

Real-time 3D Camera Tracking for Industrial Augmented Reality Applications

Gabriele Bleser
Koblenz-Landau University
Universitätsstraße 1
56070 Koblenz, Germany

Yulian Pastarmov
Fraunhofer IGD
Fraunhoferstraße 5
64283 Darmstadt, Germany

Didier Stricker
Fraunhofer IGD
Fraunhoferstraße 5
64283 Darmstadt, Germany

{gbleser, ypastarm, stricker} @ igd.fhg.de

ABSTRACT

In this paper we present a new solution for real-time 3D camera pose estimation for Augmented Reality (AR) applications. The tracking system does not require special engineering of the environment, such as placing markers or beacons. The required input data are a CAD model of the target object to be tracked, and a calibrated reference image of it. We consider the whole process of camera tracking, and developed both an autonomous initialization and a real-time tracking procedure. The system is robust to abrupt camera motions, strong changes of the lighting conditions and partial occlusions. To avoid typical jitter and drift problems the tracker performs feature matching not only in an iterative manner, but also against stable reference features, which are dynamically cached in case of high confidence. We present experimental results generated with help of synthetic ground truth, real off-line and on-line image sequences using different types of target objects.

Keywords

Computer vision, real-time marker-less camera tracking, automatic initialization, augmented reality

1. INTRODUCTION

Augmented Reality (AR) opens new perspectives for a lot of application areas [Azum95], such as maintenance of machines, design, medicine [Bock03, Wesa04], or cultural heritage [Vass02]. Nevertheless one major difficulty of AR is the user-tracking, which is often unstable or requires special infrastructure in the environment, and thus limits severely the application. Computer vision based methods provide the best accuracy, and represent the currently most developed approach. They rely on 2D/2D or 2D/3D correspondences between features (interesting points, edges, regions) of the image frames. This can be either artificially designed and positioned patterns (marker-based tracking) [Thom97] or natural characteristics of the scene (markerless tracking) [Lauc00, Poll99]. For industrial application, the preparation of the scene with markers is not economically viable, so

that only marker-less solutions are accepted.

There are generally two approaches to this problem. The global image-based approach computes a 2D transformation, which registers the current frame as a whole on a reference pattern. Stricker [Stri01] uses the Fourier-Mellin Transform to retrieve an Euclidian transformation between the incoming frame and one from a set of calibrated reference images. The current pose is deduced from this transformation assuming the camera being fixed to a tripod and the viewer being far away from the scene. An advantage of the registration on reference images is that no error is accumulated over time (drift).

The local model-based approach fully solves the 3D problem estimating the current pose with 6 DOF. It is very similar to the marker-based approach searching for 2D/3D correspondences by using natural features instead of artificial ones. An interesting approach is presented in [Lepe03]. To avoid jitter and drift during tracking, he merges the information of subsequent frames with that of off-line calibrated reference images. Genc [Genc02] provides useful criteria of stable features. Comport [Comp03] uses edge features instead of points.

Interesting developments have been made in the field of feature extraction during the last years. Classical methods like KLT tracker [Shi94] or Harris detector

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG 2005 conference proceedings, ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

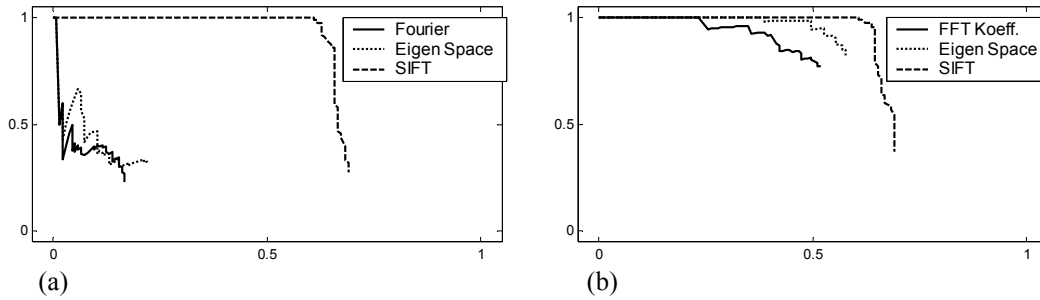


Figure 1: Precision-recall-plots showing the behavior of different descriptors concerning Gaussian blur and Euclidian transformations: (a) 10 degrees rotation, 1.1 scaling and (b) 30 degrees rotation, 1.4 scaling.

