A Multi-Camera Pose Tracker for Assisting the Visually Impaired

Frank Dellaert and Sarah Tariq
College of Computing, Georgia Institute of Technology
{*dellaert,sarah*}*@cc.gatech.edu*

Abstract

6DOF Pose tracking is useful in many contexts, e.g., in augmented reality (AR) applications. In particular, we seek to assist visually impaired persons by providing them with an auditory interface to their environment through sonification. For this purpose, accurate head tracking in mixed indoor/outdoor settings is the key enabling technology. Most of the work to date has concentrated on single-camera systems with a relatively small field of view, but this presents a fundamental limit on the accuracy of such systems. We present a multi-camera pose tracker that handles an arbitrary configuration of cameras rigidly fixed to the object of interest. By using multiple cameras, we increase both the robustness and the accuracy by which a 6-DOF pose is tracked. However, in a multi-camera rig setting, earlier methods for determining the unknown pose from three world-to-camera correspondences are no longer applicable, as they all assume a common center of projection. In this paper, we present a RANSAC-based method that copes with this limitation and handles multi-camera rigs. In addition, we present quantitative results to serve as a design guide for full system deployments based on multi-camera rigs. Our formulation is completely general, in that it handles an arbitrary, heterogeneous collection of cameras in any arbitrary configuration.

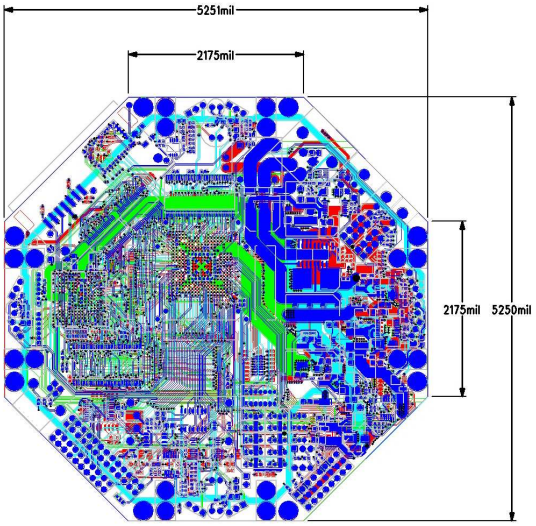


Figure 1: PCB layout for the miniature camera rig we have designed and are testing now. The octagonal board can support up to 4 CMOS cameras (mount holes can be seen on every other side) and the board has room for an Xscale processor and a Xilinx field-programmable gate array (FPGA), which will handle the feature detection in parallel for all cameras.

1. Introduction

Pose tracking is useful in augmented reality (AR) applications where accurate head pose over time is required. We are especially interested in wide-area, markerless tracking using computer vision, which is well-suited to both indoor and outdoor environments. *In particular, we seek to assist visually impaired persons by providing them with an auditory interface to their environment through sonification. For this purpose, accurate 6DOF head tracking is the key enabling technology.*

Most of the work to date has concentrated on single-camera systems with a relatively small field of view. Using a single camera presents a fundamental limit on the accuracy of such systems, because features are only observed in a single viewing direction. While wide-angle lenses or mirror-based systems are possible solutions, these systems typically suffer from low resolution which makes detecting and tracking landmarks difficult.

We present a multi-camera pose tracker that can handle an *arbitrary configuration of cameras* rigidly fixed to the object of interest. By using multiple cameras, we can detect and track landmarks in different directions at high resolution, and hence increase both the robustness and the accuracy by which a 6-DOF pose can be tracked. However, since there is no longer a single center of projection, traditional three-point algorithms to determine pose from landmark observations are not applicable. To remedy this, we developed a RANSAC-based method

that handles multi-camera rigs using a fast non-linear minimization step in each RANSAC round. In this respect, we address the same problem as the algebraic solution method developed concurrently by [20]. As a second contribution, we present the results of a thorough quantitative evaluation of the method in a realistic markerless tracking scenario, to serve as a design guide for full system deployments based on multi-camera rigs.

