Marker-less Tracking for AR: A Learning-Based Approach

Y. Genc S. Riedel F. Souvannavong* C. Akınlar N. Navab

Real-time Vision and Modeling Department
Siemens Corporate Research
Princeton, NJ 08540, USA
Yakup.Genc@scr.siemens.com

Abstract

Estimating the pose of a camera (virtual or real) in which some augmentation takes place is one of the most important parts of an augmented reality (AR) system. Availability of powerful processors and fast frame grabbers have made vision-based trackers commonly used due to their accuracy as well as flexibility and ease of use.

Current vision-based trackers are based on tracking of markers. The use of markers increases robustness and reduces computational requirements. However, their use can be very complicated, as they require certain maintenance. Direct use of scene features for tracking, therefore, is desirable. To this end, we describe a general system that tracks the position and orientation of a camera observing a scene without any visual markers. Our method is based on a two-stage process. In the first stage, a set of features is learned with the help of an external tracking system while in action. The second stage uses these learned features for camera tracking when the system in the first stage decides that it is possible to do so. The system is very general so that it can employ any available feature tracking and pose estimation system for learning and tracking. We experimentally demonstrate the viability of the method in real-life examples.

1 Introduction

Augmented reality (AR) is a technology in which a user's perception of the real world is enhanced with additional information generated from a computer model. The visual enhancements may include labels, three-dimensional rendered models, and shading and illumination changes. AR allows a user to work with and examine the physical world, while receiving additional information about the objects in it through a display.

In a typical AR system, a user's view of a real scene is augmented with graphics. The graphics are generated from geometric models of both virtual objects and real objects in the environment. In order for the graphics and the scene to align properly, the pose and optical properties of the real and virtual cameras must be the same.

Estimating the pose of the camera (virtual or real), in which some augmentation takes place, is the most important part of an AR system. This estimation process is usually called tracking[1]. Many different tracking methods are available (see [7, 14, 2] for review of tracking systems for augmented reality and [4] for a review of tracking systems in general) including mechanical, magnetic, ultrasound, inertial, vision-based, and hybrid systems that try to combine the advantages of two or more technologies.

Availability of powerful processors and fast frame grabbers have made vision-based trackers

*F. Souvannavong is currently with Institut Eurecom, Sophia Antipolis, France.

[1]Virtual and augmented reality (VR and AR) research communities use the term "tracking" in a different context than the computer vision community. "Tracking" in AR and VR refers to determining the pose, i.e., three-dimensional position and orientation, of the camera. "Tracking" in computer vision means data association, also called matching or correspondence, between consecutive frames in an image sequence.

a common choice amongst many other technologies mostly due to their accuracy as well as flexibility and ease of use. Although very elaborate object tracking techniques exist in computer vision (e.g., [5] provides fast and robust object tracking in video streams), they are not practical for pose estimation. The vision-based trackers used in AR are based on tracking of markers (see [15, 11, 26]). The use of markers increases robustness and reduces computational requirements. However, their use can be complicated as they require certain maintenance. For example, placing a marker in the workspace of the user can be intrusive and the markers can from time to time need re-calibration.

Direct use of scene features for tracking instead of the markers is much desirable, especially when certain parts of the workspace do not change in time. For example, a control panel has fixed buttons and knobs that remain the same over its lifetime. The use of these rigid and unchanging features for tracking simplifies the preparation of the scenarios for scene augmentation as well.

Attempts in the past at solving this problem remained limited in their aims. They are sometimes used for increasing the accuracy and the range of the tracking in the presence of a marker based tracking system or in combination with other tracking modalities (hybrid systems).

We describe a general system that tracks the position and orientation of a camera observing a scene without any visual markers. The method is based on a two-stage process. In the first stage, a set of features is learned with the help of an external tracking system while in action. The second stage uses these learned features for camera tracking when the system in the first stage decides that it is possible to do so.

This paper is organized as follows. Section 2 describes the problem we are addressing and states the related work. While Section 3 gives the details of the proposed method, Section 4 describes the system built for experiments whose results are given in Section 5. Summary and conclusions are provided in Section 6.

