The 2nd ACM Workshop on Video Surveillance and Sensor Networks, October 2004

Calibration of a Reconfigurable Array of Omnidirectional Cameras Using a Moving Person

Tarak Gandhi
Computer Vision and Robotics Research Laboratory
University of California at San Diego
La Jolla, CA, USA
tgandhi@ucsd.edu

Mohan M. Trivedi
Computer Vision and Robotics Research Laboratory
University of California at San Diego
La Jolla, CA, USA
mtrivedi@ucsd.edu

ABSTRACT

Reconfigurable arrays of omnidirectional cameras are useful for applications where multiple cameras working together are to be deployed at a short notice. This paper addresses the important issue of calibration of such arrays in terms of the relative camera positions and orientations. The location of a one-dimensional object moving parallel to itself, such as a moving person is used to establish correspondences between multiple cameras. In such case, the non-linear 3-D problem of calibration can be approximated by a 2-D problem in plan view. This enables an initial solution using factorization method. A non-linear optimization stage is then used to account for the the approximations, as well as to minimize the geometric error between the observed and projected omni pixel coordinates. Experimental results with simulated and real data illustrate the effectiveness of the method.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene analysis—*Stereo*

General Terms

Algorithms

Keywords

Calibration, Panoramic vision, Stereo vision, Surveillance

1. INTRODUCTION AND MOTIVATION

Omnidirectional cameras which give a 360 degree field of view of the surroundings have been of great interest to vision researchers [1, 3, 7]. A network of multiple omni based camera arrays can offer advantages of wide area coverage, redundancy, abstraction at multiple levels such as detection, tracking and recognition. Arrays of omnidirectional as well as rectilinear cameras may be integrated with the infrastructure for applications such as surveillance, people tracking, face detection, traffic analysis, and infrastructure monitoring. In addition to the fixed cameras, there may also be arrays of cameras deployed as needed to work as a part of the larger system. Such arrays would give advantages of reconfigurability, flexibility, and ad hoc deployment and be of value in the new generation of Sensor Networks capable of monitoring dynamic events from remote environments. In applications such as structural health monitoring of civil infrastructures, reconfigurable video arrays can provide important collaborative visual record.

However, in order to ensure that multiple cameras would work in an integrated manner, it is necessary to calibrate the cameras with respect to each other. However, measuring locations and orientations of the cameras is laborious procedure. Hence, if one wants to install such a network at short notice, one should have a calibration procedure for the cameras that is fast and preferably automatic.

For example, an array of four fixed omni cameras are used in [7, 13] to perform 3D tracking of persons in an indoor setting. Figure 1 (a) shows the omni images acquired from four cameras. This system detects objects using background subtraction as shown in Figure 1 (b). Using the known location of cameras, it obtains the location and heights of the targets by triangulation as shown in Figure 1 (c). We intend to use this system for reconfigurable array of omni cameras such as the one shown in Figure 1 (d), for which a fast calibration procedure is necessary.

1.1 Related Work

Calibration can be performed by registering corresponding features between a number of cameras. However, when the cameras are widely separated, finding corresponding features is a difficult task. Due to this reason, researchers have proposed the use of moving features to register the images [8]. In [9], the calibration parameters of a single rectilinear camera were determined by observing motion of a person in the scene.

Factorization methods have been widely used for structure from motion problem where images from a single moving camera is used to determine the scene structure. Tomasi and Kanade [16] proposed the factorization method for orthographic cameras, which was extended to para-perspective

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
VSSN '04, New York, New York USA
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