1.1. Related Work

One of the fundamental tasks in AR applications is tracking pose, see for example [14] or [3]. Klein and Drummond [13] note that augmentation results using vision are generally more accurate than with other sensors. Traditionally, vision-based trackers have relied upon fiducial markers, but this is often undesirable for a number of obvious reasons (cost, maintenance, accessibility), and there has been a move toward markerless vision based tracking. Several types of features have been used, including line segments, groupings of edges, regions [23], and point-based features [23, 26, 24, 13].

The vision community recently noted the superiority of affine invariant features in object recognition, matching, and indexing [22, 15, 18, 19, 6]. These features provide robustness against partial occlusion, nearby clutter, illumination and viewpoint changes, and object pose variations. However, using these features induces a non-trivial computational burden [24]. We note however, that recently impressive frame rates have been demonstrated in vision using high end programmable hardware [1], and we are in the process of developing an FPGA-based solution to quickly detect affine invariant features (Section 3) which should alleviate these concerns.

The key computational step in vision-based tracking with a single camera is determining the pose of the camera from a set of correspondences between 3D reference points and their images. This is one of the oldest and most important problems in computer vision and photogrammetry [21]. Fast closed-form methods to accomplish this (see [9, 21] for an overview) are typically used within a random sample consensus (RANSAC) scheme [7] to reject spurious correspondences proposed in a putative matching step. Alternatively, recursive estimation methods such as extended Kalman filters [4, 25] can be used in conjunction with validation gates.

The usefulness of omni-directional video for tracking has been noted in the robotics literature [16, 12]. Hence, below we propose a tracking system consisting of an arbitrary configuration of cameras rigidly attached to the object whose pose is of interest. [20] has very recently presented an algebraic solution method, developed concurrently and independently from us. Our approach is arguably slower but considerably simpler to implement.

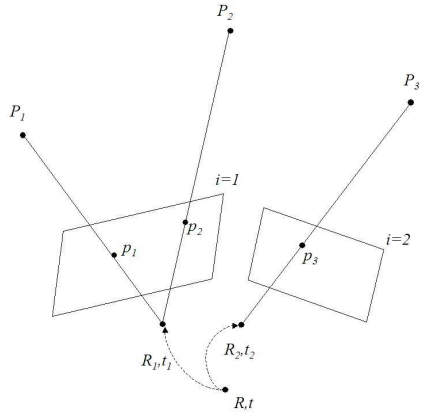


Figure 2: Multi-camera rig example.

2. Multi-Camera Pose Tracking

In a multi-camera rig setting, the minimal number of correspondences between image measurements and known world landmarks is still three, but these three correspondences no longer imply viewing directions from a common camera center. To cope with this much more difficult situation, we propose to use a fast non-linear minimization step in each RANSAC round, that accurately models the viewing geometry.

2.1. Multi-Camera Rig Geometry

We assume a measurement model whereby n previously surveyed landmarks $\{P_j\}_{j=1}^n$, with $P_j \in \mathbb{R}^3$, are observed by a multi-camera rig consisting of m cameras, as illustrated in Figure 2. This yields n measurements $\{(i_j, p_j)\}_{j=1}^n$, each consisting of a camera index $i_j \in 1..m$ and a 2D landmark *image* $p_j \in \mathbb{R}^2$. Also assumed known is the calibration of the rig $K_R \triangleq \{(K_i, R_i, t_i)\}_{i=1}^m$, with m the number of cameras. Hence, given the global pose (R, t) of the entire rig in a given reference frame, we obtain the following measurement equations for each 3D to 2D correspondence (P, i, p):

$$p = \Pi_i(K_i, R_i(R(P - t) - t_i) + n_i$$

where

- $P' \triangleq R \times (P - t)$ are the rig-centered 3D coordinates of the landmark P.
- $P^{(i)} \triangleq R_i \times (P' - t_i)$ are the camera-centered 3D coordinates of the landmark P in camera i.
- $y \triangleq \Pi(K_i, P^{(i)})$ is the ideal projection of the camera-centered point $P^{(i)}$ into 2D image coordi-

nates of camera i, according to intrinsic calibration parameters K_i.