[Harr88] with correlation assume small frame-to-frame difference and restrict the searching region of a feature in the near neighborhood of its previous location. Therefore they fail in case of wide baseline. SIFT [Lowe99] has been developed for that particular purpose. The so-called SIFT-keys are Euclidean invariant, robust against small affine and 3D projective transformations, and linear lighting changes. Moreover they can be stored and matched efficiently without the related images or patches.

2. APPROACH

The approach presented here provides a full solution of 3D pose estimation without any restrictions on camera motions or the scene configuration. Changes of the lighting conditions are also handled in a reasonable range. Our method consists in 3D-model-based tracking with SIFT features "lying" on a predefined CAD model of a target scene object. The system initializes autonomously and recalibrates itself in case of tracking failure. It is designed for small environments and because of its robustness, speed and fast automatic (re-)initialization, it is particularly usable in industrial processes. The approach is similar to Lepetit's one, but the feature extraction is different and enables to reduce the amount of offline data and to use a single reference image. The feature matching is done not only in iterative manner, but also against the features of the reference image in order to avoid drift. To be independent of occlusions, new features are taken into account by back-projecting them onto the CAD model. We thus generate in a dynamic way new 2D/3D correspondences after successful pose estimation and can handle large changes of the features.

3. INITIALIZATION

The initialization method yields the initial camera pose within a global reference coordinate system using one calibrated "bootstrap" reference frame. It is based on the matching of the features and provides proper results even if the initial pose is rela-

tively far away from that related to the reference image.

Only a minimal preparation is required. It consists in taking one photograph of the target object and calibrating it manually. Extrinsic parameters are computed by choosing at least four correspondent points on the 3D model and the snapshot.

When the tracking system starts, the initialization procedure is invoked receiving one calibrated reference frame, a CAD model of the target object and the incoming frame as input. Firstly it performs feature extraction from the reference frame and back-projects all interesting points on the 3D model to obtain 3D coordinates. The back-projection is done by sending rays from the related camera position through the image plane and computing their intercept points with the 3D model. The intersection test is implemented with OpenSG, which employs bounding volume hierarchies for efficient ray tracing. All features that lie on the target object surface are kept ready for (re-)initialization during the whole system run-time. Now we have a reference set of 2D/3D corresponding points and features describing their local appearances within the reference image. The following steps consist in firstly detecting and matching features from the incoming frame with those of the reference image, removing spurious matches by applying geometric constraints and afterwards estimating the initial pose from the resulting 2D/3D correspondences if there are enough (usually more than 10). These steps are processed for each incoming frame until the initial pose could be determined adequately. If the initialization has been successful, the features and corresponding 2D and 3D points of the current frame are added to the reference data set, whereas those, which have not been matched, are back-projected from the initial pose to obtain 3D coordinates. This technique dynamically extends reference data for further (re-)initializations whereas the dynamically added reference features resemble current lighting conditions and camera parameters better than the bootstrap reference frame.

