Olsen and A. Hoover. Calibrating camera network using domino grid. *Pattern Recognition*, 34:1105–1117, 2001.

[15] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence*, 14(4):407–422, 2005.

[16] R. 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*, RA 3(4):323–344, 1987.

[17] Z. Zhang. Camera calibration with one-dimensional objects. Technical Report MSR-TR-2001-120, Microsoft Research, August 2002.

Digital Object Identifier: 10.4108/ICST.IMMERSCOM2009.6119
http://dx.doi.org/10.4108/ICST.IMMERSCOM2009.6119