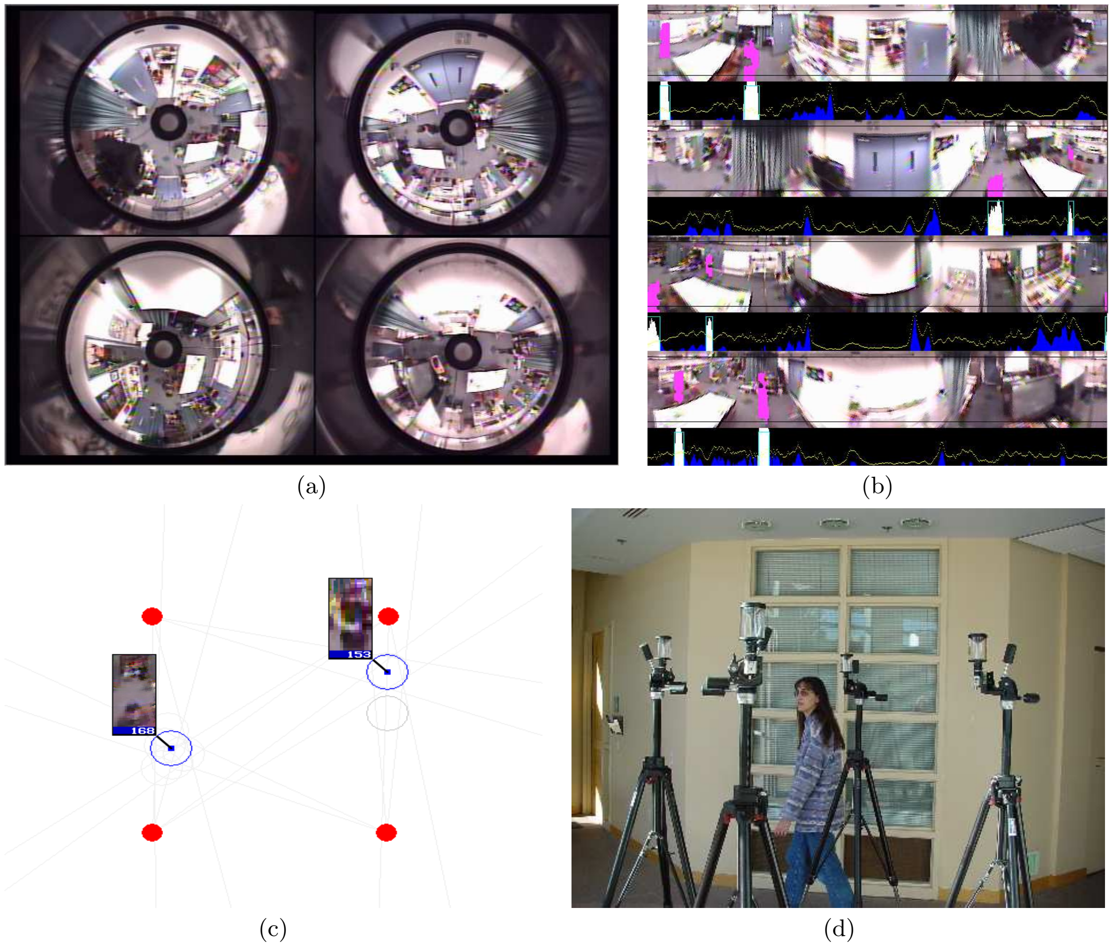


Figure 1: (a) Images from the omni cameras with two moving persons. (b) Panoramic views showing object detection (c) Estimated locations and heights of the detected persons. (d) Proposed reconfigurable array of omni cameras.

cameras [12] as well as perspective cameras [5, 14]. Svoboda [15] have developed a multi-camera calibration software that uses a modified laser pointer to get corresponding points in multiple cameras. Self-calibration is performed using factorization algorithm by Martinec [10] which accounts for outliers and occlusions. Chen et al. [4] as well as Baretto et al. [2] use the path of a moving point to calibrate a wide area network of cameras, all of which cannot see the point at the same time. To introduce robustness to the calibration procedure, a planar target pattern is used by Pedersini [11]. The target is moved around to get multiple views in the area of interest, which gives enough constraints to solve the calibration problem robustly.

This work deals with use of a one-dimensional target, such as a moving person to calibrate multiple stationary omnidirectional cameras. Using the fact that the person is oriented vertically, the non-linear 3-D structure from motion problem is projected to a 2-D problem in the ground plane perpendicular to the person's axis. This problem is solved using factorization method to give an initial approximate solution. The solution is then used in a non-linear optimization framework to account for the non-vertical camera axes, and minimizing a geometric error measure between the observed and the projected omni pixel coordinates.

2. THREE-STAGE CALIBRATION ALGORITHM FOR RECONFIGURABLE OMNI ARRAYS

The omni cameras consist of a hyperbolic mirror and a camera placed on its axis [3]. These cameras have a single viewpoint that permits the image to be suitably transformed to obtain perspective views. Each camera covers a 360 degrees field of view around its center. However, the images are distorted with straight lines transformed into curves. This distortion should be taken into account when performing calibration. The conversion between the homogenous perspective coordinates $\lambda(x, y, z)^T$ and the omni coordinates $(u, v)^T$ is given by:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \equiv K \begin{pmatrix} x \\ y \\ -c_1 z + c_2\sqrt{x^2 + y^2 + z^2} \end{pmatrix} \qquad (1)$$

where c_1 and c_2 are determined by the omni mirror geometry as well as the focal length of the lens, whereas K is the normalized calibration matrix of the CCD, with r being the aspect ratio, s the skew, and (t_u, t_v) the camera center.

$$K = \begin{pmatrix} r & s & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The flow chart of the calibration method is shown in Figure 2. Initially, it is assumed that the camera axes as well as the person are nearly vertical. Using this approximation, the non-linear 3-D problem of structure from motion is reduced to a 2-D problem in plan view coordinates. Using the plan view coordinates of person's position in all cameras, the world coordinates as well as the camera orientation and location is obtained using factorization method. The solution obtained is then used in a non-linear least squares optimization framework, minimizing the geometric error measure between the observed and projected pixels in the omni camera domain, and also incorporating the possibility of camera axes not being exactly vertical.

