# Calibration of the Teleimmersion System 

Dr. Gregorij Kurillo<br>Teleimmersion Lab, University of California, Berkeley

January 5, 2010

## Contents

1 Introduction ..... 2
2 Geometric Calibration of the Stereo Cluster ..... 3
2.1 Camera Model ..... 3
2.2 Intrinsic Calibration of the cluster ..... 4
2.3 Results ..... 6
3 Photometric Calibration of the Stereo Cluster ..... 7
3.1 Results ..... 7
4 External Geometric Calibration of Multiple Stereo Clusters ..... 8
4.1 Sub-pixel LED Marker Detection ..... 9
4.2 Pairwise External Calibration ..... 11
4.2.1 Epipolar Geometry and Essential Matrix ..... 11
4.2.2 Scale Factor Determination ..... 12
4.2.3 Stereo Triangulation ..... 13
4.3 Multi-camera External Calibration ..... 14
4.3.1 Graph ..... 15
4.3.2 Camera Pose Calculation ..... 15
4.3.3 Global Optimization ..... 16
4.4 Results ..... 19

## 1 Introduction

Note: More detailed explanation of the procedures described in this paper can be found in [8].
In this paper we approach geometric and photometric calibration of multi-camera system used for tele-immersion [7, 9]. The tele-immersion apparatus consists of 48 Dragonfly cameras (Point Grey Research Inc, Vancouver, Canada) which are arranged in 12 clusters covering $360^{\circ}$ view of the user(s). The cameras, equipped with 6 and 3.8 mm lenses, are mounted on an aluminum frame with dimensions of about $4.0 \times 4.0 \times 2.5 \mathrm{~m}^{3}$. The arrangement of cameras was optimized to increase the usable workspace coverage to about $2.0 \times 2.0 \times 2.5 \mathrm{~m}^{3}$. Each cluster consists of three black and white cameras intended for stereo reconstruction and a color camera used for texture acquisition. The viewing workspace of the four cameras in each cluster maximally overlaps at the distance of about 2.5 m . The cameras therefore share very similar views of the same scene. This property can be exploited for geometric and photometric calibration.

In the tele-immersion application the (3D) stereo reconstruction requires accurate geometric calibration. The 3D reconstruction is performed independently on each cluster while the data from different clusters are combined into the final model inside the renderer. The geometric and photometric properties of the grayscale cameras within the cluster influence the stereo reconstruction quality (errors), while the external calibration of the clusters affects the quality of the projection of the individual point clouds into the 3D space. Photometric calibration of the color cameras directly affects pixel (texture) blending for the rendering of the model. Considering our setup, the photometric and geometric calibration can be performed hierarchically as it is described in the following sections.

The accuracy of the stereo reconstruction is greatly affected by the quality of the camera calibration. The errors can occur due to lens distortion, misalignment between image and lens plane, and deviations of position and rotation between the stereo cameras [18],[16]. The quality of the stereo reconstruction is further affected by the photometric calibration of the cameras, illumination conditions, texture and color of the objects.

For the geometric calibration of the camera system we propose hierarchical approach defined in two steps. In the first step each cluster is independently calibrated for the internal (i.e. calibration matrix $K$ and lens distortion) and external camera parameters (i.e. position and orientation vectors) to define relative positioning of each camera inside the cluster to the reference camera. The intrinsic calibration of the cluster is performed by the well-known Tsai algorithm [15] using a checkerboard. The central camera is chosen as the reference cameras. Once all the parameters are found and optimized, the geometric calibration of the external parameters can be performed. In this step we only require to obtain the position and orientation of each reference camera for all the stereo clusters. The external parameters of the cameras can be defined in the coordinate system of a calibration object or of one of the reference cameras. We decide on the later choice since our calibration will not be performed with a fixed rigid object.

The photometric calibration between the cameras inside the cluster affects stereo reconstruction since the stereo matching is image-based. Different grayscale values of the same feature or area of the
scene detected by the cameras may affect the matching algorithm which performs the correlation between the two parts of the image pair. We propose optimization procedure to perform more accurate photometric calibration of the cameras within the cluster after initial manual adjustment. Using the Levenberg-Marquardt (LM) algorithm, histograms are optimized by adjusting cameras brightness and shutter setting.

Finally, we will calibrate the color camera to provide accurate blending of the textures obtained from the different clusters. For this purpose we use the histogram of the MacBeth ColorChecker chart to adjust the white balance of the cameras.

For the photometric calibration we assume that the illumination is somewhat homogeneous inside the tele-immersion system. The illumination inside is diffused by filters to reduce shadows and specular highlights which interfere with the stereo matching algorithm. We assume Lambertian properties for the object surfaces that will be introduced inside the environment.