- n_i is an additive 2D noise vector whose density is assumed known in all cameras
- $p = y + n_i$ is the final observed 2D image measurement

The formulation above is completely general in that it handles heterogeneous camera rigs with arbitrary relative poses. For example, it can model a mix of standard perspective cameras with a centered omni directional catadioptric camera. In the results below we concentrate on a set of identical perspective cameras symmetrically arranged in a ring. The formulation above is still needed in its full generality, however, to model the differences in calibration and mounting inaccuracies inevitable in a real (low-cost) system.

For the common case of perspective cameras, the calibration parameters K_i for each rig consist of the 5 usual intrinsic parameters

$$K = \begin{bmatrix} f_x & s & u_0 \\ & f_y & v_0 \\ & & 1 \end{bmatrix}$$

and the projection function $\Pi$is given by

$$\Pi([X, Y, Z]^T) = [f_x\, x + s\, y + u_0, f_y\, y + v_0]^T$$

Radial distortion is easily incorporated but the details are omitted here, see e.g. [11].

2.2. Pose from Known Correspondences

Given a list of n correspondences $\{(P_j, i_j, p_j)\}_{j=1}^n$, we can optimally estimate the rig pose (R, t) by maximum a posteriori (MAP) estimation:

$$(R,t)^* \quad = \quad \underset{R,t}{\text{argmax}} \left\{ P(R,t) \prod_{j=1}^{n} P\left(P_j, i_j, p_j | R, t\right) \right\}$$

where we applied Bayes law and assumed conditional independence of all measurements p_j given the rig pose. The prior $P(R, t)$ can be derived from the previous time step using a motion model, using standard recursive estimation methods. The above is easily extended to incorporate rate variables and/or other sensor modalities, as has been adequately described elsewhere [17]. Hence, in the following we will assume the prior $P(R, t)$ to be of known form.

Given an initial estimate $(R^{(0)}, t^{(0)})$, e.g. the mean of the predictive density $P(R, t)$, we can now optimize for (R, t) using standard non-linear minimization techniques. The above formulation supports Gaussian as well as robust, non-Gaussian noise models. However, in the common case of assumed Gaussian noise, we obtain the following non-linear minimization problem

$$(R,t)^* = \underset{R,t}{\text{argmin}} \left\{ \frac{1}{2} \sum_{j=1}^{n} J(P_j, i_j, p_j) - \log P(R,t) \right\}$$

where $J(P_j, i_j, p_j)$ is the objective function contribution resulting from the j^{th} correspondence, given by

$$J(P, i, p) \triangleq \|p - \Pi_i(K_i, R_i(R(P - t) - t_i)\|^2_{\Sigma_i} \quad (1)$$

Here $\|\mu - x\|^2_\Sigma$ in (1) is the squared *Mahalanobis distance* from x to μ and defined as below:

$$\|\mu - x\|^2_\Sigma \triangleq (\mu - x)^T \Sigma^{-1} (\mu - x)$$

Minimization is implemented using a standard Levenberg-Marquardt non-linear optimization scheme in conjunction with a sparse QR solver. A crucial step is the computation of the $2n \times 6$ Jacobian H at every iteration. H has $2n$ rows, 2 rows for each measurement, and 6 columns for each of the 6 degrees of freedoms (3 translation, 3 rotation). We handle rotations in terms of an incremental Euler parameterization around the current estimate. To compute the Jacobian H, we use an in-house automatic differentiation (AD) framework. AD is neither symbolic nor numerical differentiation, and calculates the Jacobian at any given value exactly, efficiently, and free of numerical instabilities. See [8] for more details.

2.3. Robust Outlier Rejection

In a tracking context, we then use RANSAC to obtain a robust pose estimate using the machinery in Section 2.2 as a subroutine. At each step, we assume that a number of putative 3D to 2D correspondences $\{(P_j, i_j, p_j)\}_{j=1}^N$ can be obtained, with $N \gg 3$. In Section 3 below we present one way to do this, but any method will do. We then use RANSAC [7] to obtain a set of inlier correspondences. Briefly, we randomly select minimal sets of 3 correspondences, obtain the MAP pose, and check for support among the other inliers. We use an adaptive threshold version of RANSAC to automatically determine the number of RANSAC rounds needed, see e.g. Hartley and Zisserman [11] for a thorough exposition. As a final step, the basis set of correspondences with the highest support is then used with its inlier support to refine the MAP pose estimate.