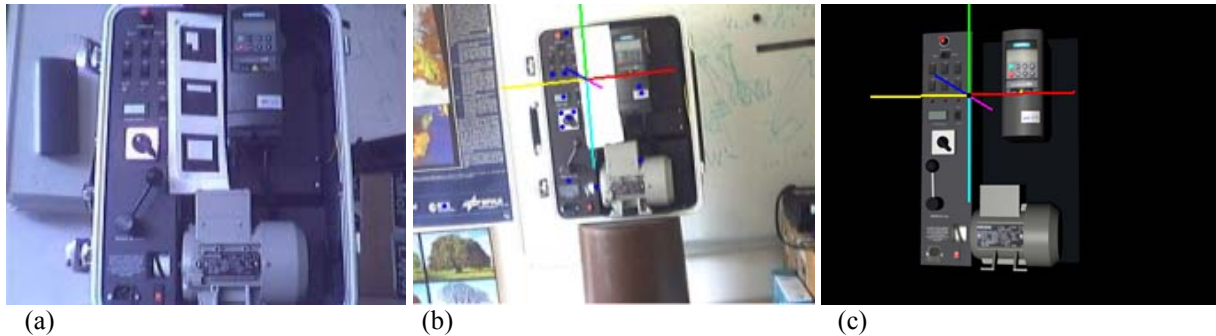


Figure 2: Example of automatic initialization: (a) shows the reference frame, (b) the incoming frame. Note, that the viewpoints as well as the lighting conditions are quite different. The incoming frame is augmented by the matched feature points (blue) and the axis of the world coordinate system projected from the calculated pose. A synthetic image generated from the textured CAD model shows the correct projection of the coordinate axis (c). This allows for optical verification of the calculated pose. The matching yielded 17 point correspondences. 10 were accepted by RANSAC running 50 iterations.

Initial Feature Matching

Unlike iterative feature tracking, the initial matching between features of the reference image and the incoming frame cannot profit from temporal coherence. The initial camera pose can be relatively far away from that related to the reference frame. So there exists no initial guess of the current pose and, consequently, no meaningful assumption about feature displacements or the 2D location of the target object within the current frame. The whole incoming frame has to be processed in terms of feature extraction solving the subsequent matching as problem with quadratic costs. Moreover, a drastic change of lighting conditions often occurs in praxis. Ideally, viewpoint and illumination insensitive feature extraction and characterization methods yielding small and distinctive descriptors are required for the tracker initialization. Scale-invariant features can be identified by looking for local optima of pyramidal difference-of-Gaussian functions in scale-space [Lowe99]. We took this approach for feature detection and performed a simple test comparing the robustness of different local descriptors in terms of "precision" and "recall" at the end of which we chose the most appropriate one for feature characterization in our initialization procedure. The test consists in extracting scale-invariant points from two images, - whereat the second image was synthetically generated from the first, applying different Euclidian transformations and adding Gaussian noise of standard deviation 10 - characterizing there local point neighborhoods by the different descriptors, finding matches based on simple Euclidean vector distance in combination with a threshold and obtaining precision-recall-curves by varying this threshold.

Precision is defined as the rate between the number of correct matches and the number of returned matches. It is the opposite of the outlier rate. Recall

is defined as the number of correct matches against the number of possible matches.

We tested three different descriptors. The first one characterizes interesting points by applying the Fourier transformation to their local fixed-sized point neighborhoods and choosing few Fourier coefficients (FFT Koeff) [Spie00]. The second one describes them as coordinates within a low dimensional coordinate system (Eigen space), spanned by the Eigen vectors of the covariance matrix of a training set [Turk91]. Lepetit [Lepeti03] uses this approach for tracker initialization, whereat the Eigen-space is computed offline for the set of reference images. The last descriptor uses an orientation histogram of the image patch centering the point for characterization (SIFT). Furthermore, the size of the local neighborhood is adapted to the pyramid level, in which the point has been detected, and the histogram is given relative to the major gradient orientation. The resulting plots (see Fig. 1) show the overall Euclidian invariance of the SIFT keys and made our decision to choose SIFT for initial feature matching. Unlike Eigen features a further advantage of SIFT is its independency from a training set. SIFT features are computed autonomously and therefore support dynamical generation of reference data, if the camera pose is highly correct. For key generation we use as described in Lowe's paper, a 4x4 subsampling of the local point neighborhood and consider 8 discrete gradient orientations. So the resulting vector contains 128 elements and can be stored and matched efficiently without the related image.

Feature matching follows a simple criterion. A 2D point \mathbf{a} of the reference set matches a 2D point \mathbf{b}_i of the incoming frame, if the Euclidian distance between the related descriptors \mathbf{D}_a and \mathbf{D}_{b_i} is minimal and an additional rule is fulfilled. Let \mathbf{D}_{b_i} be the

descriptor related to point \mathbf{b}_j with the second nearest Euclidian distance to \mathbf{D}_a :

$$\| \mathbf{D}_a - \mathbf{D}_{b_i} \| < t \cdot \| \mathbf{D}_a - \mathbf{D}_{b_j} \| \quad (1)$$

$$t \in [0 \dots 1]$$

The threshold (e.g. $t = 0.6$) sees that badly defined matches are discarded immediately. This includes interesting points on periodic image textures. If multiple matches occur, we choose the one with the smallest Euclidian distance and discard the others. Now we have a set of 2D/2D correspondences between the incoming frame and the reference set. Since the latter also contains 3D coordinates for every interesting 2D feature, we easily obtain 2D/3D correspondences by connecting every 2D point of the incoming frame with the 3D coordinate related to its matching reference point.