Let R_i, d_i be the rotation and translation parameters of cameras $i = 1 \ldots M$, and P_j be the world coordinates of a reference point on the object in image frame j. Let $k = 1 \ldots K$ points on the reference objects be at the heights $H_1, H_2 \ldots H_K$ from the reference point in the world coordinate system. For example, if the object is a person with height ΔH, one can have $K = 2$ points with $H_1 = (0, 0, 0)$ and $H_2 = (0, 0, \Delta H)$ representing the top and bottom of the person. If the height is unknown, one can take $\Delta H = 1$ and determine the calibration within a scale factor.

The world coordinates of these points would be $P_{jk} = P_j + H_k$ and the coordinates p_{ijk} in the camera i system are then given by:

$$p_{ijk} = R_i^T(P_j + H_k - D_i) \tag{3}$$

Let the omni image coordinates of the respective point using equation 1 be denoted by $q_{ijk} = (u_{ijk}, v_{ijk})^T$.

2.1 Reduction to plan view

Since this step is applied separately to each camera i and image j, let the subscripts i, j denoting the camera and image be temporarily dropped for the sake of brevity. The omni image coordinates q_k of each object point k is converted to homogenous coordinates $\hat{p}_k$ by inverting equation (1). The coordinates of the points in the camera system are then given by

$$p_k = \lambda_k \hat{p}_k = R^T(P + H_k - D) \tag{4}$$

Let the rotation matrix R be decomposed as $R = R_z R_{xy}$ with R_z being rotation about Z axis and R_{xy} containing rotations about X and Y axes. Let $h_k = R^T H_k = R_{xy}^T R_z^T H_k$. If the target is a linear object perpendicular to the X-Y plane, then the rotation around Z axis does not affect its coordinates, hence $R_z^T H_k = H_k$ and:

$$h_k = R_{xy}^T H_k \tag{5}$$

If the camera axis is nearly vertical then $R_{xy} \approx I$ and $h_k \approx H_k$. Otherwise, R_{xy} can be estimated using vanishing points as described later in this section, and in [9].

If $K = 2$ as in case of using top and bottom points of a moving person, one can subtract the equation for $k = 1$ from that for $k = 2$ to get:

$$\hat{p}_2 \lambda_2 - \hat{p}_1 \lambda_1 = R^T(H_2 - H_1) = h_2 - h_1 \tag{6}$$

This gives 3 equations in 2 variables which can be solved using least squares to obtain λ_1 and λ_2.

In case of $K > 2$ the mean of equation (4) is taken over $k = 1 \ldots K$:

$$\frac{1}{K} \sum_k \lambda_k \hat{p}_k = R^T(P + \frac{1}{K} \sum_k H_k - D) \tag{7}$$

Subtracting this from equation (4), for each k we get:

$$p_k - \frac{\sum_k \lambda_k \hat{p}_k}{K} = R^T(H_k - \frac{1}{K} \sum_k H_k) = h_k - \frac{1}{K} \sum_k h_k \tag{8}$$

Writing these equations in terms of unknown parameters λ_k gives:

$$G \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_K \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{pmatrix} - \frac{1}{K} \sum_k h_k \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{9}$$

$$G = \begin{pmatrix} \hat{p}_1(1 - 1/K) & -\hat{p}_2/K & \ldots & -\hat{p}_K/K \\ -\hat{p}_1/K & \hat{p}_2(1 - 1/K) & \ldots & -\hat{p}_K/K \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{p}_2/K & -\hat{p}_2/K & \ldots & \hat{p}_K(1 - 1/K) \end{pmatrix} \tag{10}$$

is a $3K \times K$ matrix with $3(K - 1)$ independent equations and K variables.

The depths λ_k are obtained using least squares. Using these values, the coordinates p_k in the camera systems are obtained. Since the line joining the p_k's is assumed vertical, the plan view coordinates of the person w_{ij} are obtained by computing the average projection of p_{ijk} into the X-Y plane, and the height d_i of the camera i above the ground is obtained by averaging the Z-coordinates over all j and k as:

$$w_{ij} = \frac{1}{K} R_{xy} \sum_k p_{ijk} \tag{11}$$

$$d_i = \frac{1}{NK} \sum_j \sum_k [(p_z)_{ijk} - H_k] \tag{12}$$

If the camera axes are nearly vertical, one can use $R_{xy} = I$. However, if that is not the case, one can estimate R_{xy} using vanishing points as in [9]. If $K = 2$, the line joining p_1 and p_2 is given by $l = p_1 \times p_2$. In case of $K > 2$ the line can be estimated using singular value decomposition (SVD) as the singular vector corresponding to the smallest singular value of the matrix:

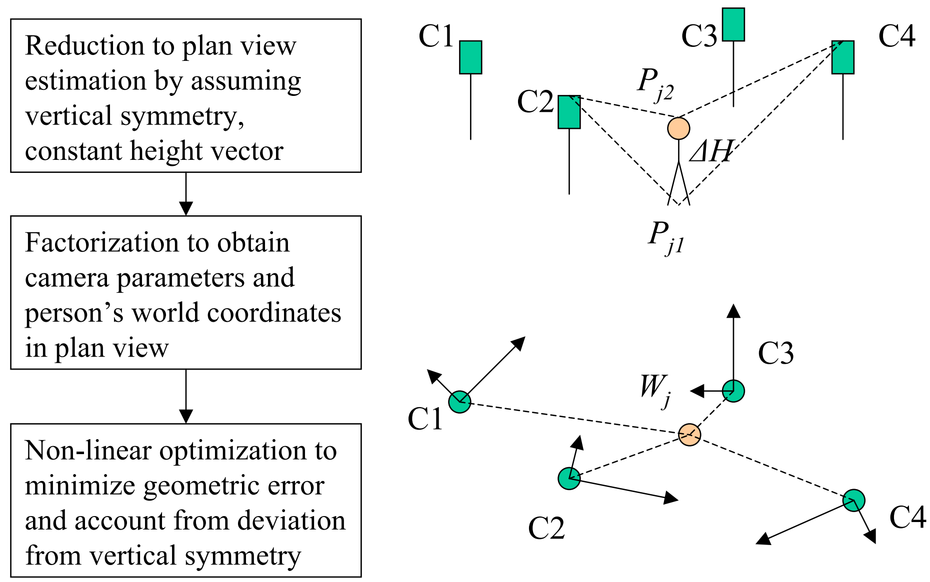
$$\begin{pmatrix} p_1 & p_2 & \ldots & p_K \end{pmatrix} \tag{13}$$

Note that if the elements of p_K are imbalanced, Hartley normalization [6] should be used. Alternatively, robust methods such as RANSAC [5] could be used to fit the line if the data is prone to outliers, and the SVD used after the valid points are determined.

Once the lines l_{ijk} corresponding to object are found for all images $j = 1 \ldots N$ of a particular camera i, the vanishing point v_i of these lines for camera i is found similarly, using SVD of:

$$\begin{pmatrix} l_{i1} & l_{i2} & \ldots & l_{iK} \end{pmatrix} \tag{14}$$

Again, RANSAC could be used for robust fitting, followed by SVD on inlier points.



(a) (b)

Figure 2: (a) Flow chart for the three stage omni-array calibration algorithm. (b) Projection of 3D configuration to plan view.

Since the vanishing point is the image of the point at infinity in direction $(0, 0, 1)$, it corresponds to the 3^{rd} column of the matrix R^T (or R_{xy}^T). The other columns can be specified using orthonormalization, upto the ambiguity of rotation around Z axis.

2.2 Factorization in plan view coordinates

Using the approximately known plan view coordinates of the person in camera system of all cameras, the camera positions and rotations, as well as the world coordinates of the points are obtained using factorization method similar to [16].

Let the relation between the plan view coordinates in the world and camera system be given by $w_{ij} = A_i W_j + b_i$ where A_i, b_i represent the affine transformation due to the camera. Assuming the world origin to be at the centroid of W_j, we have:

$$\begin{aligned} b_i &= \sum_j w_{ij} \\ \tilde{w}_{ij} &= w_{ij} - b_i = A_i W_j \end{aligned} \tag{15}$$

If $\tilde{w}_{ij}$, A_i, and W_j are stacked as matrices with i along rows and j along columns we get a matrix equation:

$$\Omega_{2M\times N} = A_{2M\times 2} W_{2\times N} \tag{16}$$

Hence, Ω is a rank 2 matrix that can be factorized into the camera parameter component A and the world coordinate component W using SVD as:

$$\begin{aligned} \Omega &= USV^T \\ \tilde{A} &= U \\ \tilde{W} &= S_W V^T \end{aligned} \tag{17}$$

where S_W is obtained by setting all but the largest two singular values of S to zero. However, the components $\tilde{A}$ and $\tilde{W}$ are obtained modulo affine transformation Q so that any $A = \tilde{A}Q$ and $Q^{-1}W = \tilde{W}$ would satisfy the factorization. To obtain A so that its 2×2 sub-matrices A_i are orthonormal, one can write as in [5, 12]:

$$A_i^\alpha C A_i^\beta = \delta_{\alpha\beta} \tag{18}$$

where $C = QQ^T$, A_i^α is the α^{th} row of submatrix A_i, and $\delta(\cdot)$ is the Kroneker delta function. These equations are linear in terms of elements of the symmetric matrix C, and are solved using least squares. The matrix Q is obtained using eigenvalue decomposition of C. The signs of columns of Q are adjusted to ensure that the resulting matrices $A_i = \tilde{A}_i Q$ have positive determinant.