3. A Complete System

3.1. Overview

We implemented a markerless tracking system based on affine invariant features, popular in object recognition

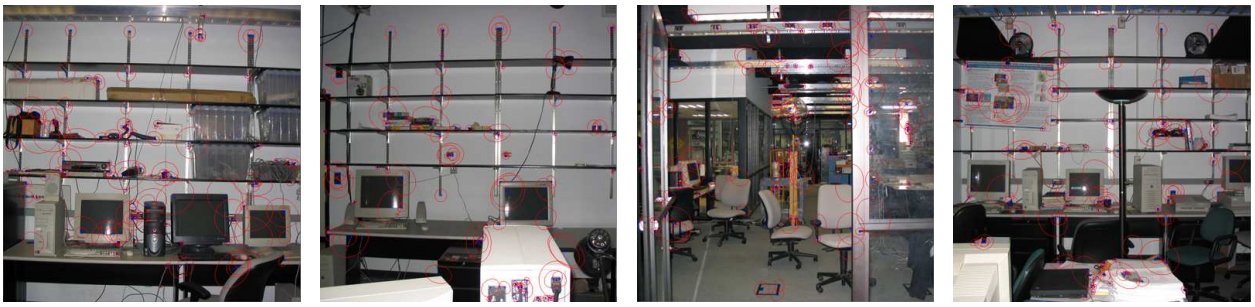


Figure 3: The database of Features is constructed by finding interest points and their affine invariant descriptors in the environment. Here we show four images from the environment. The blue crosses denote the location of the interest point, and the red ellipses the region of the descriptor

[22, 2, 18, 19, 6]. We implemented the run-time pipeline from images to pose, but were not yet able to test the system in real environments as the hardware component is still under development; the hardware will consist of a head-mounted miniature camera rig currently under development (see Figure 1). We are also developing an FPGA-based solution to detect affine invariant features in real time for up to 4 cameras in parallel.

As explained above, the system estimates pose by finding 2D to 3D point correspondences between the images captured from the rig and survey features in the environment. RANSAC is then used to robustly estimate the true pose of the rig. To deploy or test the system, we need a surveying phase to create known 3D landmarks, after which we can run the pose-tracking at run-time.

3.2. Landmark Surveying

In the surveying phase the system detects affine invariant features in the environment and logs them in a database along with their locations. The location estimation for the features can be done completely automatically using well-known structure from motion approaches [11]. Affine invariant features are found using the method outlined in [2]. We first find scale-space features in the images by detecting Harris features [10] at a number of scales, ordering them according to their strength and picking the top n features. Next we calculate a descriptor for each feature based on an affine invariant region around it. To compress the database and enable faster comparisons between features we perform principal component analysis (PCA) on the descriptors, keeping only the first 20 eigenvectors. Descriptors are reduced from 625 bytes per feature to 80 bytes, reducing the storage cost and comparison time per feature.

3.3. Run-time Tracking

In the tracking step we estimate at each frame the absolute pose of the rig relative to the environment. We first detect affine invariant features in the images from the rig, as outlined in Section 3.2. These features are then projected into the eigenspace of the database. Putative correspondences are obtained by finding the closest feature in the database for each image feature, rejecting those with a large error. To estimate the pose of the rig using RANSAC we need initial estimate, which can be calculated in two ways: (a) if we know that the pose of the rig is almost planar then we can use the quick linear estimate outlined in [5]; (b) we can also use the pose estimate obtained at the previous frame.