2 Background

We define the tracking problem for AR as the task of estimating the 6 DOF pose of an object, e.g., the head-mounted display, in a given coordinate system. This differs from the computer vision problem of tracking objects in video sequences which can be described as data association between consecutive frames.

Many tracking technologies have been used in AR applications. These include mechanical [23], magnetic [12], ultrasound, inertial, vision-based and hybrid trackers. Vision-based trackers come in many different forms. Some of these uses a mobile camera to track a set of markers in the visible spectrum [15, 11, 26], and some tracks retroreflective markers in the infrared spectrum [18]. More involved systems use stationary cameras to track markers attached to objects, e.g., POLARIS. Hybrid systems [21, 3, 1] try to combine the advantages of different tracking modalities.

Attempts to use scene features other than specially designed markers have been made in the literature. Most of these were limited to either increasing the accuracy of other tracking methods or to extend the range of tracking [17, 16]. [20] discusses a tracking system that looks for planar structures in the scene. Other methods that are based on image matching techniques have also been proposed [22].

Work in computer vision has yielded very fast and robust methods for object tracking (e.g., [5]). However, none of these is particularly useful for accurate pose estimation required by most AR applications. Pose estimation requires a match between a three-dimensional model and its image [6]. Object tracking does not necessarily provide such a match between the model and its image. Instead, it provides a match between the consecutive views of the object.

The method, proposed in this paper, suggests a general method for feature-based pose estimation in video streams. It differs from the existing methods in several ways. First, the proposed method is a two stage process. The system first learns and builds a model of the scene using off the shelve pose and feature tracking methods. After this learning process, tracking for pose is achieved

by tracking these learned features.

The second difference is attributed to the way the first stage (training or learning) works. The outcome of the learning process is a set of three-dimensional features with some associated uncertainties. This is not achieved by a structure-from-motion but a triangulation or bundle adjustment process. Therefore, it yields more stable and robust features that can be used for accurate pose estimation.

Finally, an advantage of our method over the model based ones is that it can use features on the textures and highlights. These are not very easy to model even if a three-dimensional model of the workspace is available. More importantly, the details of the model may not be particularly suited for the application at hand. Our method builds an implicit model using only the most salient features observable in the given context.

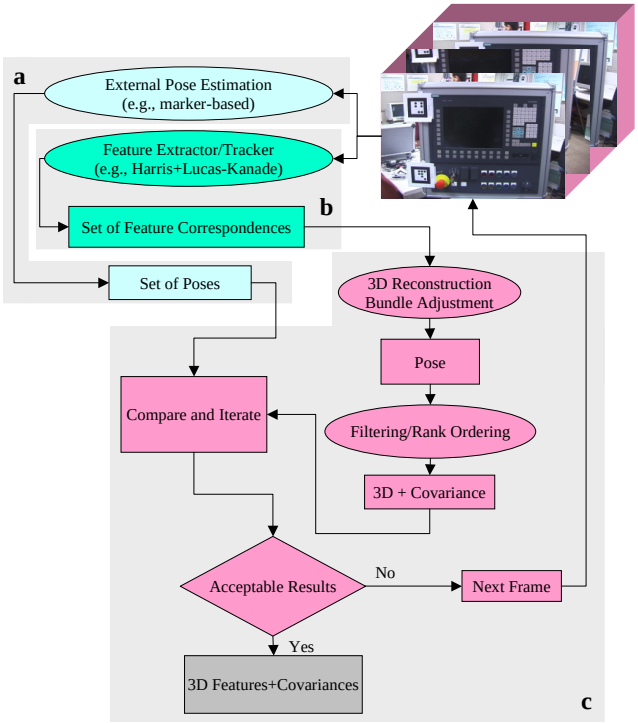

Figure 1: The learning or training phase of the proposed system is depicted. This phase contains three subsystems, i) an external tracker (the shaded region labeled "a"), ii) a feature extractor and tracker (the shaded region labeled "b"), and iii) the trainer (the shaded region labeled "c").