2.3 Non-linear optimization

The factorization method gives an approximate solution to the calibration. However, it does not account for the pan and tilt of the camera, nor does it minimize a geometrically significant measure of error. To account for these, a non-linear least squares procedure is applied by expanding the 2-D solution to 3-D and using that as the initial guess:

$$R_i = \begin{pmatrix} A_i^T & 0 \\ 0 & 1 \end{pmatrix} R_{xy},\ D_i = -R_i^T \begin{pmatrix} b_i \\ d_i \end{pmatrix},\ P_j = \begin{pmatrix} W_j \\ 0 \end{pmatrix} \tag{19}$$

The rotation matrix R is parameterized into angle axis form using Rodrigues formula. The objective function to be minimized is given by the sum of squares of the error between the actual and the projected omni coordinates:

$$\min_{d_i, \theta_i, P_j} \sum_{i,j,k} \|q_{ijk} - \hat{q}_{ijk}\|^2 \tag{20}$$

with $\hat{q}_{ijk}$ being the projections obtained using the current estimates of d_i, θ_i, P_j. To minimize the number of variables in optimization, one can split the procedure into two parts and iterate until convergence:

1. Fix d_i, θ_i for all i and for each j separately perform:

$$\min_{P_j} \sum_{i,k} \|q_{ij} - \hat{q}_{ij}\|^2 \tag{21}$$

2. Fix P_j for all j and for each i separately perform:

$$\min_{d_i,\theta_i} \sum_{j,k} \|q_{ij} - \hat{q}_{ij}\|^2 \tag{22}$$

To have a unique solution, the first camera is fixed at world origin, and the second camera is along positive world X axis by constraining the respective parameters to zero.

3. EXPERIMENTAL EVALUATION AND RESULTS

The approach described above was tested with simulated as well as real data. For simulations, two configurations of 4 omni cameras and a number of point pairs corresponding to a walking person were manually generated. A small rotation around the X and Y axes was performed on the cameras. Figure 3 (a) and (c) shows the plan views of the two configurations. The 3-D points were projected on the omni cameras. A number of simulations (100) were carried out by adding random noise ($\sigma = 3$ pixels). The camera position, orientation, and the locations of the points were estimated and plotted for all simulations as well as ground truth. It is seen that the estimated positions are quite close to the ground truth positions in all the simulations. Figure 3 (b) and (d) shows the statistics of the camera parameter estimation for the configurations.

In order to use the calibration algorithm in conjunction with person tracking and event analysis applications [7], a reconfigurable array of 4 omni cameras were placed in a hallway as shown in Figure 1. Figure 4 (a) shows a typical snapshot of the moving person in all 4 cameras. To detect the moving person, a background image generated using several frames of the scene was subtracted from the images and the largest blob in each camera was extracted as shown in Figure 4 (b). The bottom and top of the blobs were extracted. Frames in which their positions were correct were manually selected and given to the estimation algorithm. The estimated position and orientation of the cameras, as well as the positions of the tracked points are shown in Figure 4 (c). Ground truth camera positions are also shown for comparison.

Another set of experiments was performed in outdoor setting. However, due to poor lighting conditions, the positions of the person were marked manually in the images. Figure 5 shows the results of this experiment. The error was considerably greater in this case. However, it was observed that most of the error was in the scale of estimation, and the shapes of the estimated and ground truth positions matched better if scaled by a uniform factor. Possible reasons for this discrepancy could be greater sensitivity in estimation of scale than of orientation, a bias in the marking of person's height, or inaccuracies in internal parameters.

4. CONCLUDING REMARKS

The paper described a method for calibrating reconfigurable arrays of omnidirectional cameras using the image positions of a moving person in all the cameras. Factorization method followed by non-linear optimization was used to determine the camera parameters as well as the position of moving person in multiple frames. Experimental results using simulated and real data show that the method is effective in getting a calibration within a reasonable accuracy.

If greater accuracy is required, one can use poles standing vertically with distinctive marks at K different heights. We are working on applying the algorithm in that scenario and evaluate the accuracy obtained using this method. We are also working on integrating calibration of rectilinear cameras along with omni cameras. For rectilinear cameras, the camera axis is likely to deviate from vertical, and vanishing point method described above and in [9] would be useful. For handling missing points and outliers, the approaches used by Martinez et al. [10] would be useful in factorization. Currently, the internal calibration of omni camera is predetermined using known points on the ground. However, we would like to study if it is feasible to estimate the internal camera parameters simultaneously with sufficient accuracy. Finally, we plan to integrate the calibration with person tracking and event analysis systems shown in Figure 1 for real world deployment of sensor networks containing omni as well as rectilinear cameras.