Initial Pose Estimation

To obtain a well-conditioned set of 2D/3D correspondent points for pose estimation, we filter the correspondences retrieved from the matching by applying geometric constraints. We use the RANSAC algorithm by employing the projection matrix as geometric model [Hart00]. The projection matrix is calculated linearly from four sample correspondences by either using the original POSIT algorithm or the extension for coplanar model points according to the configuration of the sample set [DeMe92]. We obtain good results and not more than 50 RANSAC iterations were sufficient. Having robustly removed all outliers, the current pose is linearly estimated from all remaining 2D/3D correspondent points $\mathbf{m}_i \leftrightarrow \mathbf{M}_i$. To obtain the final pose, we optimize the reprojection errors over camera rotation \mathbf{R} and translation \mathbf{T} using the linear estimation as initial guess:

$$\min_{\mathbf{R}, \mathbf{T}} \sum_i \| \mathbf{m}_i - \mathbf{m}_i' \|^2 \quad (2)$$

\mathbf{m}_i' is the projection of \mathbf{M}_i from the current pose and \mathbf{R} is parameterized by a rotation axis and an angle.

Results

Figure 2 shows an example of our autonomous initialization method. The markers fixed to the target object have been occluded during initialization and have no influence on the procedure. Tracker initialization succeeded for the first incoming frame, although the initial pose was relatively far away from that related to the reference frame especially concerning the distance from the target object. Moreover the lighting conditions are quite different.

4. TRACKING

Abrupt motions and drastic changes of lighting conditions are typical for AR applications and do not only make initialization but also iterative tracking difficult. The paradigm of temporal coherence, which is the underlying principle for most traditional feature tracking methods basing on correlation, is often not fulfilled. For that reason we decided to treat the problem of iterative tracking similar to that of tracker initialization, i.e. as a matching problem with quadratic costs. To speed up the tracker we do not process the whole incoming frame in terms of feature extraction, but only a fixed-sized image region, which follows the projection of the target object within the image. Because of the latter, most features we loose by regarding not the whole image do not lie on the model surface anyway and would be useless for pose estimation. So if the size of the tracking region isn't chosen too small, the tracking procedure doesn't suffer from this technique but works far more efficiently.

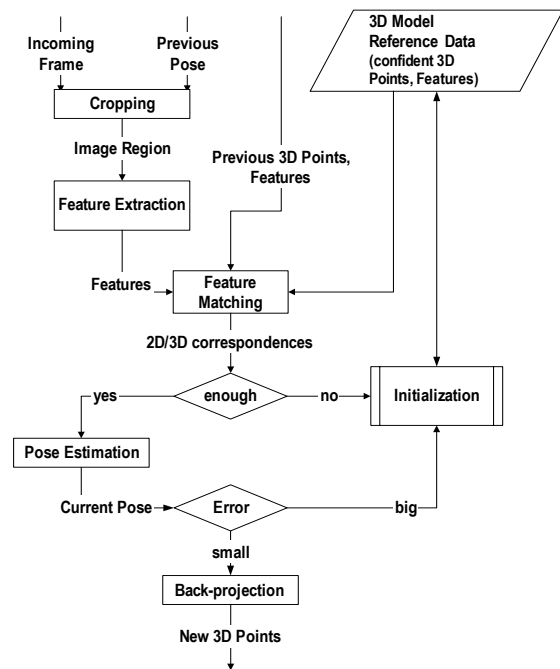


Figure 3: The principles of our tracking procedure are: processing only a relevant region of the incoming frame in terms of feature extraction, matching its features not only against those of the previous frame but also against reference features with confident 3D points to avoid drift, back-projecting new features after robust pose estimation to obtain corresponding 3D coordinates for the next iteration and automatically invoking the initialization procedure in case of tracking failure.