## 2 Geometric Calibration of the Stereo Cluster

### 2.1 Camera Model

The geometric calibration of the four cameras inside the stereo cluster is based on Tsai algorithm [15]. A planar target placed in different positions and orientations is used to generate a set of points for homography calculation between the camera model and the plane. We use the standard pinhole camera model while considering radial and tangential distortion models:

$$
\begin{gather*}
{\left[\begin{array}{c}
x_{p} \\
y_{p} \\
1
\end{array}\right]=\left[\begin{array}{ccc}
f_{x} & \alpha & c_{x} \\
0 & f_{y} & c_{y} \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{array}\right]\left[\begin{array}{cc}
\mathbf{R}_{i} & \mathbf{T}_{i} \\
& \\
0 & 1
\end{array}\right]\left[\begin{array}{c}
X_{0} \\
Y_{0} \\
Z_{0} \\
1
\end{array}\right]}  \tag{1}\\
\mathbf{x}_{p}=\mathbf{K}_{f} \mathbf{\Pi}_{0} \mathbf{g} \mathbf{X}_{0} \tag{2}
\end{gather*}
$$

The model in Eq. 1 represents the transformation from a point $\mathbf{X}_{0}$ seen by camera to the corresponding image pixel coordinate $\mathbf{x}_{p}$ defined on the image plane. The parameters $\left(f_{x}, f_{y}\right)$ represent horizontal and vertical focal length of the camera lens (expressed in pixels), $\left(c_{x}, c_{y}\right)$ represent the optical center of the camera which is normally located close to the image center and parameter $\alpha$ represents the skew angle between the x - and y -axes of the image plane. In most cases $\alpha$ can be set to 1 . The matrix $\mathbf{R}_{i}$ represents $3 \times 3$ rotation matrix comprising of three rotation angles which define camera orientation with respect to the object coordinate system. Finally, the vector $\mathbf{T}_{i}$ defines 3D position of the camera center from the object coordinate system origin. The lens system of a real camera creates radial and tangential shift of the projected points, mainly noticeable at the image corners. The lens distortion is modeled by two parameters of radial distortion $\left(k_{1}\right.$, $\left.k_{2}\right)$ and two parameters for tangential distortion $\left(p_{1}, p_{2}\right)$. In total, our camera model consists of 8


Figure 1: Typical image of the checkerboard calibration object as captured by four cameras in the cluster
internal parameters and 6 external parameters which have to be determined in the first step of the calibration.

### 2.2 Intrinsic Calibration of the cluster

The algorithm for camera calibration uses a set of known points $\mathbf{X}_{i}(\mathrm{i}=1,2, \ldots \mathrm{~N})$ 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. The accuracy of the calibration depends on the number of points, size of the checkerboard squares, and the distribution of the points over the camera viewing volume. Linear solution of the camera parameters can be obtained by writing a set of equations from the camera model equation 1 for the transformation of known grid coordinates $\mathbf{X}_{i}$ into 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 checkerboard positions. Next non-linear refinement of parameters is needed to eliminate the effect of noise in the data. Levenberg-Marquardt (LM) algorithm is applied for this purpose. 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:

$$
\begin{equation*}
e_{i}=\sum_{p=1}^{P} \sum_{j=1}^{M} F\left(\mathbf{K}_{i}, \mathbf{R}_{i}^{p}, \mathbf{t}_{i}^{p}\right) \tag{3}
\end{equation*}
$$

After each camera is calibrated independently, the relative orientation and position of the cameras within the cluster $\left(R_{i 0}, t_{i 0}\right)$ needs to be obtained. The relative relationship between an arbitrary camera $C_{i}$ and selected reference camera $C_{0}$ can be expressed as follows:

$$
\begin{equation*}
\mathbf{R}_{i 0}=\mathbf{R}_{i} \mathbf{R}_{0}^{-1}, \quad \mathbf{t}_{i 0}=\mathbf{t}_{i}-\mathbf{R}_{i} \mathbf{R}_{0}^{-1} \mathbf{t}_{0} \tag{4}
\end{equation*}
$$

In Eq. 4 the parameters $\left(\mathbf{R}_{0}, \mathbf{t}_{0}\right)$ denote the orientation and position of the reference camera and $\left(\mathbf{R}_{i}, \mathbf{t}_{i}\right)$ denote the orientation and position of the arbitrary camera with regard to the current checkerboard position. Due to noise, $\left(\mathbf{R}_{i 0}, \mathbf{t}_{i 0}\right)$ parameters will slightly vary for different checkerboard positions. Average value of the parameters is used in further computations.

To replace $\left(\mathbf{R}_{i}, \mathbf{t}_{i}\right)$ in Eq. 1, the transformation in Eq. 4 is rearranged:

$$
\begin{equation*}
\mathbf{R}_{i}=\mathbf{R}_{i 0} \mathbf{R}_{0}, \quad \mathbf{t}_{i}=\mathbf{t}_{i 0}+\mathbf{R}_{i 0} \mathbf{t}_{0} \tag{5}
\end{equation*}
$$

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

$$
x_{i}=\mathbf{K}_{f i} \boldsymbol{\Pi}_{0}\left[\begin{array}{cc}
\mathbf{R}_{i 0} \mathbf{R}_{0} & \mathbf{t}_{i 0}+\mathbf{R}_{i 0} \mathbf{t}_{0}  \tag{6}\\
0 & 1
\end{array}\right] \mathbf{X}_{0}
$$