3 System Definition

The proposed vision-based marker-less tracking system aims at the use of real scene features for estimating the pose of a camera. The solution allows the user to move from using markers or any applicable tracking and pose estimation methods to using real scene features through an automatic process. This process increases the success of the overall registration accuracy for the AR application. The basic idea is to first use the markers or any applicable tracking device for pose and motion estimation. The user could start using the system in his or her usual environment. As the user works with the current system an automated process runs in the background. This process remains hidden until the feature-based system decides to take over the pose estimation task from the other tracker. The take over happens only after enough number of salient features are learned and the pose obtained from these are as good as the one provided by the external tracker. The automated process has two phases, i.e., (i) learning, and (ii) tracking for pose estimation.

3.1 Learning

For a vision-based tracking system, a model is needed which is matched against the images for estimating the pose of the camera taking the images. In the proposed method, we use an automated process to learn the underlying model of the workspace where the tracking is to take place.

The general idea of learning or training is presented in Figure 1. While the AR system together with another tracking system is in use, the system uses any available feature extraction and tracking methods to detect reliable features[2]. These may include basic features such points, lines, circles and planar patches or composite features such as polygons, cylinders etc. The system tracks each feature in the video stream. Once a feature is reasonably tracked over a number of frames, the system uses the 6 DOF pose provided by the existing tracking system to obtain a 3D model for this particular feature. At this point the feature tracking,

[2]Depending on the performance of the system, this can be done in real time or on recorded videos along with the pose as provided by the external tracker system.

for this particular feature, becomes a mixed 2D-2D and 3D-2D matching and bundle adjustment problem. The system evaluates each set of feature correspondences in order to define whether this feature is a stable one, which means that:

- Over time the 3D feature does not move independently from the observer (i.e., static position in the world coordinate system),
- The distribution of the intensity characteristics of the feature does not change significantly over time,
- The feature is robust enough that the system could find the right detection algorithm to extract it under the normal changes in lighting conditions (i.e., changes which normally occur in the workspace),
- The feature is reconstructed and back-projected, using the motion estimated by the external tracker, with acceptable back-projection error,
- The subset of the stable features chosen need to allow accurate localization, compared to the ground truth from the external tracker.

3.2 Tracking for Pose Estimation

Once a model is available, the marker-less tracking system uses the available feature extractors and trackers to extract features and match them against the model for the initial frame and then track them over consecutive frames in the stream. This process is depicted in Figure 2.

Once the tracking system has been initialized, i.e., the pose for the current frame is known approximately, it can estimate the pose for consecutive frames. This estimation is very fast and robust since it uses the same feature-tracking engine as in training and under similar working conditions.

Initial model matching can be done by an object recognition system. This task does not need to be real-time, i.e., a recognition system that can detect the presence of an object with less than 1fps speed can be used. Due to the fact that the environment is very restricted, the recognition system can be engineered for speed and performance.

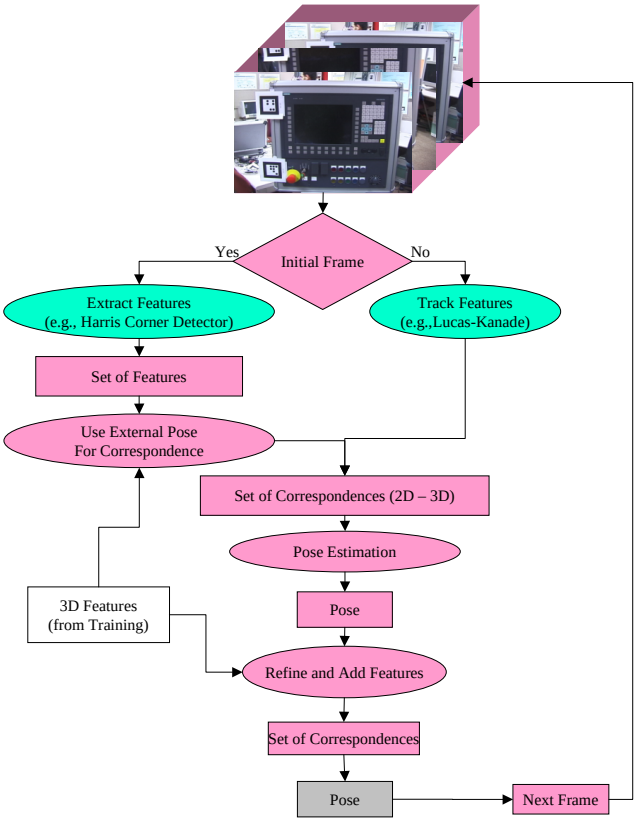

Figure 2: The tracking process in the proposed system is depicted. Similar to training, this phase contains two subsystems, i) a feature extractor, and ii) the marker-less tracker.