Figure 3 shows a global overview of the tracking procedure. It always focuses two successive frames, consisting in the current frame and the previous one with its calculation results. It has additional access to the 3D model and reference data similar to features with confident 3D coordinates, which come from initialization. At every switch from initialization to tracking, the frame, for which the initial pose could be calculated adequately (consecutively called “initial frame”), is given to the tracking module as previous frame and is taken for calculating the 3D position of the tracking region. If the tracking fails because of few matches or a badly defined pose concerning residual error during pose estimation, we invoke the initialization procedure for automatically re-initializing the tracker. To deal with appearing and disappearing points, our approach bases on constant back-projection onto the CAD model. A successfully calculated pose is always employed to back-project new features for obtaining 3D coordinates, thus handling changes concerning occlusions or light differences.

The Tracking Region

During initialization we have no guess, where the projection of the target object is located within the initial frame. Although all matches, which do not lie on the target object surface, will have to be discarded again we have to process the whole image concerning feature extraction. Throughout iterative tracking we can make the assumption that the location of the target object does not change significantly between successive frames. As we know the previous pose as well as the 3D model, we get the previous location simply by projecting the model from that pose. We do not consider something like a convex hull of the projection but only a fixed-sized image region which mainly includes the target object. Furthermore we take into account that the target object is not necessarily textured equally well in all parts. We make the image region to contain the richly textured parts that deliver many features. The realization of these ideas is simple. The center of the tracking region is related to a 3D coordinate on the target object surface, which is projected into the current frame from the previous pose. The fixed-size rectangular tracking region is then simply cropped from the incoming frame centering this projection and in that connection follows the projection of the object. We automatized the process of dynamically finding a good 3D center of the tracking region after every (re-)initialization. During the first iteration of the tracking procedure we know all features and corresponding 3D coordinates of the initial frame. We simply choose the 3D center of the tracking region as one of those 3D coordinates

optimizing the overall number of features, which are included in the resulting image region. So if the size of the tracking region is chosen smaller than the current projection of target object, we get the region with most features inside even though.

Feature Matching

We match the SIFT features of the incoming frame not only with those of the previous one but also with those of the reference data set. This provides confident 3D coordinates and significantly increases the stability and precision of the tracker by avoiding drift. Let the interesting points and related features and 3D coordinates at time $t-1$ be:

$$\begin{aligned} \mathbf{m}^{t-1} &= \{\mathbf{m}_0^{t-1}, \dots, \mathbf{m}_n^{t-1}\} \\ \mathbf{D}^{t-1} &= \{\mathbf{D}_0^{t-1}, \dots, \mathbf{D}_n^{t-1}\} \\ \mathbf{M}^{t-1} &= \{\mathbf{M}_0^{t-1}, \dots, \mathbf{M}_n^{t-1}\} \end{aligned}$$

The reference data set ($\mathbf{m}^{\text{ref}}, \mathbf{D}^{\text{ref}}, \mathbf{M}^{\text{ref}}$) is defined in the same way. For each descriptor in the current frame \mathbf{D}^t we choose the one in either set \mathbf{D}^{t-1} or \mathbf{D}^{ref} that minimizes the Euclidian distance and fulfils equation (1). Now we have some matches between the corresponding 2D points:

$$\mathbf{m}^{t_i} \leftrightarrow \mathbf{m}^{t-1_j} \text{ or } \mathbf{m}^{t_i} \leftrightarrow \mathbf{m}^{\text{ref}_j}$$

As correctly matched image points are different projections of the same 3D point we associate \mathbf{M}^t with the known 3D coordinate of its match \mathbf{M}^{t-1_j} or $\mathbf{M}^{\text{ref}_j}$. Obviously it is desirable to have as much matches as possible with reference features.

Pose Estimation

Similar to initial pose estimation we obtain the current pose by first filtering the 2D/3D correspondences with the RANSAC algorithm and afterwards optimizing the reprojection errors of the remaining correspondences over the current camera rotation \mathbf{R}_t and translation \mathbf{T}_t . During iterative tracking the 2D/3D correspondences from matching are better conditioned and therefore RANSAC mainly converges after few iterations. Concerning nonlinear optimization we made two little changes in comparison with the initial pose estimation. Firstly, we take the pose of the previous frame as initial guess. Second, we use the robust TUKEY estimator [Rous87] for optimization. This estimator assigns a special weight [0...1] to each correspondence and thereby varies its influence on pose estimation. Distant outliers are weighted by zero and therefore have no influence.