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:

$$
\begin{equation*}
e_{\text {total }}=\sum_{p=1}^{P} \sum_{i=1}^{N} \sum_{j=1}^{M} F\left(\mathbf{K}_{i}, \mathbf{R}_{i 0}, \mathbf{t}_{i 0}, \mathbf{R}_{0}^{p}, \mathbf{t}_{0}^{p}\right) \tag{7}
\end{equation*}
$$

Since the internal parameters of the cameras are independent and have already been optimized, only external parameters are considered in this optimization. If the internal parameters are fixed, lower reprojection errors can be obtained. In total $6 \times(N-1)+6 \times P$ parameters need to be optimized. For the final optimization checkerboard positions that produced large error on some cameras are removed to reduce the global error. Average error, standard deviation and the maximal error are analyzed to determine the quality of the calibration.

### 2.3 Results

For the calibration of the clusters, we used a black and white checkerboard with $15 \times 10$ 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 [10] libraries. Numerical approximation of the Jacobian matrix was used for the optimization process.


Figure 2: (a) Reprojection error of the grid points captured by the color camera and projected to the reference camera image plane. (b) Histogram of the reprojection error for the same image.

Each of the cameras is first calibrated independently. The maximal reprojection errors are checked for each image and images with high maximal reprojection error (e.g., $\leq 1 \mathrm{px}$ ) are removed from the set. Afterwards the internal and external parameters on each camera are further optimized by non-linear Levenberg-Marquardt (LM) optimization algorithm. Next, global non-linear optimization is applied. The grid points captured by individual cameras for different checkerboard positions are projected to the image plane and compared with the points detected from the images. Reprojection error is calculated as RMS between the reference camera grid point and the reprojected grid point. All images and all cameras are considered in the optimization loop of the LM algorithm.

Figure 2 shows a typical error distribution as obtained in a single image. After the global optimization on all four cameras, the error distribution resembles Gaussian distribution (Figure 3) with the mean value of 0.18 pixels and standard deviation of 0.11 pixels. The maximal error for this set of images was 0.8 though only a small number of points had errors in that range. Figure 4 shows detailed view of the reprojection error across the image plane. The results show that the highest errors occurred in the top section.


Figure 3: a) Reprojection of grid points for all cameras and all images (all checkerboard positions) as projected to the reference camera. (b) Colormap of the reprojection error points.

## 3 Photometric Calibration of the Stereo Cluster

### 3.1 Results



Figure 4: (a) 3D view of the average reprojection error for all the checkerboard positions and all the cameras. (b) The histogram of the reprojection error for all cameras (4 cameras) and all checkerboard positions (16 images). The mean reprojection error is 0.18 pixels (standard deviation: 0.11 pixels)

## 4 External Geometric Calibration of Multiple Stereo Clusters

After calibrating each stereo cluster, the position and orientation of the clusters with regard to a common (reference) coordinate system is required. In this section we present the main idea for the calibration of the external parameters of the clusters. The middle, grayscale, camera is chosen as the reference camera in the cluster. Since the cluster is internally calibrated, we do not need to adjust any of the internal camera parameters of the relative positioning of cameras within a cluster. Thus, we need to determine only 6 external parameters for each cluster (i.e. orientation matrix $R$ and position vector $T$ ).

For the calibration we decide on calibration method with LED marker(s). Calibration via other methods is possible but due to the large volume coverage construction of such objects with sufficient accuracy would be difficult to achieve. Another reason for selecting calibration with markers is due to small overlap among some of the cameras (e.g. set of cameras capturing top part of the body and set of cameras for the lower part of the body). Several researchers have shown high accuracy for geometric calibration using one dimensional objects [17][13][4][3].

In this approach we calibrate cameras pairwise since the overlap between arbitrary chosen reference camera and the remaining cameras is limited. From the number of common points between different camera pairs, weighted graph representing the camera set-up is constructed. Dijkstra algorithm for optimal path [2] is used to determine optimal transformation between the cameras. We assume that higher number of common points will result in better calibration.

Our algorithm for external multi-camera calibration can be summarized as follows:


Figure 5: MacBeth chart as captured by the stereo pair of cameras before and after photometric correction.

- image acquisition and marker detection on multiple cameras
- determination of overlap
- composition of adjacency matrix based on the number of common points between camera pairs
- computation of fundamental matrix $\mathbf{F}$ using RANSAC and computation of essential matrix $E$ from known internal parameters ( $\mathbf{K}, \mathbf{k}_{c}$ )
- essential matrix decomposition to obtain $\mathbf{R}, \mathbf{T}$ up to a scale factor $\lambda$
- determination of the scale factor $\lambda$ through triangulation and LM optimization
- finding optimal path in the weighted graph from the reference camera to the remaining cameras
- global optimization of the parameters


### 4.1 Sub-pixel LED Marker Detection

In camera calibration the accuracy of detecting the correct position of the marker center can significantly influence calibration quality. Marker detection has to be reliable and highly accurate. Detection can be influenced by CCD noise, uniformity of the background, other sources of illumination in the scene and poor thresholding of the captured image [14].

