Automatic Calibration of Camera Networks based on Local Motion Features

Cem Taylan Aslan, Kai Bernardin, Rainer Stiefelhagen

Automatic Calibration of Camera Networks based on Local Motion Features

Cem Taylan Aslan, Keni Bernardin, Rainer Stiefelhagen

Institut für Theoretische Informatik, Universität Karlsruhe
76131 Karlsruhe, Germany

Abstract. This paper introduces a new technique to automatically calibrate the extrinsic parameters of a network of cameras without using dedicated calibration objects or markers. Instead, the motion of persons walking naturally through a scene is used. Simple foreground and motion features are extracted from the individual image sequences. A Hough transform is applied in a specially defined parameter space to estimate the relative geometry between camera pairs without solving the problem of finding correct feature correspondences between views. All possible feature correspondences are examined, and a modified gradient descent algorithm is used to find the set of optimum calibration parameters within the Hough space. After calibrating each camera pair the resulting camera network is built up using a global error minimization technique. The approach is tested on several indoor scenarios and shows a high degree of robustness, especially when multiple persons enter the scene, making it difficult to resolve feature correspondences. The correctness and precision of individual camera calibrations and of the resulting camera network are thoroughly evaluated, showing that triangulation errors as low as 5cm can be reached using very little observation data.

1 Introduction and Related Work

Camera calibration can be an invaluable tool for visual analysis and scene understanding. On the one hand, it allows to make inferences about the real-world size of observed objects, their distance to the camera, or their 3D scene location, on the other hand, it may be used to efficiently correlate the information from several visual sources. One application is e.g. object tracking using a network of cameras, a field which is gaining more and more attention with the steady increase in availability of cheap high quality sensors. In common, the calibration information consists of two types of parameters: The internal parameters which describe optical properties such as the focal length, principal point, skew and distortions coefficients that arise from the camera lens, and the external parameters, describing the position and orientation of the camera in the scene. In this paper, we assume that the internal and distortion parameters are known. Due to their static nature, they usually need to be determined only once, whereas the external parameters are more prone to change and may in some cases be difficult to reliably estimate, for lack of scene knowledge.

Calibration of sensor networks has been a topic in the computer vision domain for quite some time. In general, the techniques can be grouped into two categories: The traditional way to determine extrinsic parameters consists in finding those which best map picture coordinates to corresponding known 3D world coordinates [1–3]. The main disadvantage is that these techniques need a well known calibration object with known world coordinates. The second category comprises techniques which do not require a predefined calibration object, but instead attempt to automatically extract useful features from different views of a scene, thus building a virtual calibration object "on-the-fly" . This can be done using marker-objects [4–7] or by trying to find corresponding features in natural images [8–10].

The here presented approach is designed to determine the extrinsic parameters of a network of cameras without direct human interaction, but rather by simply observing the movement of people in the scene. It assumes a relatively high amount of overlap between views and basic synchronization of all cameras. We will show that this approach is robust to errors in the feature extraction process and can notably determine the correct extrinsic parameters without resolving feature correspondences between views.

The remainder of this paper is organized as follows: Section 2 describes the feature extraction step. A detailed presentation of the developed algorithm for calibrating camera pairs is given in Section 3, followed by a brief explanation on the build-up of the camera network in Section 4. In Section 5 the accuracy of the approach is evaluated. Finally, Section 6 gives a summary and a conclusion.

2 Feature Extraction

The automatic calibration technique described in this paper is designed to operate on very general and simple features that could be reliably extracted in a wide range of natural scenarios. Likewise, the feature extraction itself is quite simple and will only briefly be described in the following. It relies on the assumption that motion occuring concurrently in two camera views is likely to belong to the same object or objects. It makes some general assumptions that hold for most visual surveillance systems. These assumptions, however, only pertain to the feature extraction step, related to our application scenario, and in principle do not limit the calibration procedure itself:

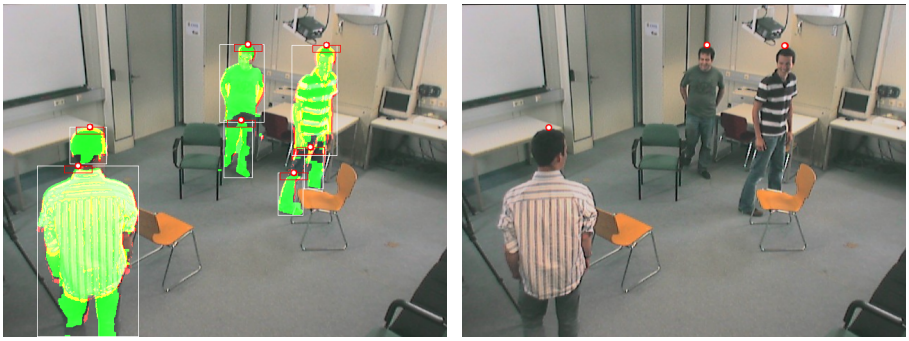
- Mostly persons are monitored.
- The cameras are mounted upright.
- The cameras are mounted above person height to offer a better view point.

Exploiting these assumptions, the highest detected point of each moving person is used as calibration feature. The main benefit is that this feature is in general less affected by occlusions. A similiar approach is used in [8].

At first, a foreground segmentation is made in each view using a background model with a low adaption rate ($\alpha = 0.002$) and a fixed threshold. The resulting support map is morphologically filtered and only blobs with a minimum size

are considered. The highest point of each detected foreground blob is then used as initial hypothesis. Next, the objects moving in the current time frame are determined. A support map of areas containing motion is created (in our case, simple difference images are used), and hypotheses are rejected if there is not sufficient motion within a 40x10 pixel window centered around them.

The reason the hypothesis points are determined on the foreground support and not directly on the difference images is that they provide much smoother contours, whereas local motion is only reliably detectable in image areas with a strong gradient in the direction of movement. On the other hand, foreground support alone is not sufficient, as it provides no information on which objects are actually moving at one point in time. The combination of techniques alleviates their respective drawbacks. In a final step, hypotheses for which the support blob lies beneath another support blob are discarded. This is to eliminate most of the false positives that arise, e.g., when a person blob is fragmented due to faulty segmentation. Of course, this also means some hypotheses may be wrongfully rejected in case of overlap or when blobs from different persons appear above each other. In our case, however, it is not necessary to correctly extract features for every person in every frame, as the subsequent calibration algorithm is robust to such preprocessing failures.



(a) **(b)**

Fig. 1. Feature extraction. (a) Extracted foreground blobs (green) with bounding boxes (white) and feature hypotheses (red/white points). Motion areas are overlayed onto the image (red/yellow). (b) Result of the feature extraction step.

Figure 1(a) shows an example frame taken from a test scenario with three persons. The foreground is colored green and the hypothesized features are marked with red/white dots. As can be seen, blob fragmentation leads to the initial extraction of seven hypotheses. Verifying for motion (red/yellow overlay) and vertical misalignment discards most false positives (see Figure 1(b)). The (x,y) coordinates of the remaining hypotheses are taken as features.

In our test scenarios, the videos are captured with 15fps. Since the motion of a person between two consecutive frames is too small to yield a meaningful gain in information, only the features from every fifth frame are used for calibration. This reduces the processing time for a sequence without much affecting the results.

3 Pairwise Calibration using a Hough Transform

The calibration of the camera network is achieved by first estimating the relative position and orientation of each camera pair, and then combining the pairwise results in a second step. One should note that as the extracted features allow no inference about the size of the observed scene (the algorithm cannot tell if it is observing life-size humans in a room or just miniature puppets in a cardboard set), the network geometry can only be estimated up to an overall scale factor. The following describes the algorithm for pairwise calibration.

The basic idea underlying our approach is the fact that when a same object is observed in two cameras views, the LOVs (Lines of View) from both cameras intersect in one point, the object position, as long as the extrinsic parameters have been correctly estimated. In practice they may not intersect exactly due to imprecisions in the cameras and the feature extraction, and instead we only require the triangulation error, the distance between LOVs, to be small. In other words, we are searching for the extrinsic camera parameters which minimize the total triangulation error for all hypothesized feature correspondences. A priori, this seems to imply a search in a six dimensional search space (three parameters for translation and three parameters for rotation). We will, however, show how the search complexity can be reduced.

Consider a pair of cameras C_1 and C_2. We wish to obtain the position and orientation of C_2 in C_1's coordinate frame. First, let us consider the translation vector $\mathcal{T} = (x, y, z)^T$ from C_1 to C_2. Without loss of generality we can transform $\mathcal{T}$ into spherical coordinates $\mathcal{T}' = (r, \theta, \varphi)^T$. As no real-world size information can be gained from the extracted point features, we can only estimate the direction of translation, not its scale [11] . Therefore, we define $\mathcal{T}$ to be of unit length: $|\mathcal{T}| = 1$ and $\mathcal{T}' = (1, \theta, \varphi)^T$, leaving only two free parameters for the translation.

