Proc. of the IS&T/SPIE Conf. on Visual Communications and Image Processing (2007)

Particle filter-based camera tracker fusing marker and feature point cues

David Marimon, Yannick Maret, Yousri Abdeljaoued and Touradj Ebrahimi

Signal Processing Institute
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland

ABSTRACT

This paper presents a video-based camera tracker that combines marker-based and feature point-based cues within a particle filter framework. The framework relies on their complementary performances. On the one hand, marker-based trackers can robustly recover camera position and orientation when a reference (marker) is available but fail once the reference becomes unavailable. On the other hand, filter-based camera trackers using feature point cues can still provide predicted estimates given the previous state. However, the trackers tend to drift and usually fail to recover when the reference reappears. Therefore, we propose a fusion where the estimate of the filter is updated from the individual measurements of each cue. More precisely, the marker-based cue is selected when the reference is available whereas the feature point-based cue is selected otherwise. Evaluations on real cases show that the fusion of the two approaches outperforms the individual tracking results.

Keywords: vision-based registration, camera tracking, particle filters, marker-based tracking, feature points.

1. INTRODUCTION

Camera tracking has brought the attention of many researchers due to the large range of possible applications. Example areas of application are simultaneous localisation and mapping (SLAM),[1] structure from motion (SfM),[2] and augmented/mixed reality (AR/MR).[3] In order to register the scene, the orientation and position of the camera have to be known. Tracking of camera motion is known as the estimation of the transformation between the camera coordinate frame and that of the world.

Fusion of tracking techniques have proven to be useful for some applications (for instance, augmented reality and object tracking). The objective of fusing is to build a synergy out of its components. To achieve this, techniques with complementary performance have first to be identified.

Among the various solutions for camera tracking, video-based ones offer more flexibility while providing enough accuracy. Video-based camera tracking can be classified into two categories that have compensated weaknesses and strengths: bottom-up and top-down approaches.[4] For the first category, the six degrees of freedom (3D position and 3D orientation) estimate is obtained from low-level 2D features and their 3D geometric relation (such as homography, epipolar geometry, CAD models or patterns), whereas for the second group, the 6D estimate is obtained from top-down state space approaches using motion models and prediction.

Marker-based systems[5] can be classified in the first group. Indeed, they use the geometric properties of a particular pattern in the scene (such as the size of a square or the distance between centres of circles) and their 2D projection. They have proven to be an easy-to-use and attractive tool for camera tracking, as they require a very limited set-up (depending on the application). Although marker-based systems have a high detection rate and estimation speed, marker-based systems still lack tracking robustness, as the marker(s) must be visible at all time, limiting the user actions.

In contrast with bottom-up approaches, top-down techniques such as *filter-based camera tracking* allow track continuation even when the reference is temporarily unavailable. This is a desirable characteristic for applications where the user is allowed to move around freely. A filter maintains a time-coherent estimation of the position

e-mail: {david.marimon, yannick.maret, yousri.abdeljaoued, touradj.ebrahimi}@epfl.ch

and orientation of the camera. This filter relies on predictive motion models that are updated as long as the reference is visible. In most cases, the reference consists in feature points (salient points such as corners or edges). Examples of this type of tracking can be found in,[1],[6] and more recently.[7] In general, the disadvantage of top-down techniques lies in the drift that appears during the absence of a stable reference (usually due to feature points difficult to recognise after perspective distortions).

In this paper, we present a camera tracking system where a particle filter keeps track of the six degrees of freedom of the camera. It fuses the measurements coming from a marker-based cue (MC) and a feature point-based one (FPC). The MC is provided by the marker-based tracking system contained in ARToolkit.[8] The FPC is based on the work presented by Davison,[1] which estimates camera pose using a Kalman filter updated through feature points. The particularity of the proposed fusion filter is to manipulate different sorts of cues within a single framework. More precisely, the framework has a single motion model and its prediction is corrected by one cue at a time (the FPC is used when the MC fails to detect a marker). The filter's state is updated by switching between two different likelihood distributions, each adapted to the measurement (cue) type. As a matter of fact, the fusion is applicable to other sort of trackers, just by changing the likelihood distribution. This flexibility is one of the advantages of the fusion framework proposed.

The paper is structured as follows. Section 2 describes previous related research. The techniques involved in the fusion and the proposed tracker are presented in Section 3. Several experiments and results are given in Section 4. Conclusions and future research directions are finally exposed.