For our geometric calibration we used a rigid bar with two LED markers attached on each end. The LED were emitting in visible and IR spectrum. The distance between the two markers was approximately 320 mm .


Figure 6: Histogram of the captured MacBeth chart before and after photometric correction.

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 to reduce the visible scene to LEDs. For each frame, the image was first converted to grayscale (if needed) and then thresholded with threshold of about 120-150. To remove noise, erosion and dilatation were applied. Next, ellipse fitting algorithm was implemented to find circular images of markers while eliminating any large or oddly shaped objects left in the scene after the initial processing. The range of ellipse size was determined experimentally (e.g. expected marker size was up to 15 pixels, the main axes ratio between 0.6 and 1.5). If the number of ellipsoidal objects found in the image matched the required number of markers (i.e. two), marker center was calculated.

To calculate sub-pixel marker center we used squared gray scale centroid (SGSC) method. According to Shortis et al. [14] SGSC method is appropriate for small and large LED targets. The input for SGSC 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:

$$
\begin{equation*}
\bar{x}=\sum_{j=1}^{m} \sum_{i=1}^{n} i \cdot I_{i, j}^{2} / \sum_{j=1}^{m} \sum_{i=1}^{n} I_{i, j}^{2} \tag{8}
\end{equation*}
$$

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 equation 8 is calculated for rows $(i)$ and columns $(i)$ separately.

### 4.2 Pairwise External Calibration

Given two images from calibrated cameras, camera pose and position of the points in space can be obtained through epipolar geometry. For any two pairs of cameras, the internal camera parameters are already known from the internal (checkerboard) calibration. First, all the effects of the camera parameters are undone by undistorting and normalizing the image points on all cameras. Afterwards, we can apply epipolar geometry constraints and calculate essential matrix. 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.

### 4.2.1 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 $\mathbf{x}_{i 1}$ and $\mathbf{x}_{i 2}$ lie on the same plane $\mathbf{P}_{i}[5]$. The geometric relationship between the two cameras can be described by the fundamental matrix $\mathbf{F}$ and the following relation:

$$
\begin{equation*}
\mathbf{x}_{i 2}^{T} \mathbf{F} \mathbf{x}_{i 1}=0 \tag{9}
\end{equation*}
$$

The fundamental matrix includes the internal parameters of the cameras $\left(\mathbf{K}_{1}, \mathbf{K}_{2}\right)$ 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 8-point algorithm with RANSAC [todo:cite] implemented in OpenCV.

When describing the relationship between normalized image coordinates ( $\hat{\mathbf{x}}=\mathbf{K}^{-1} \mathbf{x}$ ), specialized fundamental matrix denoted as essential matrix $(\mathbf{E})$ is defined. The essential matrix has fewer degrees of freedom as compared to the fundamental matrix and it only contains relative pose between the two cameras (views). Considering the two definitions, the following relationship exists between the fundamental and essential matrices:

$$
\begin{equation*}
\mathbf{F}=\mathbf{K}_{2}^{-T} \mathbf{E} \mathbf{K}_{1}^{-1} \quad \text { and } \quad \mathbf{E}=\mathbf{K}_{2}^{T} \mathbf{F} \mathbf{K}_{1} \tag{10}
\end{equation*}
$$

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

$$
\begin{equation*}
\hat{\mathbf{x}}_{i 2}^{T} \mathbf{E} \hat{\mathbf{x}}_{i 1}=0 \tag{11}
\end{equation*}
$$

In the subsequent text we omit symbol ${ }^{\wedge}$ and assume that the image coordinates have been normalized, unless otherwise stated.

The essential matrix is defined as follows:

$$
\begin{equation*}
\mathbf{E}=[\mathbf{t}]_{\times} \mathbf{R}=\hat{\mathbf{T}} \mathbf{R} \tag{12}
\end{equation*}
$$

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 six
degrees of freedom, the essential matrix has only five degrees of freedom where the vector $\mathbf{t}$ can only be obtained up to a scale factor. 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 [5]:

$$
\begin{equation*}
(\mathrm{SVD}) \mathbf{E}=\mathbf{U} \boldsymbol{\Sigma} \mathbf{V} \quad \text { with } \quad \boldsymbol{\Sigma}=\operatorname{diag}\{\sigma, \sigma, 0\} \tag{13}
\end{equation*}
$$

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

$$
\begin{align*}
& \left(\hat{\mathbf{T}}_{1}, \mathbf{R}_{1}\right)=\left(\mathbf{U R}_{Z}\left(+\frac{\pi}{2}\right) \boldsymbol{\Sigma} \mathbf{U}^{T}, \mathbf{U R}_{Z}^{T}\left(+\frac{\pi}{2}\right) \mathbf{V}^{T}\right)  \tag{14}\\
& \left(\hat{\mathbf{T}}_{2}, \mathbf{R}_{2}\right)=\left(\mathbf{U R}_{Z}\left(-\frac{\pi}{2}\right) \boldsymbol{\Sigma} \mathbf{U}^{T}, \mathbf{U R}_{Z}^{T}\left(-\frac{\pi}{2}\right) \mathbf{V}^{T}\right)
\end{align*}
$$

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 $180^{\circ}$ about the line joining the two camera centers. In only one of the solutions a reconstructed point $X$ 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 of the two cameras.