4. Results

To demonstrate the quality of the proposed system we conducted extensive experiments in a synthetic environment using real human motion. The environment consists of texture mapped planes, see figure 3. Affine invariant features are detected in all the textures and their 3D locations are derived from the positions of the planes. Note that although we use only planes in our experiments for ease of building the database, our method is not restricted to them and can in fact work on arbitrarily complex environments. In all experiments the movement was obtained using Motion Capture of realistic human motions. Data was captured at a rate of 120 frames per second with translational units of millimeters. The data was converted into head poses for each frame and scaled down to fit the dimensions of the synthetic environment. The poses at each frame were used to capture images from a synthetic rig in our environment, and estimates of the poses were generated from these images according to the method outlined in section 3. These estimates were then scaled up appropri-

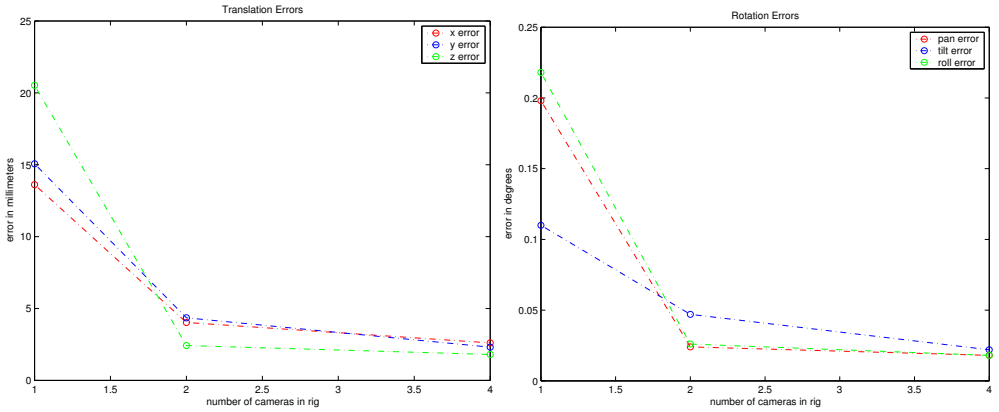


Figure 4: RT: 515 frames. Average deviations from ground truth for a varying number of cameras.

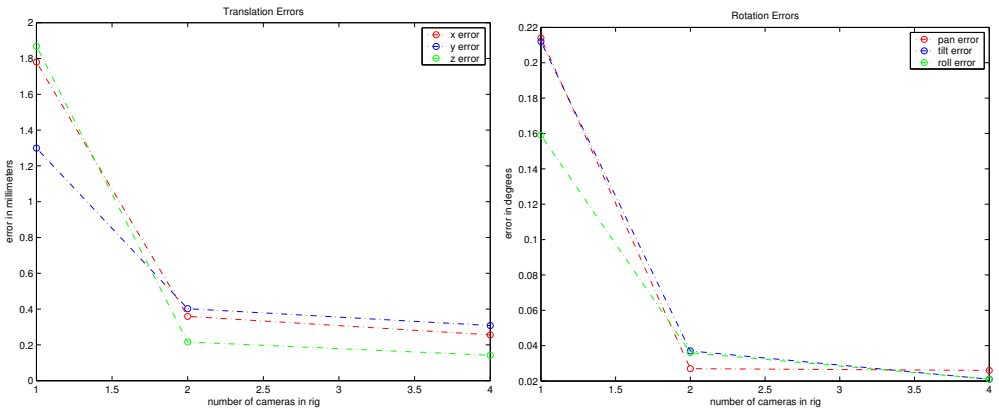


Figure 5: I1: 800 frames.Average deviations from ground truth for a varying number of cameras.

ately to ensure that all results were in millimeters. All experiments were done on an Intel Pentium 4 machine running at 2.80 GHz. Below we present results from four of the sequences:

1. RT: a small sequence (515 frames) in which the subject makes a right turn.
2. I1: a medium sized sequence (800 frames) of a subject looking around an environment
3. I2: a large sequence (1172 frames) of a subject looking around an environment
4. SS1:a large sequence (3386 frames) with large motion and relatively large out of plane rotations.

4.1. different number of cameras

Tables 1 to 4 show the mean translational and rotational errors from the ground truth for each of the four sequences. Errors are calculated by summing the absolute difference of the real and estimated poses at each frame and dividing by the total number of frames. In Figures 4 to 8 we show these translational and rotational errors graphically.