Acknowledgements

We are thankful for the grant awarded by the Technical Support Working Group (TSWG) of the US Department of Defense which provided the primary sponsorship of the reported research. We also thank our colleagues from the UCSD Computer Vision and Research Laboratory for their contributions and support.

5. REFERENCES

[1] S. Baker and S. K. Nayar. A theory of catadioptric image formation. In *ICCV98*, pages 35–42, 1998.

[2] J. Barreto and K. Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In *Proc. 5th Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras*, 2004.

[3] R. Benosman and S. B. Kang. *Panoramic Vision: Sensors, Theory, and Applications*. Springer, 2001.

[4] X. Chen, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2520–2527.

[5] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. 2003.

[6] R. I. Hartley. In defence of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6).

[7] K. C. Huang and M. M. Trivedi. Video arrays for real-time tracking of persons, head and face in an intelligent room. *Machine Vision and Applications*, 14(2):103–111, 2003.

[8] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *PAMI*, 22(8):758–767, August 2000.

[9] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *ICPR02*, pages I: 562–567, 2002.

[10] D. Martinec and T. Pajdla. Automatic factorization-based reconstruction from perspective

images with occlusions and outliers. In *Proc. 8th Computer Vision Winter Workshop (CVWW)*, pages 147–152. Czech Pattern Recognition Society, Prague, February.

[11] F. Pedersini, A. Sarti, and S. Tubaro. Accurate and simple geometric calibration of multi-camera systems. *Signal Processing*, 77:309–334, 1999.

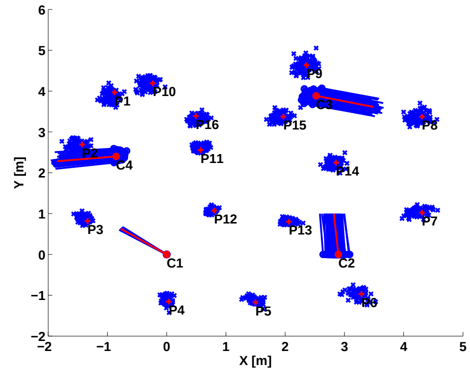
[12] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3):206–218, March 1997.

[13] T. Sogo, H. Ishiguro, and M. Trivedi. N-ocular stereo for real-time human tracking. In R. Benosman and S. Kang, editors, *Panoramic Vision: Sensors, Theory, and Applications*, pages 309–396. Springer, 2001.

[14] P. Strum and B. Triggs. A factorization-based algorithm for milti-image projective structure and motion. In *Proc. European Conference on Computer Vision*, pages 709–720, 1996.

[15] T. Svoboda. Quick guide to multi-camera self calibration. Technical Report BiWi-TR-263 Version 0.2, Swiss Federal Institute of Technology, Zurich, Switzerland, August 2003.

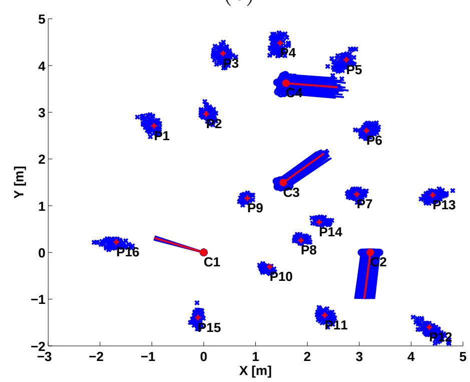
[16] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.



(a)

		Actual	Mean	RMS err		Actual	Mean	RMS err
			Camera 1				Camera 2	
Position	d_x	0.000	fixed	fixed		3.213	3.206	0.079
[m]	d_y	0.000	fixed	fixed		0.000	fixed	fixed
	d_z	1.840	1.849	0.031		1.840	1.865	0.036
Orientatior	theta_x	-0.015	-0.010	0.018		-0.003	-0.004	0.014
[radians]	theta_y	0.009	0.005	0.016		-0.016	-0.009	0.014
	theta_z	2.828	2.828	0.014		-1.689	-1.689	0.015
			Camera 3				Camera 4	
Position	d_x	1.534	1.526	0.059		1.587	1.565	0.075
[m]	d_y	1.494	1.482	0.046		3.622	3.607	0.088
	d_z	-1.800	1.812	0.022		-1.836	-1.846	0.029
Orientatior	theta_x	-0.004	-0.001	0.012		-0.023	-0.023	0.010
[radians]	theta_y	-0.001	0.001	0.012		-0.014	-0.009	0.014
	theta_z	0.671	0.674	0.017		-0.088	-0.084	0.019

(b)



(c)