To optimize the results for $\mathbf{R}$ and $t$, we apply LM algorithm for bundle adjustment [12] with three rotational and three position parameters as input. The following function is minimized:

$$
\begin{equation*}
\Phi(\mathbf{R}, \mathbf{T})=\sum_{j=1}^{N} \frac{\left(\tilde{\mathbf{x}}_{2}^{j T} \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_{1}^{j T}\right)^{2}}{\left\|\hat{\mathbf{e}}_{3} \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_{i}^{j}\right\|^{2}}+\frac{\left(\tilde{\mathbf{x}}_{2}^{j T} \hat{\mathbf{T}} \mathbf{R} \tilde{\mathbf{x}}_{1}^{j T}\right)^{2}}{\left\|\tilde{\mathbf{x}}_{2}^{j T} \hat{\mathbf{T}} \mathbf{R} \hat{\mathbf{e}}_{3}^{T}\right\|^{2}} \tag{15}
\end{equation*}
$$

where $\hat{\mathbf{e}}_{3}$ is the anti-symmetric matrix of vector $\mathbf{e}_{3}=[0,0,1]^{T}$. The expression (15) is based on the properties of the epipolar geometry: $\mathbf{x}_{1}^{j T} \mathbf{e}_{3}=1, \mathbf{x}_{2}^{j T} \mathbf{e}_{3}=1$ and $\mathbf{x}_{2}^{T} \mathbf{E x}_{1}=0$.

From the essential matrix decomposition, the position vector $\mathbf{t}$ is obtained only up to a scale factor $\lambda$. The scale factor $\lambda$ can be obtained using known geometry from the 3D scene [3].

### 4.2.2 Scale Factor Determination

In case of our camera calibration, the unknown scale factor $\lambda$ is obtained from the known dimension of the calibration bar (i.e. distance between the two LED markers). Pair of points $\hat{X}_{1}$ and $\hat{X}_{2}$ in the normalized 3D space can be reconstructed from their respective images using stereo triangulation while their coordinates in the absolute 3D space ( $X_{1}$ and $X_{2}$ ) remain unknown. Their relative position, however, is defined by the distance $\hat{d}$ between the normalized points and the true length of the calibration bar $d_{0}$. The scale factor $\lambda$ can be determined as follows:

$$
\begin{equation*}
\left(\mathbf{X}_{1}-\mathbf{X}_{2}\right)=\lambda\left(\hat{\mathbf{X}}_{1}-\overline{\mathbf{X}}_{2}\right) \Rightarrow \lambda=\frac{d_{0}}{\bar{d}} \tag{16}
\end{equation*}
$$

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

$$
\begin{equation*}
\bar{\lambda}=\frac{d_{0}}{N} \sum_{i=1}^{N} \frac{1}{d_{i}} \tag{17}
\end{equation*}
$$

To further improve the accuracy of the scale determination, we implement non-linear optimization of the obtained solution using LM algorithm [10] where we minimize the error in the distance of the estimated $d$ and the true bar length $d_{0}$ :

$$
\begin{equation*}
\delta(\lambda)=\sum_{i=1}^{N} d_{0}-\left\|\mathbf{X}_{1 i}(\lambda)-\mathbf{X}_{2 i}(\lambda)\right\| \tag{18}
\end{equation*}
$$



Figure 7: Reconstructed length of the calibration bar before (left) and after non-linear optimization (right).

### 4.2.3 Stereo Triangulation

Below is the stereo triangulation algorithm that was implemented in the calibration. Note, that input points first have to be normalized.

Let a 3D point $P$ have coordinates $\mathbf{X}_{L}=\left[X_{L}, Y_{L}, Z_{L}\right]$ and $\mathbf{X}_{R}=\left[X_{R}, Y_{R}, Z_{R}\right]$ in the left and right camera reference frames, respectively. The two vectors are related to one another with the following rigid transformation:

$$
\begin{equation*}
\mathbf{X}_{R}=\mathbf{R} \mathbf{X}_{L}+t \quad \text { and } \quad \mathbf{X}_{L}=\mathbf{R}^{T}\left(\mathbf{X}_{R}-t\right) \tag{19}
\end{equation*}
$$

where the position of the right camera relatively to the left is described by the rotation $R$ and translation vector $t$.

The projection of the 3D point to both image planes are defined with the homogeneous vectors $\mathbf{x}_{L}=\mathbf{X}_{L} / Z_{L}=\left[x_{L}, y_{L}, 1\right]^{T}$ and $\mathbf{x}_{R}=\mathbf{X}_{R} / Z_{R}=\left[x_{R}, y_{R}, 1\right]^{T}$.