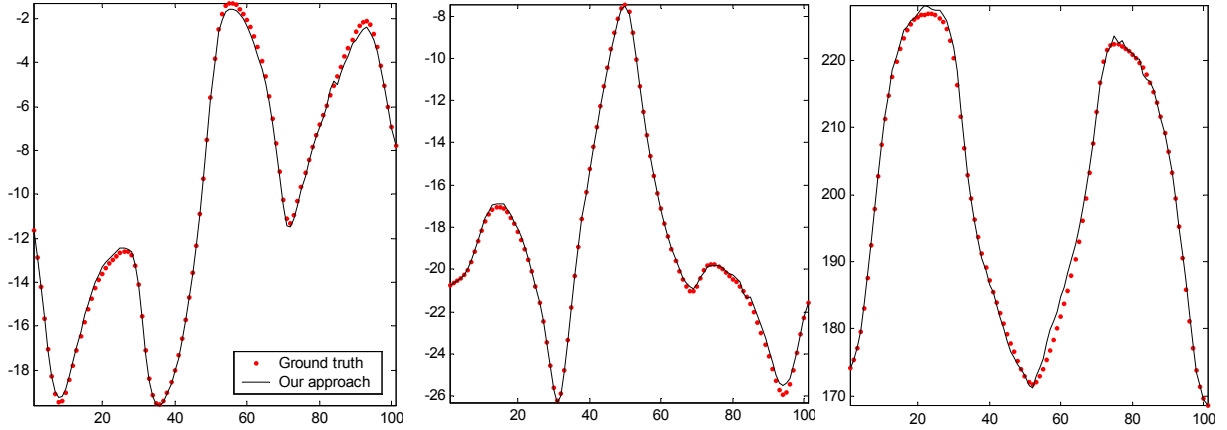


Figure 4: These plots show the precision of our method on a synthetic image sequence concerning the three coordinates of camera translation. The dots represent ground truth. Note that the reconstructed camera trajectory neither suffers from drift nor from jitter.

So equation (2) becomes

$$\min_{R_i, T_i} \sum_i \rho_{\text{TUKEY}} \left(\| \mathbf{m}_i - \mathbf{m}_i' \|^2 \right)$$

where ρ_{TUKEY} is the weighting function. Due to the former RANSAC filtering, TUKEY weights mainly stay within the upper quarter of the legal interval. But the employment of this estimator provides a more continuous camera trajectory, as the weight calculation partly depends on the previous pose.

5. EVALUTATION

We applied our method to synthetic and real data using different types of target objects like simple and more complex, planar and non-planar as well as highly and poorly textured ones. Following validation criteria are established: precision, robustness and speed. For optical verification of the current pose, the incoming frame is augmented with the axis of the world coordinate system as well as the tracked feature points. Synthetic image sequences, generated from a CAD model, provide ground truth for exact verification in terms of precision.

Synthetic Images

From a textured CAD model (see Figure 2 (c)) we rendered a 100 frames sequence with quick camera movements knowing extrinsic camera parameters for each frame except for a digitalization error. Applying our method to the synthetic images, which do not suffer from noise or radial distortion, and knowing intrinsic camera parameters exactly, we obtained very good results concerning precision (see Figure 4). The reconstructed camera translation exhibits an average Euclidian error of not more than 0.87 cm from the correct position without showing any drift or jitter. The maximum distance

has been measured with 3.2 cm mainly resulting from the z coordinate. We also measured the average reprojection error over four vertices of the CAD model with 1.24 pixels, whereat 2.83 was the maximum value.

Off-line Video Sequences

With a low-end USB web cam we captured a real image sequence wearing a HMD with the camera being fixed at it and thus simulating industrial practice. Results are shown in Figure 5 (a–f) and argue for the robustness of our method against partial occlusions (a, f) and changes of lighting conditions (b) meanwhile providing a large field of activity (c–e). We also tested our method using a package box as mainly planar target object (see Figure 5 (g–i)) and a BMW armrest as poorly textured one (see Figure 5 (j–l)).

Live Video

We applied our method to 320x240 images from both a USB web cam and a Firewire camera. Although the latter yields higher quality images in terms of noise and radial distortion, we obtained comparably stable results. With the different parameters (SIFT, size of tracking region, RANSAC iterations) being optimized for stability the system runs at real-time (19 frames/sec).