The rotation is specified by three angles: the pan (α), tilt (β) and roll (γ) of C_2 relative to C_1. The computational cost for estimating all 5 parameters jointly would be extremely high, which is why we decompose the problem into two parts: First, we show how the pan and tilt angles can be computed from hypothesized feature correspondences if the translation vector $\mathcal{T}'$ and the roll angle γ are known. Then we define a search in the 3-dimensional $(\theta, \varphi, \gamma)$ space to find the optimum parameter combination.

As stated above, when observing one same object at point X from two different views, the LOVs from the cameras to the object should intersect. Let C_1's coordinate frame be the reference frame, and let C_2 be positioned at $\mathcal{T}'$ and rotated around its Z-Axis by the angle γ. Let lov_1 and lov_2 be the LOVs from

C_1 and C_2, with c_1 and c_2 the respective focal points. If the distance from c_1 to X were known, we could directly compute the α and β values for C_2. But in our case this information is not available, and X could lie anywhere on lov_1, resulting in a 1-dimensional solution space. The set of all solutions can be obtained by sampling every point on lov_1 and calculating the corresponding angular values for lov_2. The computation is made efficiently by exploiting two geometric constraints: First, every line connecting c_2 to a point on lov_1 is contained in the epipolar plane π, which is spanned by the two camera focal points and lov_1. Second, the point X can only lie between c_1 and infinity. For both extremes, lov_2's orientation is known (in the latter case it is parallel to lov_1). Figure 2(b) shows the two extreme solutions for lov_2. All other solutions lie in between.

We now construct a line segment L_{join} in the following way:

$$L_{\text{join}} = s \cdot (l_1 + \mathcal{T}), \forall s \in \mathbb{R} : 0 \leq s \leq 1 \tag{1}$$

with l_1 the normalized direction vector for lov_1. The set of lines connecting c_2 to all points on L_{join} is the same as the set connecting c_2 to all points on lov_1 that lie between c_1 and infinity. Thus, we can efficiently sample the solution space by sampling points on L_{join} and calculating the corresponding α and β angles for lov_2.

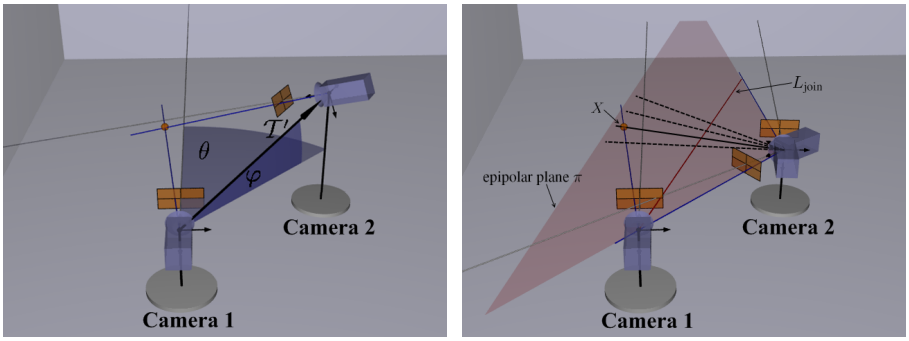


Fig. 2. Pairwise extrinsic calibration. (a) The angles θ and φ corresponding to translation $\mathcal{T}'$. (b) The solution space for lov_2 (red overlay) in Camera 1's reference frame, given an observation X and fixed translation and roll values for Camera 2. L_{join} is represented by the red line segment on the epipolar plane π.