In the triangulation algorithm, we try to retrieve 3 D point coordinates $\mathbf{X}_{L}$ and $\mathbf{X}_{2}$ from image points $\mathbf{x}_{L}$ and $\mathbf{x}_{R}$. From equations 19, the following expression can be obtained [cite]:

$$
\begin{gather*}
u=\mathbf{R} x_{1} \\
Z_{1}=\frac{\left\langle u, x_{2}\right\rangle\left\langle x_{2}, t\right\rangle-\left\|x_{2}\right\|^{2}\langle u, t\rangle}{\left\|x_{1}\right\|^{2}\left\|x_{2}\right\|^{2}-\left\langle u, x_{2}\right\rangle^{2}} \\
Z_{2}=\frac{\left\|x_{1}\right\|^{2}\left\langle x_{2}, t\right\rangle-\langle u, t\rangle\left\langle u, x_{2}\right\rangle}{\left\|x_{1}\right\|^{2}\left\|x_{2}\right\|^{2}-\left\langle u, x_{2}\right\rangle^{2}}  \tag{20}\\
\mathbf{x}_{1}^{\prime}=\mathbf{x}_{1} \cdot \mathbf{Z}_{1} \\
\mathbf{x}_{2}^{\prime}=\mathbf{R}\left(\mathbf{x}_{2} \cdot \mathbf{Z}_{2}-\mathbf{t}\right)
\end{gather*}
$$

where $\langle.,$.$\rangle denotes scalar product of two vectors and \|$.$\| denotes L2 norm of a vector.$
The left camera coordinates:

$$
\begin{equation*}
\mathbf{X}_{L}=\frac{1}{2}\left(\mathbf{x}_{1}^{\prime}+\mathbf{x}_{2}^{\prime}\right) \tag{21}
\end{equation*}
$$

and the right camera coordinates:

$$
\begin{equation*}
\mathbf{X}_{R}=\mathbf{R} \mathbf{X}_{L}+\mathbf{t} \tag{22}
\end{equation*}
$$



Figure 8: Reconstructed position of the calibration bar as obtained from two calibrated cameras using stereo triangulation. Note that the 3D points are expressed with regard to the camera coordinate system.

### 4.3 Multi-camera External Calibration

After calibrating all the camera pairs with the highest number of overlapping points, ....

### 4.3.1 Graph

We represent the structure of the multi-camera system using tools of graph theory [2]. The layout of $M$ cameras is represented by graph $G$ consisting of $M$ vertices $V_{i}$ which represent individual cameras (i.e. the central cameras of each cluster). In order for the calibration of the entire camera system to be possible, the graph of the camera layout must 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)[6]. We describe the overlap between different camera pairs by assigning weights to the graph edges. The weights $\omega_{i j}$ correspond to the number of captured LED points by both cameras simultaneously. The weight value is defined as follows:

$$
\omega_{i j}=\left\{\begin{array}{cc}
\frac{1}{M} & M \text { common points exist between } i \text { and } j  \tag{23}\\
0 & \text { if } i=j \\
-1 & \text { no common points (or calibration failed) }
\end{array}\right.
$$

To describe the graph structure, adjacency matrix $\mathbf{A}(G)$ is used. In the case of a weighted graph, the elements of the adjacency matrix $a_{i j}$ correspond to the weights $\omega_{i j}$. The diagonal elements of the adjacency matrix for graphs with no self-connected vertices are zero. The weights $\omega_{i j}$ can be further modified to prioritize certain features between the cameras, such as closeness to the reference camera, accuracy of the internal calibration etc.

The adjacency matrix of the camera setup is updated after all camera pairs are calibrated. In case of failed calibration between two cameras (e.g. not sufficient number of common points, too few points returned by RANSAC, too large error), the corresponding matrix element is assigned value -1 .

After the relative poses between all the camera pairs have 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. The construction of the adjacency matrix for the weighted graph already considers the first requirement.

To find the optimal path from the reference camera to all the other cameras, we employ Dijkstra's shortest path algorithm [2]. 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.

### 4.3.2 Camera Pose Calculation

Using the shortest path from the reference camera to each camera, we can calculate the absolute position of the clusters. 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
(a)

(b)


Figure 9: (a) Generated graph after calibration of 7 cameras. Number of common points are shown at the edges with the corresponding weights in the parenthesis. For clarity, the connections with less than 200 common points are omitted. (b) Optimal path from reference node \# 3 was found by Dijkstra algorithm.
as $\left(\mathbf{R}_{i j}, \mathbf{t}_{i j}\right)$ and $\left(\mathbf{R}_{j k}, \mathbf{t}_{j k}\right)$. The transformation from $i$ to $k$ can be calculated as follows:

$$
\begin{equation*}
\mathbf{t}_{i k}=\mathbf{t}_{i j}+\mathbf{R}_{j k} \mathbf{t}_{j k} \quad \text { and } \quad \mathbf{R}_{i k}=\mathbf{R}_{j k} \mathbf{R}_{i j} \tag{24}
\end{equation*}
$$

If a path from the reference camera has a length longer than two, the equation (24) is applied sequentially to cover the entire path.