		Actual	Mean	RMS err		Actual	Mean	RMS err
			Camera 1				Camera 2	
Position	d_x	0.000	fixed	fixed		3.213	3.206	0.079
[m]	d_y	0.000	fixed	fixed		0.000	fixed	fixed
	d_z	1.840	1.849	0.031		1.840	1.865	0.036
Orientatior	theta_x	-0.015	-0.010	0.018		-0.003	-0.004	0.014
[radians]	theta_y	0.009	0.005	0.016		-0.016	-0.009	0.014
	theta_z	2.828	2.828	0.014		-1.689	-1.689	0.015
			Camera 3				Camera 4	
Position	d_x	1.534	1.526	0.059		1.587	1.565	0.075
[m]	d_y	1.494	1.482	0.046		3.622	3.607	0.088
	d_z	-1.800	1.812	0.022		-1.836	-1.846	0.029
Orientatior	theta_x	-0.004	-0.001	0.012		-0.023	-0.023	0.010
[radians]	theta_y	-0.001	0.001	0.012		-0.014	-0.009	0.014
	theta_z	0.671	0.674	0.017		-0.088	-0.084	0.019

(d)

○ Camera position − Camera orientation + Point position

Figure 3: (a) and (c) Two simulated camera configurations showing the camera locations, orientations, and track points. Ground truth as well as estimated values from simulations are shown. (b) and (d) Statistics of camera parameter estimation. Note that camera 1 is fixed at the world origin, and the camera 2 along X axis.

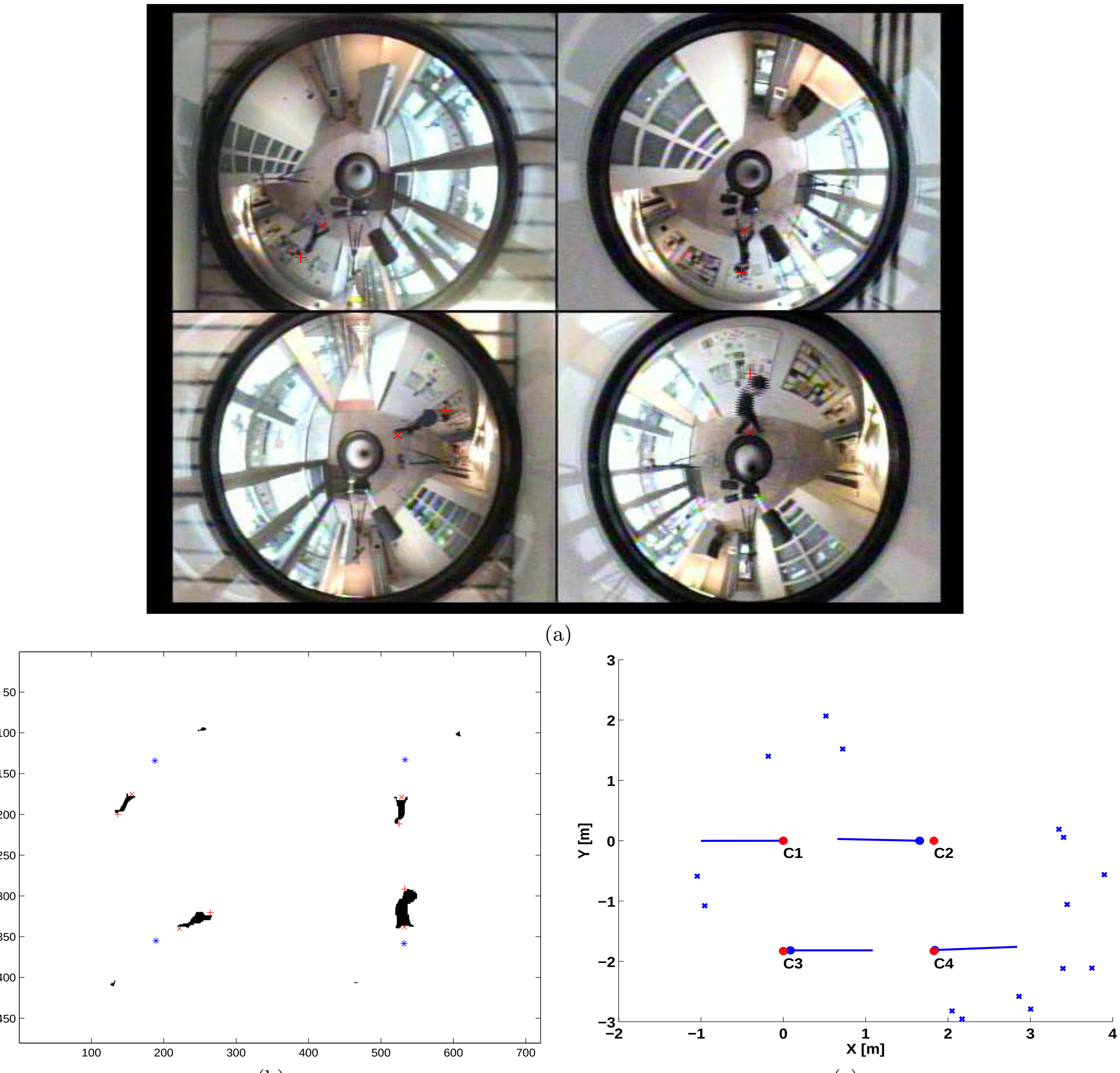
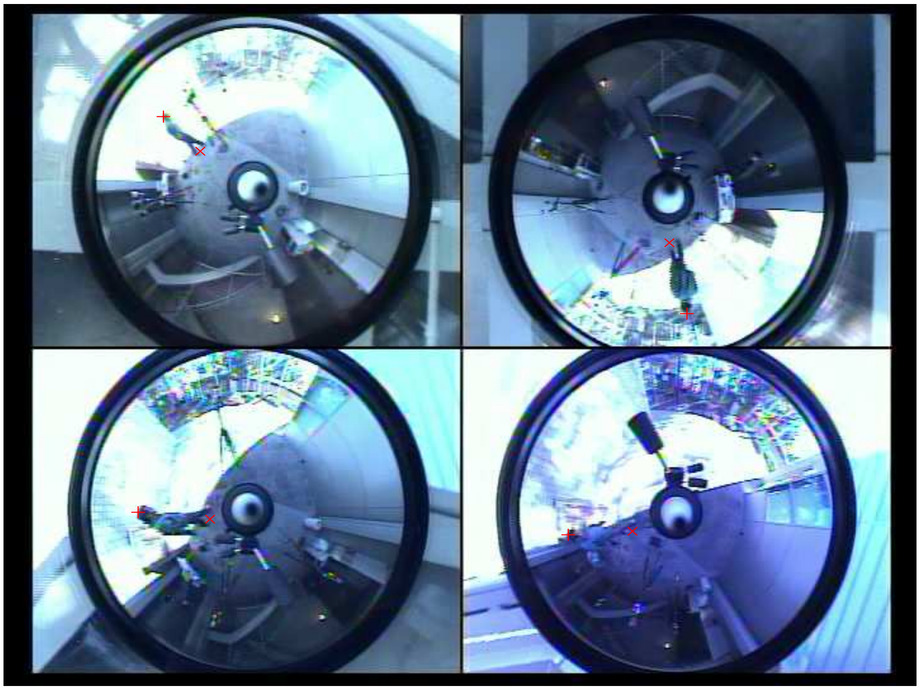
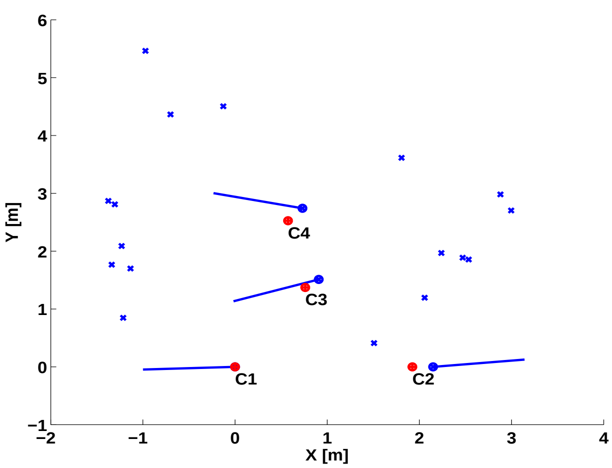