4 Implementation

We have implemented the system described in the previous section. The system consists of (i) an external tracker, (ii) a feature tracker, (iii) a model builder, (iv) a pose estimator, and (v) an augmentation engine. This section describes the details of our implementation and the choices that we have made.

External Tracker: We have used the marker-based tracking system[3] described in [26]. This tracker returns 8 point features per marker. Once calibrated in 3D, these points are used to compute the 6 DOF pose for the camera (Tsai [25]).

[3]In this particular implementation, the same images coming from the tracker camera are used both by the external tracker and the learning system. This readily solves the synchronization issue between the camera and the external tracker.

Feauture Tracker: For simplicity, our system only considers point features in tracking. For this, a pyramidal implementation of Lucas-Kanade algorithm [13] is used (with the pyramid depth as 3 and the search window of the optical flow as 10x10). The tracked features are initially selected with the Shi-Tomasi algorithm [19].

Model Building: Using the points tracked by the system and the pose provided by the external tracker, the system performs an initial reconstruction of the 3D positions of these points using triangulation [10]. A RANSAC type of process [8] is implemented to eliminate points and frames that may be outliers. This is followed by a bundle adjustment process (see [24] for a recent review of bundle adjustment) allowing a better estimate of the point locations as well as their uncertainties. This uses a selected number of frames to process.

Pose Estimation: Given the 2D-3D point matches, the pose of the camera is computed using the algorithm by Tsai [25]. An internal calibration is performed for the camera before the training to account for radial distortion up to 6th degree.

Augmentation Engine: In order to show the results, we have implemented a display engine which overlays line segments representing the virtual objects in wire-frame. Each line is represented by its two end points. After the two endpoints of a line are projected, a line connecting the two projected point is drawn on the image. In the presence of radial distortion, this will present a one-to-one registration between the vertices of the virtual model and their images. However, the virtual line and the image of the corresponding line will not match. One can correct the distortion in the image so that the virtual line matches exactly with the real one. Yet, since one of the aim of the augmentation engine is to visually demonstrate how good the pose is with respect to the image quality, no radial distortion correction in the image is performed.

5 Experiments

We have conducted extensive experiments to validate our method on real data sets. This section provides the details of these experiments and the results.

The first set of experiments test the learning or training part of the system. We have used a Sony™ DV camera and obtained several sets of video sequences of our workspace where tracking is to take place. Our workspace (see Figure 3) includes a cabinet and a control panel. Each video sequence is captured under the real working conditions of the target AR application.

Figure 3: The experimental workspace including a control panel and a cabinet where the tracking is to take place.

We have used a marker-based tracker as the external tracker. The external tracker provides the ground truth tracking information to the learning system. In particular, we have used the markers described in [26]. A set of these markers are placed in the workspace (see Figure 4). They are then calibrated using a photogrammetry process with high resolution digital pictures.

Once the markers are calibrated, i.e., their positions are calculated, all of the cameras used in the experiments are internally calibrated using these markers. We use Tsai's method [25] to allow radial distortion correction up to 6th degree, which ensures a very good pose estimation for the camera when the right correspondences are provided.

As explained earlier, while the external tracking provides the AR system with the 6 DOF pose, the

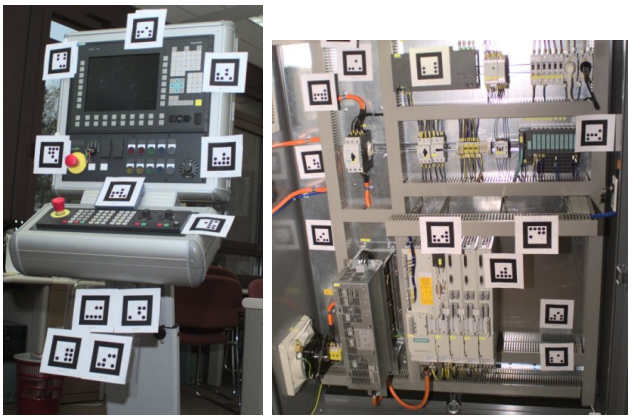

Figure 4: The placement of the markers in the workspace provides ground truth pose for the learning process. The marker positions are calibrated using photogrammetry techniques with high resolution images. The system starts with a dense set of markers for a good calibration. The markers are then removed while the system learns more and more features for marker-less tracking.