#	fps	pan	tilt	roll	x	y	z
1	19	0.20	0.11	0.22	13.6	15.1	20.5
2	12	0.02	0.05	0.03	4.03	4.36	2.43
4	11	0.02	0.02	0.02	2.61	2.31	1.80

Table 1: RT: 515 frames. Average deviations from ground truth for a varying number of cameras. The rotational errors are in degrees and the translational errors are in mm

These results convincingly demonstrate the advantage of using a multi-camera rig tracker over a single, limited field of view camera. Both translational and ro-

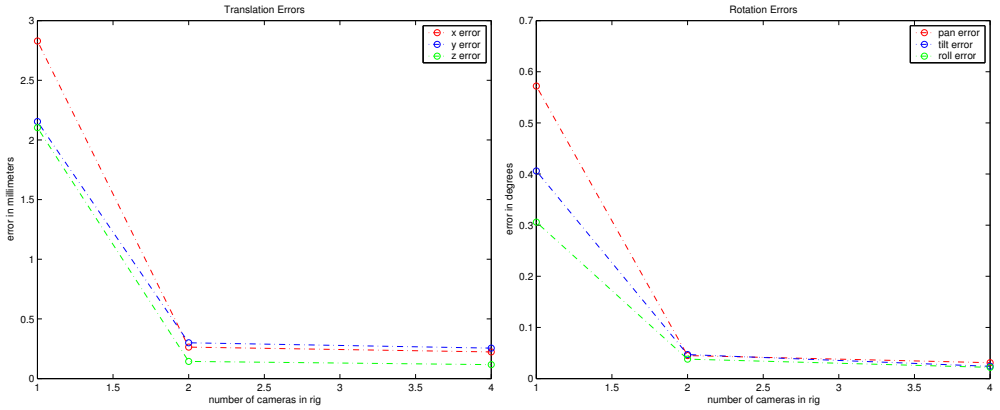


Figure 6: I2:1172 frames. Average deviations from ground truth for a varying number of cameras.

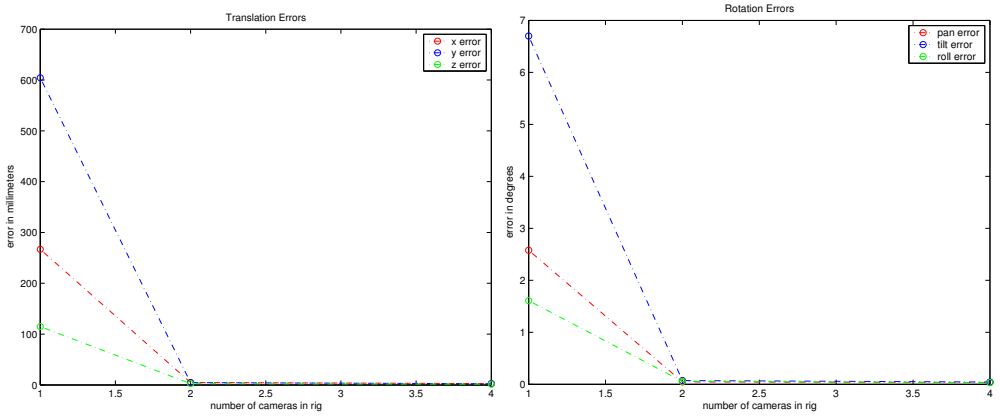


Figure 7: SS1:3386 frames. Average deviations from ground truth for a varying number of cameras

#	fps	pan	tilt	roll	x	y	z
1	14	0.21	0.21	0.16	1.78	1.30	1.88
2	13	0.03	0.04	0.04	0.36	0.40	0.22
4	12	0.03	0.02	0.02	0.26	0.31	0.14

Table 2: I1: 800 frames.Average deviations from ground truth for a varying number of cameras. The rotational errors are in degrees and the translational errors are in mm

#	fps	pan	tilt	roll	x	y	z
1	8	0.57	0.41	0.31	2.83	2.15	2.10
2	9	0.05	0.05	0.04	0.26	0.30	0.14
4	9	0.03	0.02	0.02	0.22	0.26	0.12

Table 3: I2:1172 frames. Average deviations from ground truth for a varying number of cameras. The rotational errors are in degrees and the translational errors are in mm

tational errors decrease substantially as the number of cameras is increased, and this happens consistently over all types of of mocap sequences.