Let us now consider the case that two objects are viewed from two cameras. As the correct feature mapping between views is not known, there are 4 possible correspondences: 2 correct and 2 false, which will all be used in the parameter estimation. Figure 3 (a) shows the solution set for one correct correspondence. Calculating the pan and tilt angles for the second correct correspondence results in another solution set. The correct overall solution can be easily identified

as the intersection of both sets, as shown in Figure 3 (b). Next, we calculate the solution sets for the false correspondences. Figure 3 (c) shows the solution sets for the two correct and the two false correspondences. In the present case, the solution sets for the false correspondences have no pan-tilt-combination in common. When computing the hough transform for an entire sequence of observations, the solution sets for all feature correspondences are accumulated for all time frames. In general, it can be expected that the correct pan-tilt-combination will be included in the solution set for every correct correspondence, leading to a high peak, whereas the solutions for false correspondences will be more or less spread over the hough space. This is shown in Figure 3 (d) where 800 correct and 800 false feature correspondences are used as input to the hough transform. The correct pan-tilt-combination is found as the maximum accumulated value (noted by the red circle). This shows how the α and β parameters can be estimated, using the observed image features, if the $\theta \in [-\pi : \pi[\,, \varphi \in \left[-\frac{1}{2}\pi : \frac{1}{2}\pi\right]$ (translation) and $\gamma \in [-\pi : \pi[$ (roll) parameters are known, thus reducing the dimensions of our search space to three.

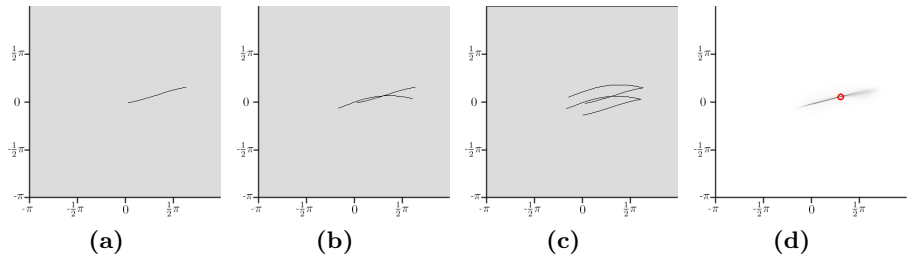


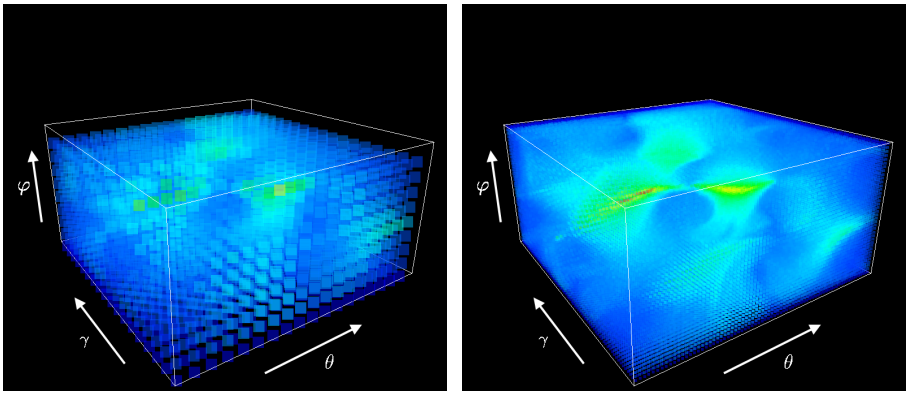
Fig. 3. Example solutions spaces using hough transforms: The X- and Y-axes represent the pan and tilt angles, respectively. (a) one correct feature correspondence (b) two correct correspondences (c) two correct and two false correspondences (d) 1600 correspondences (800 correct, 800 false).

The search for the optimal θ, φ and γ values is made using a gradient descent algorithm. The idea is that if the translation and roll values are not correctly chosen, the LOVs for all correct feature correspondences can not be brought to intersect for any pan/tilt combination. This means that the maximum accumulated value in the corresponding Hough transform will be lower than with correct parameters. This maximal value can therefore be used as quality measure for a given state in the $(\theta, \varphi, \gamma)$ search space. Figures 4(a) and 4(b) show a visualization of the whole search space. As can be seen, there are several local maxima that complicate the search for the global maximum. To find it, a hierarchic gradient descent algorithm with variable step width is applied. To obtain initial starting points for the search, the entire search space is sampled in 10 degree steps and all sample points with values above 90% of the overall maximum are taken. From these initial points, a gradient descent search is started with a step

width of 5 degrees. After termination of the algorithm, the search is restarted from the end points with half the step width, and the procedure is repeated until a precision of 0.05 degrees is reached.