### 4.3.3 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 a selected reference camera. For this global solution, the reprojection error of a single 3D point located in the space of all cameras should be minimized. Since the viewing volumes of the cameras may not overlap with each other or the reference camera, the point may not be projected on all image planes. 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 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 bundle adjustment problem, where there is lack of interaction between different 3D points and cameras, sparse bundle adjustment (SBA) can be applied [11]. The SBA
algorithm assumes we have $n$ 3D points which are seen by $i t m$ cameras. Projection of $i$-th point on camera plane $j$ is denoted as $\mathbf{x}_{i j}$. Each camera can be parametrized by a vector $\mathbf{a}_{j}$ and each 3D point $i$ by a vector $\mathbf{b}_{i}$. Function $d(\mathbf{x}, \mathbf{y})$ denotes Euclidean distance between between image points represented by $\mathbf{x}$ and $\mathbf{y}$. Bundle adjustment minimizes the following reprojection error:

$$
\begin{equation*}
\min _{\mathbf{a}_{j}, \mathbf{b}_{i}} \sum_{i=1}^{n} \sum_{j=1}^{m} d\left(\mathbf{Q}\left(\mathbf{a}_{j}, \mathbf{b}_{i}\right), \mathbf{x}_{i j}\right)^{2} \tag{25}
\end{equation*}
$$

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.


Figure 10: Reprojection of the two calibration bar markers into image planes of the cameras.
To obtain the position of 3D points in the reference camera coordinate system, we use pairwise stereo triangulation to obtain the position of points with regard to a particular camera pair. Finally we transform the points into the reference camera coordinate system using the initial guess of the camera pose. Since some points may be seen by several cameras, average point location is calculated for the final 3D point position. To remove outliers, we check the distance between the two 3D points of the calibration bar. If the distance is larger than the selected threshold (e.g. error greater than $1.0 \%$ of distance), the point pair is removed from the set (i.e. two points are set as invisible to all cameras).

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 3 D point, and 3 rotational and 3 translational parameters) and the dimensions of the measurement space are $N \leq n \times 2\left(\mathbf{x}_{i j} \in \mathbb{R}^{2}\right)$. In general all cameras will not see each 3 D point, therefore the dimension of $N$ will be smaller. Overall, the parameter space is very large. For example, 5003 D points observed by 5 cameras will be defined by 1524 parameters while the image projection space will be $\leq 5000$. Sparsity of the parameter interaction can reduce the computation time. The number of the input parameters can be reduced if we take into account the rigid connection between each 3 D point pair on the calibration bar. The position of the two 3 D points can be described by the starting point $\mathbf{X}_{i, 1}$ while the location of the 2 nd point is defined by the normalized direction vector $\mathbf{n}_{i}$ between the two points and their distance $d_{0}$ which is kept constant for all image pairs. The normal can be parameterized as follows:

$$
\mathbf{n}_{i}=\frac{\mathbf{X}_{i, 2}-\mathbf{X}_{i, 1}}{\left\|\mathbf{X}_{i, 2}-\mathbf{X}_{i, 1}\right\|}=\left[\begin{array}{c}
n_{i x}  \tag{26}\\
n_{i y} \\
n_{i z}
\end{array}\right]=\left[\begin{array}{c}
n_{i x} \\
\sqrt{1-n_{i x}^{2}-n_{i y}^{2}} \\
n_{i z}^{2}
\end{array}\right]
$$

Normal coordinate $n_{i y}$ is parametrized because we expect that the bar will mostly be in a vertical position and therefore its value will be close to 1 . Inside the LM loop we enforce the condition $n_{i x}^{2}+n_{i z}^{2} \leq 1$ to keep the direction vector normalized. Finally, we can calculate the 2 nd coordinate of the point bar as follows:


Figure 11: Parametrization of calibration markers using normalized direction.
Above parametrization will decrease the parameter space size to $M=\frac{n}{2} \times 5+6 \times(m-1)$ or for our numerical example decrease the number of parameters from 1524 to 1274 . The dimensions of the measurement vector remains the same. The equation 25 is applied for reprojection minimization. The parameterization additionally constraints the LM optimization to keep the distance between the two 3D points constant.

For the implementation of our calibration algorithm we have used expert drivers of SBA library [11] where $\mathbf{a}$ and $\mathbf{b}$ were defined as follows:

$$
\begin{equation*}
\mathbf{a}=\left(\mathbf{T}_{1}^{T}, \mathbf{r}_{1}^{T}, \ldots, \mathbf{T}_{j}^{T}, \mathbf{r}_{j}^{T}, \ldots, \mathbf{T}_{m-1}^{T}, \mathbf{r}_{m-1}^{T}\right)^{T} \tag{28}
\end{equation*}
$$

and

$$
\begin{equation*}
\mathbf{b}=\left(\mathbf{X}_{1,1}^{T}, n_{1 x}, n_{1 z}, \ldots, \mathbf{X}_{i, 1}^{T}, n_{i x}, n_{i z}, \ldots, \mathbf{X}_{p, 1}^{T}, n_{p x}, n_{p z}\right)^{T} \tag{29}
\end{equation*}
$$

In above equation $p$ represents the number of image pairs which is equal to $\frac{n}{2}$. The measured image point coordinates across all cameras are defined in the following vector:

$$
\begin{equation*}
\mathbf{X}=\left(\mathbf{x}_{11}^{T}, \ldots, \mathbf{x}_{1 m}^{T}, \mathbf{x}_{21}^{T}, \ldots, \mathbf{x}_{2 m}^{T}, \ldots, \mathbf{x}_{n 1}^{T}, \ldots, \mathbf{x}_{n m}^{T}\right)^{T} \tag{30}
\end{equation*}
$$

Vector $\mathbf{X}$ is in practice sparse due to some 3D points not being visible in all cameras simultaneously. SBA algorithm takes into account the sparsity of the problem to increase computational speed and efficiency.