The average error for the SS1 sequence is dramatically higher for the one-camera case, which warrants closer examination. Therefore, in Figure 9 (a) and (b) we have plotted the time-series of both translation along the X-axis and the tilt, for 1 camera and 4 cameras, respectively. From the figures one can see that, especially when the tilt angles were large, a considerable number of catastrophic failures occurred. Our hypothesis is that in the one camera case, when the subject looks up or down the number of correct putative correspondences falls below three, but we have not yet been able to very

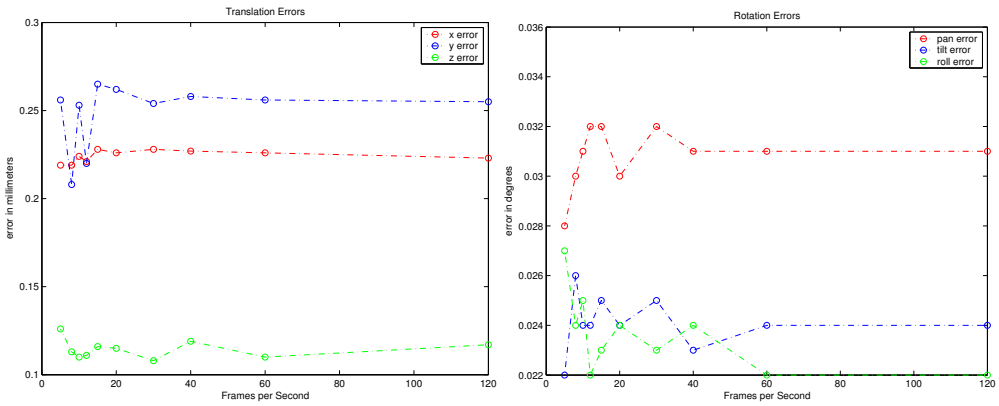
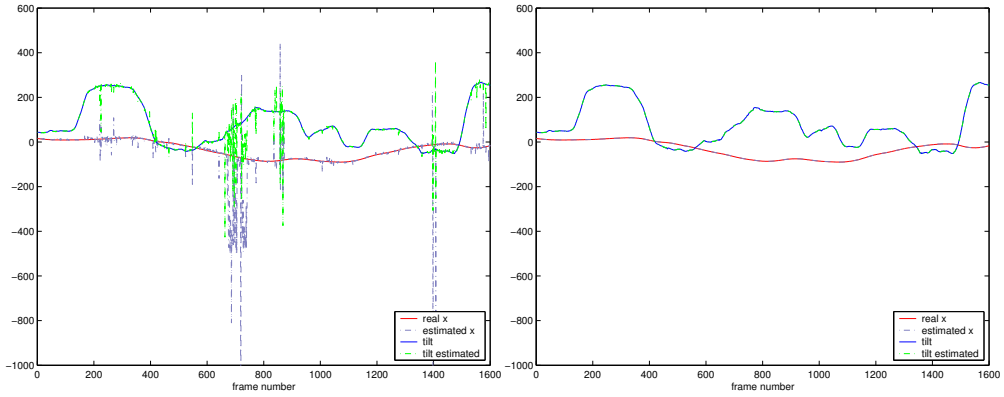


Figure 8: SS1:3386 frames. Average deviations from ground truth for a varying number of frames



(a) 1 camera

(b) 4 cameras

Figure 9: Time series for SS1 showing the real and estimated translations along x and real and estimated tilt.

verify that. The problem disappears for 2 or 4 cameras as they are able to acquire a large number of correctly matched features.

4.2. Varying frame rates

#	fps	pan	tilt	roll	x	y	z
1	4	2.58	6.70	1.61	266.8	604.4	115.0
2	6	0.05	0.07	0.06	4.73	4.55	2.63
4	8	0.04	0.04	0.03	2.75	2.45	1.52

Table 4: SS1:3386 frames. Average deviations from ground truth for a varying number of cameras. The rotational errors are in degrees and the translational errors are in mm

To demonstrate the robustness of the proposed system to errors in the initial estimate we conducted experiments with different frame rates. In table 5 and figure 8 we show the results obtained by changing the input frame rate. To generate sequences with lower frame rates we sampled the 120 frames per second sequence at appropriate intervals. These down sampled sequences were then used in tracking and errors were calculated in a similar manner to section 4.1. The results show that for frame rates as low as 5 frames a second we get very good quality results for both translational and rotational pose variables even though the initial estimate is of much lower quality than at 120 fps.