When selecting the starting points for the search, it proved crucial to provide enough margin for error: Taking only the maximum scoring sample points (instead of all points with values above 90% of the maximum), increases the risk of missing the global optimum. This is illustrated in Figure 4(a) which shows an example search space sampled in 20 degree steps. Choosing only the points with maximum value, we would be searching only in the right half of the space, ignoring the high but not maximal values on the left. However, sampling the entire search space in 5 degree steps as shown in Figure 4(b) reveals that the very narrow global maximum is in fact lying on the left.

After the optimum θ, φ and γ values are determined, the already computed Hough transform for these parameters is analyzed to obtain the α and β values and thus the relative orientation. By converting $\mathcal{T}'$ back to $\mathcal{T}$, the translation vector is also obtained.



(a) **(b)**

Fig. 4. Visualization of the search space. Low values are dark blue, high values are bright red. (a) Results for 20° step size. (b) Results for 5° step size.

4 Building the Camera Network

Using the above determined relative orientation and translation for each camera pair, the camera network is built up. Similar to work described in [4] and [6], an iterative approach is taken: Starting from an initial camera pair (with relative distance normalized to 1), additional cameras are successively added by triangulation. As there are several possibilities for the starting camera and the order of inclusion, all possible resulting networks are built up and evaluated using a

global optimization criterion: For each camera pair, the angular difference between the initially hypothesized translation vector (derived from θ and φ) and the actual vector obtained after completion of the network is measured. From all network candidates, the one that minimizes the total angular error for all camera pairs is then chosen.

5 Experimental Evaluation

The presented calibration technique was evaluated on three example scenarios recorded in a room equiped with four static cameras. The evaluation consists of three parts. First, the automatically estimated pairwise calibrations are compared against manually gained ground truth. Second, the quality of the overall network calibration is measured. For this, eight colored markers were placed in the room and their exact world coordinates measured. The markers were then used to measure the average projection errors for each camera and the average triangulation error for the network. To obtain the calibration ground truth for all four cameras, the Camera Calibration Toolbox [12] was used, together with a large checkerboard reference object. Each camera is connected by firewire to a Pentium4 3GHz machine and delivers 640x480 pixel images at 15fps.

5.1 Scenarios

The first scenario consists of a video sequence with 2604 frames (2:54 min) from all four cameras. It figures one person walking continuously through the room in order to generate enough data variability for calibration. In this scenario there are, save for feature extraction errors, no ambiguous correspondences between features from two different views. The second scenario is similar to the first, except that two persons are walking simultaneously. It has a length of 2829 frames (3:09 min). The last scenario figures three persons evolving freely in the scene without specific constraints. They are free to move, stop or sit down as they please. It has a total length of 4051 frames (4:30 min). The main purpose of this scenario is to show that observations of naturally interacting persons are enough to calibrate a camera network, and further that this can be done without finding feature correspondences between views.

5.2 Pairwise Calibration Results

Figures 5(a) to 5(c) show the absolute difference between the calculated pairwise calibrations and the ground truth in degrees. Interestingly, Scenario 3 (the most complex scenario, Figure 5(c)) shows the lowest calibration error although Scenario 1 (Figure 5(a)), showing the highest errors, was no doubt the simplest scenario, with only one moving person. An investigation showed that this can be attributed to the limited amount of observation data in Scenario 1, which was insufficient for precise calibration. Results improve with Scenario 2 (Figure 5(b)), which has almost the same length, but double the amount of features,

due to the fact that two persons are moving. The results demonstrate that our approach can cope with multiple moving objects, though we do not attempt to find feature correspondences. They also show that by applying the hough transform, accuracies increase with the amount of available data (through lengthier observation sequences or more moving persons). For Scenario 3 (Figure 5(c)), most of the errors are below two degrees, and around half are even below one degree.