### 4.4 Results

Figure 12 shows example of 3D coordinate (for x-axis) estimation used for input into SBA optimization. Each 3D coordinate is estimated by calculating the average of reconstructed points in camera pairs where particular point is visible. The selection of camera pairs is based on the optimal path found in the camera graph. The point is reconstructed using triangulation and transformed into the reference coordinate system:

$$
\begin{equation*}
\mathbf{x}_{i}^{0}=\mathbf{R}_{0, j}^{T} \mathbf{x}_{i}^{j}-\mathbf{R}_{0, j}^{T} \mathbf{t}_{0, j} \tag{31}
\end{equation*}
$$

The results in Figure 12 show that the coordinates of reconstructed 3D points vary only slightly ( 12 mm ) between the cameras suggesting that the initial guess for the camera external calibration parameters is close enough to the optimal solution.

Figure 12 compares the results of marker distance calculation before and after running SBA optimization. If the optimization is performed on all the 3D points independently, the error is slightly reduced from but the mean distance may drift away from the "true" distance between the markers due to unconstrained optimization. If we constrain the 3D parameters using one point and normalized direction, the distance between the point pairs will be preserved and camera parameters will be adjusted accordingly.

Figure 14 shows the reprojection error during global optimization. The reprojection error represents mean error over all cameras as obtained by reprojecting the 3D points onto the image plane using current camera parameters. The error is calculated as quadratic error between the reprojected point and recorded image point. The final reprojection error is $0.05 \%$ which corresponds to about 0.28 px error on $640 \times 480$ image (?).


Figure 12: Initial estimate of 3D point coordinates (only x-coordinate is shown) obtained from several camera pairs. The 3D points are used for SBA algorithm to refine the external parameters of the cameras. Note that for some frames x -coordinate is set to 0 due to error thresholding.

Finally, the camera pose is refined using SBA optimization algorithm. Figure 15 shows the camera pose before and after optimization. The results show that the rotation and position of the cameras was only slightly adjusted.

## References

[1] Opencv: Open computer vision library. [web page]
http://sourceforge.net/projects/opencvlibrary/, November 2006.
[2] J. Bondy and U. Murty. Graph Theory with Applications. Elsevier Science Publishing Co., Inc., New York, 5th edition, 1982.
[3] N. Borghese and P. Cerveri. Calibrating a video camera pair with a rigid bar. In Proceedings of CVPR?, 1999.
[4] 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.
[5] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
[6] A. K. Hartman and M. Weigt. Phase transition in combinatorial optimization problems. WileyVCH Verlag, 2006.
[7] S. Jung and R. Bajcsy. A framework for constructing real-time immersive environments for training physical activities. Journal of Multimedia, 1(7):9-17, 2006.
[8] 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.
[9] G. Kurillo, R. Vasudevan, E. Lobaton, and R. Bajcsy. A framework for collaborative realtime 3d teleimmersion in a geographically distributed environment. In Proceedings of IEEE International Symposium on Multimedia (ISM 2008), Berkeley, CA, December 15-17 2008.
[10] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] http://www.ics.forth.gr/~1ourakis/levmar/, July 2004.
[11] 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. Available from http://www.ics.forth.gr/~1ourakis/sba.
[12] 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.
[13] M. Machacek, M. Sauter, and T. Rsgen. Two-step calibration of a stereo camera system for measurement in large vollumes. Institute of Physics Publishing, 14:1631-1639, 2003.
[14] M. Shortis, T. Clarke, and T. Short. A comparison of some techniques for the subpixel location of discrete target images. In S. F. El-Hakim, editor, Proceedings of SPIE Conference, volume 2350, pages 239-250, 1994.
[15] 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, RA3(4):323-344, 1987.
[16] 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, pages 292-296, 1991.
[17] Z. Zhang. Camera calibration with one-dimensional objects. Technical Report MSR-TR-2001120, Microsoft Research, August 2002.
[18] W. Zhao and N. Nandhakumar. Effects of camera alignment errors on stereoscopic depth estimates. Pattern Recognition, 29(12):2115-2126, 1996.


Figure 13: Estimation of bar length before and after optimization. Applying the constraint on the distance between the 3D point pairs preserves the true distance between the points.


Figure 14: Reprojection error during global optimization.


Figure 15: Camera pose before (green) and after (blue) global optimization using sparse bundle adjustment.