6. CONCLUSION

We presented a real-time 6 DOF camera tracking system, which includes an autonomous initialization procedure. It is designed for small environment tracking and works with different types of target objects the only restriction being, that a CAD model is available. We use 3D model information for constant back-projection of new features to be independent of partial occlusions. We perform fea-

ture matching against both the previous frame and confident reference data, thus increasing the stability of the tracker and avoiding jitter and drift. The system proves to be robust against sharp camera movements and changes of lighting conditions, which are typical for AR.

Future work includes the enhancement of pose estimation by also taking epipolar constraints between 2D/2D correspondences into account. Furthermore we want to put some effort on the improvement of feature extraction, especially by working on affine invariant representations of the image patches [Baum00, Miko02] and by searching for more efficient descriptors [Ke04].

7. REFERENCES

- [Azum95] Azuma, R. A Survey of Augmented Reality. In *Computer Graphics SIGGRAPH Proc.*, pp. 1-38, 1995
- [Baum00] Baumberg, A. Reliable feature matching across widely separated views. In *Proc. CVPR*, pages 774-781, 2000
- [Bock03] Bockholt, U., Bisler, A., Becker, M., Müller-Wittig, W.K. and Voss, G. Augmented Reality for Enhancement of Endoscopic Interventions. In *Proc. IEEE Virtual Reality Conference*, pp. 97-101, 2003
- [Comp03] Comport, A.I., Marchand, E., and Chaumette, F. A real-time tracker for markerless augmented reality. In *Proc. Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, pp. 36-45, 2003
- [DeMe92] DeMenthon, D. and Davis, L.S. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision*, pp. 335-343, 1992
- [Genc02] Genc, Y., Riedel, S., Souvannavong, F. and Navab, N. Markerless tracking for augmented reality: A learning-based approach. In *Proc. International Symposium on Mixed and Augmented Reality*, 2002
- [Harr88] Harris, C. and Stephens, M.J. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference*, Manchester, pp. 147-151, 1988
- [Hart00] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000
- [Ke04] Ke, Y. and Sukthankar, R. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *CVPR*, 2004
- [Lau00] McLauchlan, P. A Batch/Recursive Algorithm for 3D Scene Reconstruction. In *Proc. CVPR*, 2000
- [Lepe03] Lepetit, V., Vacchetti, L., Thalmann, D. and Fua, P. Fully Automated and Stable Registration for Augmented Reality Applications. In *Proc. Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2003
- [Lowe99] Lowe, D.G. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision (ICCV)*, Corfu, pp.1150-1157, 1999
- [Miko02] Mikolajczyk, K. and Schmid, C. An affine invariant interest point detector. In *ECCV*, 2002
- [Poll99] Pollefeys, M., Koch, R. and Van Gool, L. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters. In *International Journal of Computer Vision*, pp. 7-25, 1999
- [Rous87] Rousseeuw, P. and Leroy, A. *Robust Regression and Outlier Detection*. Wiley, 1987
- [Shi94] Shi, J. and Tomasi, C. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, pp. 593-600, 1994
- [Spie00] Spiess, H. and Ricketts, I. Face Recognition in Fourier Space. In *Vision Interface*, Montreal, pp. 38-44, 2000
- [Stri01] Stricker, D. Tracking with Reference Images: A Real-Time and Markerless Tracking Solution for Out-Door Augmented Reality Applications. In *Proc. of VAST*, 2001
- [Thom97] Thomas, G.A., Jin, J., Niblett, T. and Urquhart, A. A versatile camera position measurement system for virtual reality TV-production. *International Broadcasting Convention*, IEEE Conference Publication, pp.284-289, 1997
- [Turk91] Turk, M. and Pentland, A. Eigenfaces for Recognition. In *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86, 1991
- [Vass02] Vlahakis, V., Ioannidis, N., Karigiannis, J., Tstros, M., Gounaris, M., Stricker, D., Gleue, T., Dähne, P. and Almeida, L. *Archeoguide: An Augmented Reality Guide for Archaeological Sites*. In: *IEEE Computer Graphics and Applications* Vol. 22, No. 5, pp. 52-60, 2002
- [Wesa04] Wesarg, S., Firl, E., Schwald, B., Seibert, H., Zogal, P. and Roeddiger, S. Accuracy of Needle Implantation in Brachytherapy Using a Medical AR System - a Phantom Study. In *SPIE Medical Imaging Symposium*, pp. 341-352, 2004