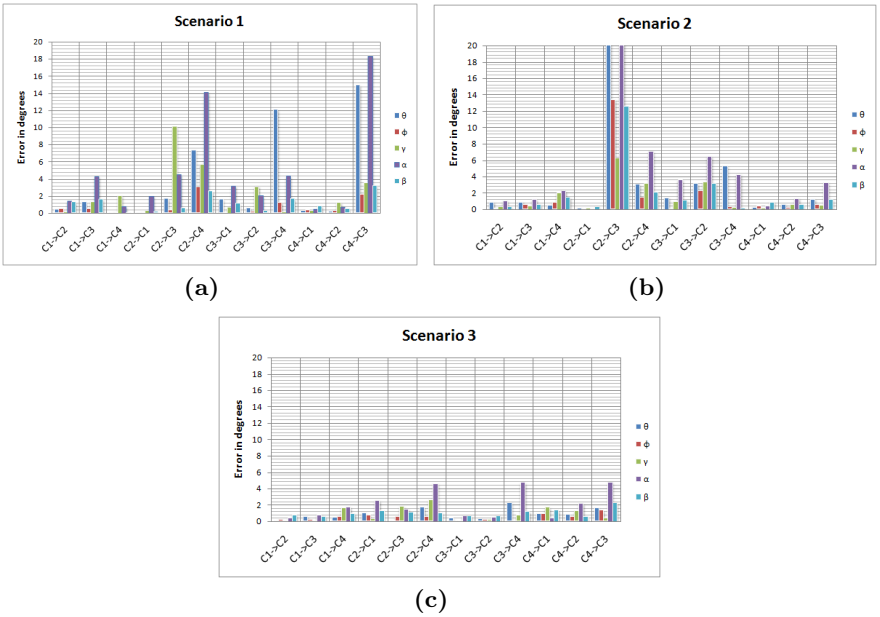


Fig. 5. Pairwise calibration error in degrees. (a) Results for scenario 1. (b) Results for scenario 2. (c) Results for scenario 3.

5.3 Projection and Triangulation Errors

Whereas the previous section analyzes the quality of the pairwise calibration process alone, the projection and triangulation errors measured here show the overall quality of our approach. To evaluate them, the camera network is built up using the pairwise calibrations and transformed to scene coordinates and scale.

The triangulation error is obtained by manually annotating each marker in each view and triangulating the scene coordinates of the markers using the automatically estimated calibration data. The error is defined as the mean error between the ground truth (measured real world coordinates) and the calculated coordinates of all markers.

	mean error	standard deviation
Triangulation	176,71 mm	13,81 mm
Projection Camera 1	10,12 Pixel	3,31 Pixel
Projection Camera 2	1,84 Pixel	0,58 Pixel
Projection Camera 3	81,66 Pixel	18,10 Pixel
Projection Camera 4	5,08 Pixel	1,66 Pixel

Table 1. Triangulation and projection errors for scenario 1.

The projection error is calculated by projecting the known scene coordinates of the markers into each camera view, using the estimated calibrations, and measuring the displacement between the projection and the annotated marker position.

Tables 1 to 3 show the triangulation and projection errors resulting from the built camera networks. In Scenario 1, the triangulation error and Camera 3's projection error are quite high. The schematic view of the resulting camera network (Figure 6(a)) shows a relatively high displacement of Camera 3 compared to its real position. It should be noted that when triangulating the marker positions without using Camera 3, the mean triangulation error decreases to 32,81 mm . It should also be noted that although the pairwise calibrations were relatively inaccurate, the final estimates for cameras 1, 2 and 4 are quite close to the real configurations, resulting in low projection errors. The results for Scenarios 2 and 3 (Tables 2 and 3) show that our approach reaches very low triangulation errors (below 5 cm) and projection errors mostly below 10 pixels, which reflects the good quality of calibration for the whole network (Figure 6(b)). The residual errors stem from the imprecise feature extraction, since the used features in each view (the highest point of a moving person blob) do not necessarily match to precisely the same point in the scene.

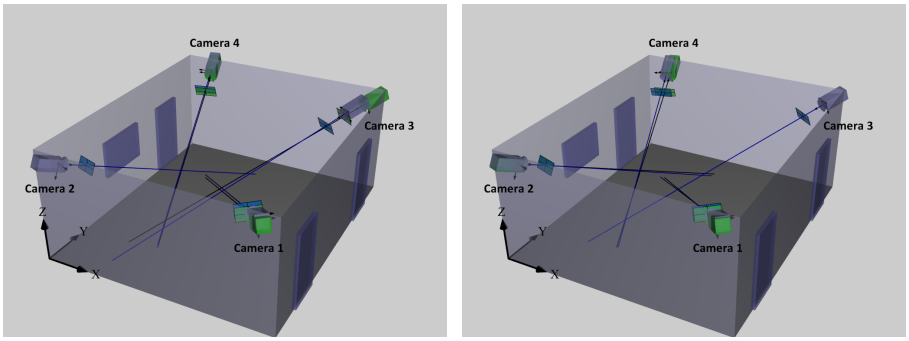
	mean error	standard deviation
Triangulation	44,47 mm	3,16 mm
Projection Camera 1	1,22 Pixel	0,60 Pixel
Projection Camera 2	7,78 Pixel	0,59 Pixel
Projection Camera 3	9,71 Pixel	2,37 Pixel
Projection Camera 4	6,32 Pixel	1,10 Pixel

Table 2. Triangulation and projection errors for scenario 2.