learning process extracts and tracks features in the video stream and reconstructs the position of the corresponding scene features. The 3D position is computed using the pose provided by the external tracker. The system lets the user to choose a certain portion of the image. The scene features only in the corresponding region are reconstructed. The rest of the process is automated. Figure 5 shows the initial frame and tracked features on a later frame in the video.

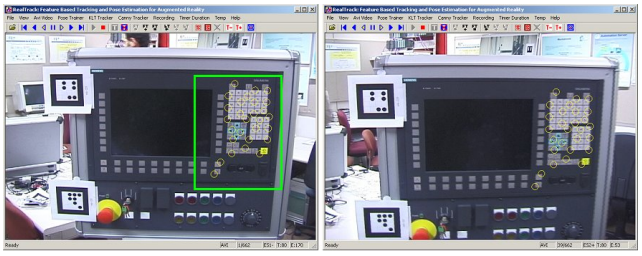

Figure 5: Training takes place with the help of a marker-based tracker and a feature tracker. The system lets the user to select a portion of the scene in the image. The underlying model of this region is reconstructed along with its uncertainty.

Figure 6 shows some example results from the learning process. After tracking a set of features in about 100 frames, the system yields to a set of reconstructed 3D points. Two views of the combined set of the 3D points are displayed in this figure. In order to provide a reference, three wire-frame boxes are inserted in the scene, which are placed on the two faces of the panel.

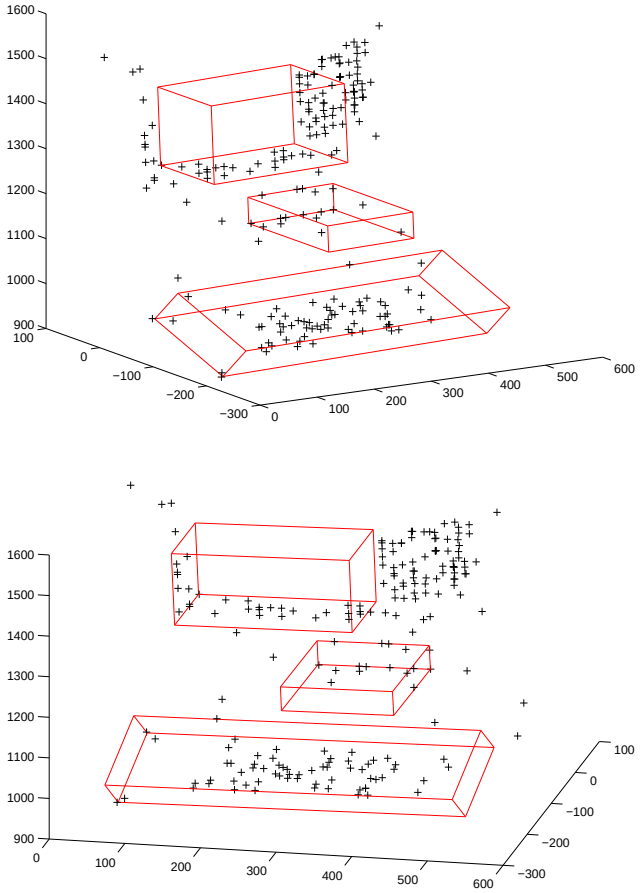

Figure 6: Sample results from the training process. Two 3D views of the reconstructed 3D points are shown along with three wire-frame boxes as reference. See Figures 7-9 for a superimposition of these on to the images of the panel.

In order to quantify the reconstruction power of our learning process, we have grouped the learned feature points into two major coplanar sets. Within each coplanar sets, the deviation from planarity is measured. In each case, less than 3% deviation is observed compared to the largest size of the panel surface. Since the motion in the

training video was mostly frontal, the largest variation was observed in the z-coordinate of the point (for reference, we take the upper panel surface as the xy-plane).