Fps	pan	tilt	roll	x	y	z
120	0.031	0.024	0.022	0.223	0.255	0.117
60	0.031	0.024	0.022	0.226	0.256	0.110
40	0.031	0.023	0.024	0.227	0.258	0.119
30	0.032	0.025	0.023	0.228	0.254	0.108
20	0.030	0.024	0.024	0.226	0.262	0.115
15	0.032	0.025	0.023	0.228	0.265	0.116
12	0.032	0.024	0.022	0.221	0.220	0.111
10	0.031	0.024	0.025	0.224	0.253	0.110
8	0.030	0.026	0.024	0.214	0.208	0.113
5	0.028	0.022	0.027	0.219	0.256	0.126

Table 5: SS1:3386 frames. Average deviations from ground truth for varying frame rates. The rotational errors are in degrees and the translational errors are in mm

5. Discussion

We introduced a pose tracking method that can be used with arbitrary multi-camera configurations, in either fiducial or markerless tracking settings. In the context of the markerless tracking system we developed, it has the potential to outperform single-camera systems by a wide margin. We tested the system in software on realistic image sequences, using motion capture data to guarantee realistic motion.

Clearly, the complete system now has to be validated in real-time on real image sequences rather than synthetic ones. To that end, we are developing a FPGA-based miniature camera rig that will be able to perform the detection of affine invariant features in real time, for multiple cameras in parallel.

References

[1] A.Darabiha, J.R.Rose, and W.J.MacLean. Video rate stereo depth measurement on programmable hardware. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 18–20, 2003.

[2] A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 774–781, 2000.

[3] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[4] T. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Trans. Aerosp. Electron. Syst.*, 26(4):639–656, July 1990.

[5] F. Dellaert and A. Stroupe. Linear 2D localization and mapping for single and multiple robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, May 2002.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[7] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.

[8] A. Griewank. On Automatic Differentiation. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 83–108. Kluwer Academic Publishers, 1989.

[9] R. Haralick, C. Lee, K. Ottenberg, and M. Noelle. Review and analysis of solutions to the three point perspective pose estimation problem. *Intl. J. of Computer Vision*, 13(3):331–356, 1994.

[10] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey*, pages 147– 152, 1988.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[12] M. Jogan and A. Leonardis. Robust localization using panoramic view-based recognition. In *Intl. Conf. on Pattern Recognition (ICPR)*, pages 136– 139, 2000.

[13] G. Klein and T. Drummond. Robust visual tracking for non-instrumented augmented reality. In *ISMAR*, 2003.

[14] K.Meyer, H. Applewhite, and F.A.Biocca. A survey of position trackers. *Presence: Teleoperators and Virtual Environments*, 1(2):173– 200, 1992.

[15] D. Lowe. Object recognition from local scale-invariant features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[16] L.Paletta, S.Frintrop, and J.Hertzberg. Robust localization using context in omnidirectional imaging. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2001.

[17] P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York, 1979.

[18] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Intl. Conf. on Computer Vision (ICCV)*, 2001.

[19] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Eur. Conf. on Computer Vision (ECCV)*, volume 1, pages 128–142, 2002.

[20] D. Nistér. A minimal solution to the generalised 3-point pose problem. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 560–567, 2004.

[21] L. Quan and Z.-D. Lan. Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(8):774–780, 1999.

[22] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(5):530–535, 1997.

[23] V.Ferrari, T.Tuytelaars, and L.VanGool. Markerless augmented reality with real-time affine region tracker. In *ISAR*, 2001.

[24] V.Lepetit, L.Vacchetti, D.Thalmann, and P.Fua. Fully automated and stable registration for augmented reality applications. In *ISMAR*, 2003.

[25] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information. *Computer Graphics*, 31(Annual Conference Series):333–344, 1997.

[26] Y.Genc, S.Riedel, F.Souvannavong, C.Akmlar, and N.Navab. Marker-less tracking for ar: A learning-based approach. In *ISMAR*, 2002.