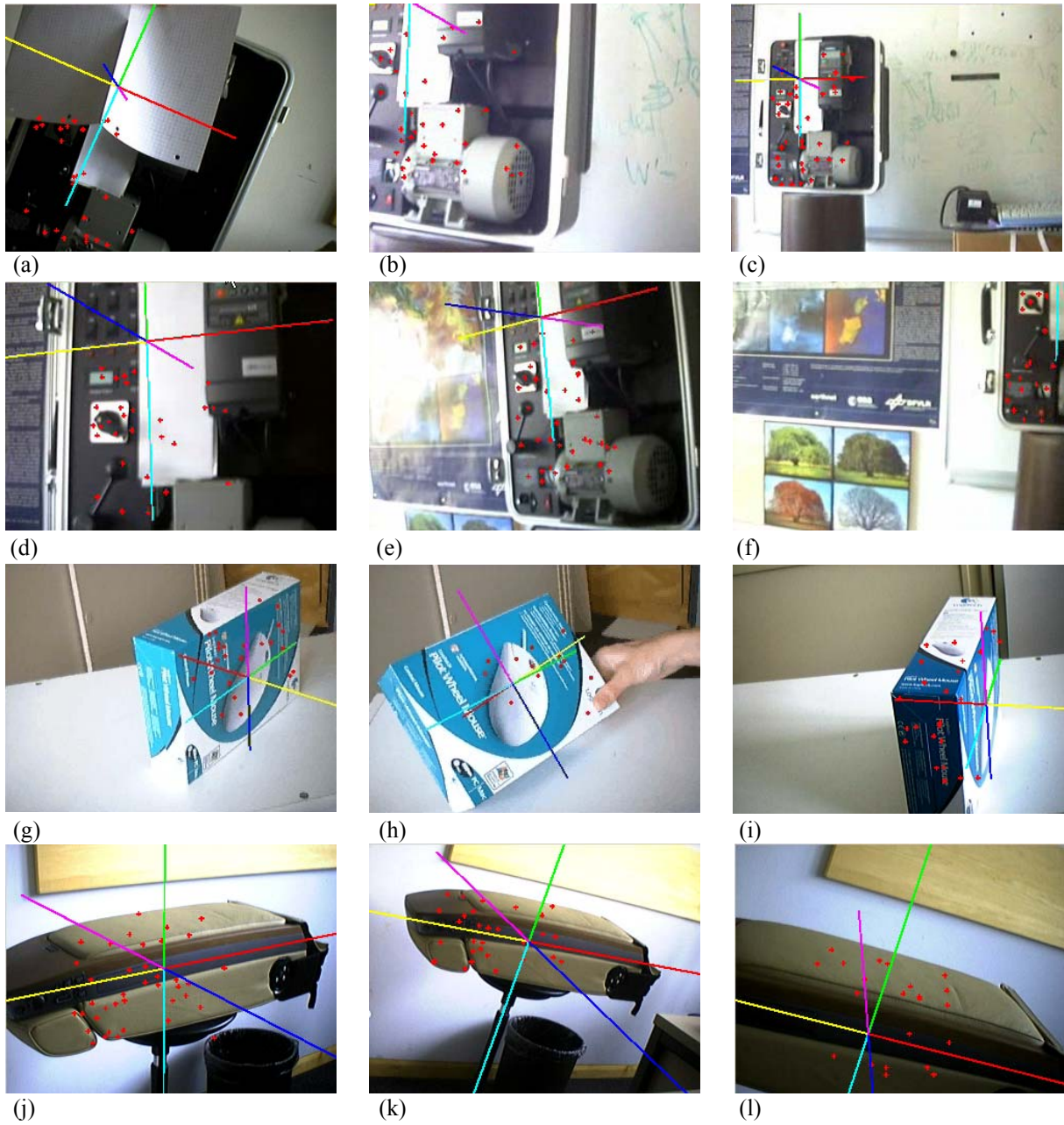


Figure 5: Results of our tracking procedure using different types of target objects (planar/non-planar, highly/poorly textured, simple/complex).