After the system has learned enough salient features, marker-less tracking is started. To test the accuracy and robustness of the tracking part of our system, we have conducted three major experiments. In all of these experiments we have assumed that the pose for the first frame of the sequence where the tracking is to take place is known. In most of these experiments, we let a RANSAC type of process determine the correspondences for the initial pose estimation. In some cases, we set the correspondences for the first frame by hand. In all cases, only three point correspondences are needed to estimate the initial pose using the three-point algorithm described in [9].

The first experiment is designed to measure the effect of the accuracy of the initial estimate of the pose on the marker-less tracking method. For this, we have run our tracker on a video sequence of about 600 frames where the pose for the first frame is computed using the markers in the scene. We then perturbed the pose of the first frame by about 10% for each dimension (3 components of the translation, and 3 components of the rotation as represented by the Euler angles). We ran our tracker using this initial pose on the same video sequence. The left column in Figure 7 shows the exact pose for the first frame, and tracking results for the consecutive frames. The right column shows the perturbed pose for the first frame and the tracking results in the consecutive frames. As can be seen from these results, even when starting with a wrong pose, the feature-based tracking algorithm converges to the same "good" pose that is estimated using the correct pose for the first frame. Similar results were obtained on other video sequences as long as 1800 frames.

The second set of experiments is conducted to see if tracking can be achieved using cameras other than the one used in training. Figure 8 shows the results obtained using a Sony™ XC55BB black-and-white camera. This camera is internally calibrated as explained above. We obtained more than 5 video sequences using this camera (on

Figure 7: Tracking results using the exact (left column) and perturbed (right column) initial pose for the first frame in the sequence of 600 frames. The estimated pose is used to superimpose the wire-frame boxes in the images. Note that the visible markers are not used during tracking. Only the learned features are used in the context of marker-less tracking. See the text for details.

the average about 1000 frames with considerable change in the view points). After initialization of the pose for the first frame, we let our marker-less tracker track the learned features. Some sample results are shown in Figure 8. Even with a very different tracker and learning camera, the system yields very good pose during tracking. High radial distortion due to larger field-of-view does not effect the accuracy and performance of the marker-less tracking system.

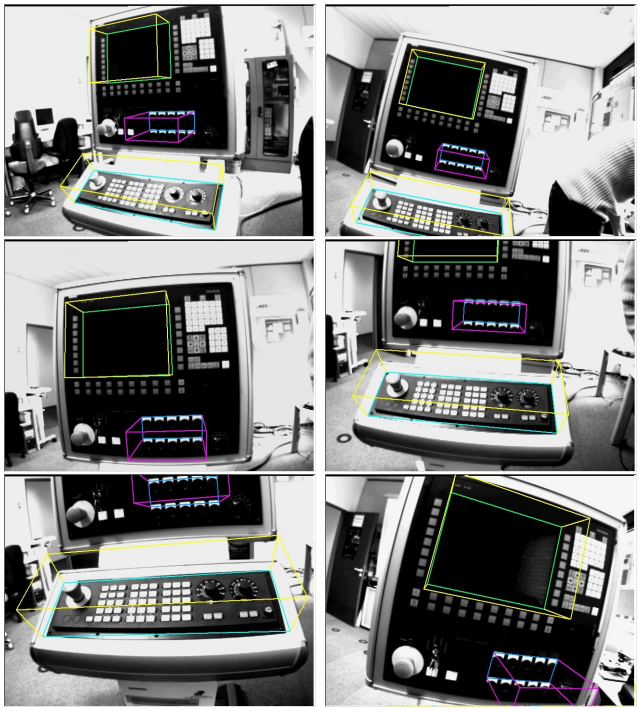

Figure 8: Tracking results using a black and white CCD camera. This camera is completely different from the one used for training the system. Even with very low image quality in black-and-white, the tracker still works very well.

The last set of experiments is conducted to show that the tracking and pose estimation is quite robust even in the presence of the non-rigid moving objects occluding the learned features. Figure 9 shows the superimposition of the wire-frame boxes using the pose obtained from the marker-less tracker for different frames in a video stream. The tracker is quite robust against occlusions caused by both rigid and non-rigid objects.