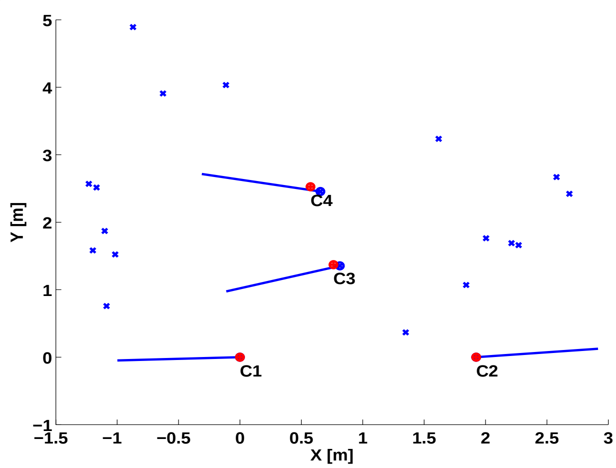
Figure 4: (a) Images from a reconfigurable array of 4 omni cameras assembled in a hallway. (b) A person moving around is detected in all 4 cameras by background image subtraction. The positions of the top and bottom of the person in this frame are extracted. (c) Result of estimation algorithm (blue) and ground truth (red) using the person's position in this and other frames.



(a)



(b)



(c)

Figure 5: (a) A network of 4 omni cameras assembled in an outdoor setup with a different camera geometry. The positions of the person were manually marked. (b) Result of estimation algorithm (blue) as well as the ground truth (red). (c) Result after scaling the estimated positions to align the cameras 1 and 2. Alignment for cameras 3 and 4 is much better now.