The results also show how the intelligent integration of pairwise calibrations into a network with overlapping views reduces the effects of calibration errors. Although the pairwise results for Scenario 2 are noticeably worse than those for Scenario 3, the triangulation error ($\approx$ 5 cm) and the average projection error ($\approx$ 6 pixels) in both cases are similar.

	mean error	standard deviation
Triangulation	49,98 mm	16,37 mm
Projection Camera 1	11,51 Pixel	1,32 Pixel
Projection Camera 2	6,06 Pixel	0,57 Pixel
Projection Camera 3	2,70 Pixel	0,55 Pixel
Projection Camera 4	5,85 Pixel	4,52 Pixel

Table 3. Triangulation and projection errors for scenario 3.



(a) **(b)**

Fig. 6. Schematic view of the resulting camera network. Green represents the ground truth and blue the system hypothesis. (a): Results for scenario 1. (b): Results for scenario 3.

6 Summary and Conclusion

In this paper, we have presented a statistical technique for the automatic calibration of camera networks using localized motion features. In contrast to other approaches, this technique does not rely on dedicated calibration objects or markers, and does not require manual intervention or a specific calibration procedure. Instead, the motion of persons evolving naturally in the scene is used. Simple foreground and motion features extracted from each view serve as observations for the statistical estimation of extrinsic camera parameters. Using a Hough transform in combination with a hierarchical gradient descent search, the parameters of the pairwise camera geometries are estimated even in the presence of multiple moving objects, without the need to resolve feature correspondences between camera views. The resulting camera network is then built up using a global error minimization technique. Evaluation of the technique in a series of scenarios of increasing complexity revealed its ability to recover the correct network geometry using very little data, even with imprecise, faulty and often ambiguous features. The tests show that average projection errors of 6 pixels are attainable with less than 5 minutes of recordings, and that precision increases with the amount and richness of available data.

Future work could include incorporating direction of motion or size information extracted from image features. Another direction could be to extend the technique to allow for the automatic and fast integration of additional cameras into an already calibrated network.

Acknowledgments

This work has been funded by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoid Robots".

References

1. Tsai, R.Y.: 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 **3** (1987) 323–344
2. Szenberg, F., Carvalho, P.C.P., Gattass, M.: Automatic camera calibration for image sequences of a football match. In Singh, S., Murshed, N.A., Kropatsch, W.G., eds.: ICAPR. Volume 2013 of Lecture Notes in Computer Science., Springer (2001) 301–310
3. Xu, Q., Ye, D., Che, R., Huang, Y.: Accurate camera calibration with new minimizing function. In: Proceedings of the IEEE International Conference on Robotics and Biomimetics, Kunming, China, IEEE (2006) 779–784
4. Chen, X., Davis, J., Slusallek, P.: Wide area camera calibration using virtual calibration objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2000) 520–527
5. Davis, J., Chen, X.: Calibrating pan-tilt cameras in wide-area surveillance networks. In: Ninth IEEE International Conference on Computer Vision (ICCV). Volume 1. (2003)
6. Chippendale, P., Tobia, F.: Collective calibration of active camera groups. In: Advanced Video and Signal Based Surveillance. (2005) 456–461
7. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell **22** (2000) 1330–1334
8. Sinha, S.N., Pollefeys, M.: Synchronization and calibration of camera networks from silhouettes. In: International Conference on Pattern Recognition (ICPR). (2004) 116–119
9. Szlávik, Z., Szirányi, T., Havasi, L., Benedek, C.: Optimizing of searching comotion point-pairs for statistical camera calibration. In: IEEE ICIP. (2005) 1178–1181
10. Zotkin, D.N., Duraiswami, R., Davis, L.S.: Joint audio-visual tracking using particle filters. EURASIP Journal on Applied Signal Processing (2002) 1154–1164
11. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
12. Camera Calibration Toolbox for Matlab http://www.vision.caltech.edu/bouguetj/calib_doc/.