Finally, we provide some results of running time

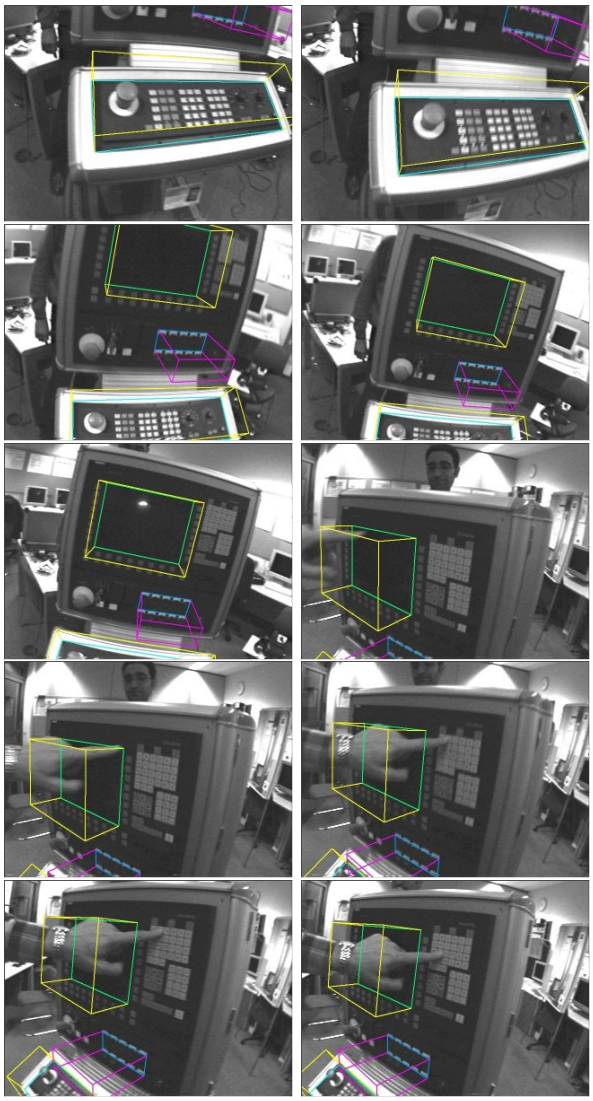

Figure 9: Tracking results when a non-rigid object occludes the trained features.

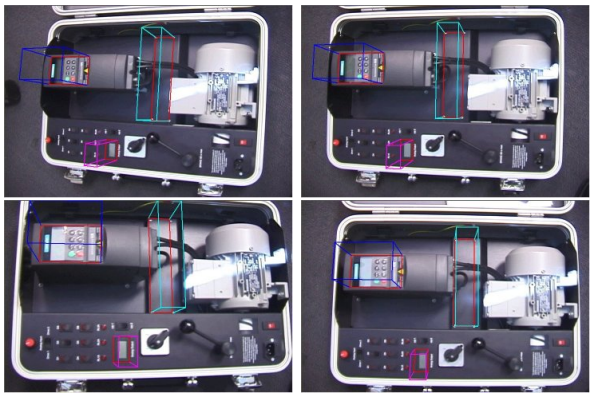

Figure 10: Tracking results for another object using the Sony™ DV camera.

performance of our method. We have run the learning part of our system off-line. This process is very computationally intensive and does not need to be on-line. The marker-less tracking part of our system runs close to full frame rate (about 22fps) on a 2GHz Intel Pentium™ III processor. This is achieved when a 640×480 video stream is captured from a black-and-white camera through an off-the-shelf frame grabber, e.g., FALCON™ from IDS. When a lower resolution video stream is tracked, e.g., 320×240, the frame rate goes well over 30fps. The processing time may increase slightly depending on the size of the learned-feature set.

6 Conclusions

We have presented a complete system that can track in real-time the position and orientation of a camera observing a scene. The system first learns the scene structure by utilizing an external tracking system, e.g., a marker-based tracker or a magnetic tracker. This training step results in an implicit model of the three-dimensional scene. This model includes the scene coordinates of salient features as well as their uncertainties. Once the model is learned, the system computes the pose of the camera observing the scene in real-time. Feature tracking is done by utilizing any available module that tracks features such as corners.

Experimental results showed that the method is quite robust even in the presence of moving non-rigid objects occluding the actual scene. Moreover, with an off the shelf computer, the tracking and pose estimation can be done in real time, i.e., 30fps.

Our plans for future work include incorporating this tracking system with a recognition system which can estimate an initial pose. A slower (about 1fps) process for approximate pose determination can be used to re-initialize the tracking system to recover the tracker when it fails.

Further improvements will be sought to improve the real-time performance of the system which may include processing lower resolution images. At the learning side, other features such as lines will be explored for tracking. We are also planning to use our tracker on an optical see-through HMD system for calibration as well as tracking.

Acknowledgements

We would like to thank Siemens A&D for supporting our research efforts in AR and real-time tracking and for providing the data used in our experiments. We would also like to thank M. Neuberger and X. Zhang for helping us calibrate the markers and V. Ramesh, D. Comaniciu and B. Bascle for very useful discussions during the course of this research.

References

[1] T. Auer and A. Pinz. Building a hybrid tracking system: Integration of optical and magnetic tracking. In *Proc. IWAR*, pages 13–22, San Francisco, CA, USA, October 1999.

[2] R. Azuma. Tracking requirements for augmented reality. *Communications of the ACM*, 36(7):597–620, July 1993.

[3] W. Birkfellner, F. Watzinger, G. Enislidis, M. Truppe, R. Ewers, and H. Bergmann. Concepts and results in the development of a hybrid tracking system for computer aided surgery". In *MICCAI '98*, pages 343–351, 1998.

[4] J. Borenstein, H.R. Everett, and L. Feng. *Navigating Mobile Robots: Sensors and Techniques.* A.K. Peters, Ltd., Wellesley, MA, 1996.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, 2000.

[6] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[7] F. J. Ferrin. Survey of helmet tracking technologies. In *SPIE Vol. 1456: Large-Screen Projection, Avionic and Helmet-Mounted Displays*, pages 86–94, 1991.

[8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–385, 1981.

[9] R.M. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. In *In Proc. CVPR*, pages 592–598, Maui, Hawaii, USA, 1991.

[10] Richard I. Hartley and Peter Sturm. Triangulation. *CVIU*, 1996.

[11] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. IWAR*, San Francisco, CA, USA, October 1999.

[12] M. A. Livingston and A. State. Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments*, 6(5):532–546, October 1997.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int. Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[14] K. Meyer, H.L. Applewhite, and F.A. Biocca. A survey of position trackers. *Presence: Teleoperators and Virtual Environments Vol. 1, No. 2*, pages 173–200, August 1992.

[15] U. Neumann and Y. Cho. A selftracking augmented reality system. In *Proceedings of the ACM Symposium on Virtual Reality and Applications*, pages 109–115, July 1996.

[16] U. Neumann and S. You. Natural feature tracking for augmented reality. *IEEE Transactions on Multimedia*, 1(1):53–64, March 1999.

[17] J. Park, S. You, and U. Neumann. Natural feature tracking for extendible robust augmented realities. In *International Workshop on Augmented Reality*, 1998.

[18] F. Sauer, F. Wenzel, S. Vogt, Y. Tao, Y. Genc, and A. Bani-Hashemi. Augmented workspace: Designing an AR testbed. In *International Symposium for Augmented Reality*, pages 165–174, Munich, Germany, October 2000.

[19] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, June 1994.

[20] G. Simon, A.W. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *International Symposium for Augmented Reality*, 2000.

[21] A. State, G. Hirota, D.T. Chen, B. Garret, and M. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *SIGGRAPH '96*, pages 429–438, New Orleans, LA, August 1996.

[22] D. Stricker and T. Kettenbach. Real-time and markerless vision-based tracking for outdoor augmented reality applications. In *ISAR*, 2001.

[23] I.E. Sutherland. A head-mounted three-dimensional display. In *Fall Joint Computer Conference*, pages 757–764, 1968.

[24] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. In *Proc. of Workshop on Vision Algorithms*, 1999.

[25] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[26] X. Zhang and N. Navab. Tracking and pose estimation for computer assisted localization in industrial environments. In *WACV*, pages 214–221, Palm Springs, CA, December 2000.