2. RELATED WORKS

Systems that combine diverse tracking techniques (known as *hybrid tracking systems*) have shown that the overall performance is enhanced by merging complementary techniques.[9] Usually the combination merges sensors having best performance at slow or at fast motion. During the last decade, interest has grown on combining inertial sensors (gyros and accelerometers) and video. Representative works are those presented by Azuma and Bishop,[10] and by You *et al.*[11]

Various researchers have selected marker-based systems as video trackers to fuse with inertial sensors. Kanbara *et al.*[12] used a stereoscopic registration system. The algorithm is able to estimate the 3D position of markers even when they are outside the field of view, thanks to the inertial sensor. You and Neumann[13] used similar trackers and presented a fusion based on a Kalman filter. Naimark and Foxlin[14] presented a framework using circular markers. The fusion framework is based on the estimate of the inertial sensor, which reduces the search area for the marker, and vision is only used for small corrections.

Despite the enhancements provided by fusions of different types of sensors, it is also possible to merge trackers within the same modality. This is the case for video-based techniques. Although little work has targeted camera tracking applications, several researchers have identified the potential of video-based tracking fusion.[4,15,16] Okuma *et al.*[4] fuse data from a single camera. The system switches between a model-based tracker and a particle filter-based tracker, similar to that of Pupilli and Calway.[7] Nonetheless, this framework takes limited advantage of the filtering framework and needs still the assistance of an inertial sensor. Satoh *et al.*[15] described a registration system with a camera sensor attached to a Head Mounted Display (HMD). Feature point-based tracking is used with this sensor. In order to produce more robust estimates, this mobile sensor is assisted with a bird's eye view camera set up in a position where it can always locate the mobile camera. A marker-based system is used to track the HMD. Limited attention is given to solving the initialisation phase, which is done each time manually by moving to a precise known pose. Najafi *et al.*[16] investigated this problem and described an automated initial registration system using a similar setup. A stationary camera is used to estimate the position of a mobile camera. For the stationary camera, gradient images are used to roughly estimate the pose of the mobile camera. The mobile camera performs model-based tracking, exploiting the estimate of the stationary one.

Our work is close to the approach of Okuma as fusion is performed within a single camera setup. However, we pay more attention to the filtering framework and design it so as to achieve more tracking robustness.

3. SYSTEM DESCRIPTION

The goal of the fusion system is to track the pose of the camera using a fiducial marker and to achieve robust tracking despite occlusions and illumination variations. The fusion uses two cues, the marker-based (MC) and the feature point-based (FPC), merged in a particle filter.

This section describes how the marker-based and the feature point-based cues are obtained as well as the procedure used to fuse them within a particle filter.

3.1. Marker-based cue (MC)

Geometric patterns such as a group of circles (minimum set of three) or a square, provide enough information to recover the pose of a camera capturing them. Among the various existing marker-based tracking techniques, we have selected ARToolkit[8] because of its high detection rate and estimation speed. This marker-based tracker and some of its variants, are at the core of many applications.[17]

The marker-based tracker produces an estimate of the camera pose, provided that the geometry of the marker is known. This estimate is one of the cues (identified as MC) given to the fusion framework. More precisely, the marker-based tracker is employed to compute the transformation

$$T = [t_X, t_Y, t_Z, rot_W, rot_X, rot_Y, rot_Z], \tag{1}$$

where t are the translations (3D position) and rot is the quaternion for the rotation (3D orientation), between the world coordinate frame (fixed to the fiducial marker) and that of the camera. The world coordinate frame is . This transformation is fed into the filter for state update (See Section 3.3).

The MC is obtained as described next. At each frame, the algorithm searches for square markers inside the field-of-view (FoV). If a marker is detected, the transformation can be computed. The detection process works as follows. First, the frame is converted to a binary image (using a fixed threshold) and the black marker contour is identified. If this identification is positive, the 6D pose of the marker relative to the camera (T) is calculated. This computation uses the geometric relation of the four projected lines that contour the marker in addition to the recognition of a non-symmetric pattern inside the marker.[18] When this information is not available, no measurement can be calculated. This occurs in the following cases:

- markers are partially, or completely, occluded by an object;
- markers are partially, or completely, outside of the FoV;
- or not all lines can be detected (e.g., due to low contrast).

3.2. Feature point-based cue (FPC)

The real world corners of the fiducial marker can be considered as feature points. These feature points are 2D back-projections of the 3D corners onto the camera's image plane. Consequently, they constrain the 3D orientation and especially the 3D position of the camera. The feature point-based tracker produces an estimate of the location of the feature points. This estimate is one of the cues given to the fusion framework (identified as FPC) and is based on the feature points tracking technique presented by Davison.[1]

In order to validate the possible candidates of feature point matches, a template of each 3D point P^i is defined as a grey-level patch around the 2D back-projection of the marker corner. This template is built once, at initialisation.

The feature points are searched in the video frame. Using the estimate of the camera pose it is possible to locate a rough region where these feature points should lie. We define r^i as the region where the 2D back-projection of P^i in the image plane may lie, given the filter's distribution of the estimated transformation T. Assume, for the moment, that this region is known. At runtime, a window search is performed: r^i is cross-correlated with the template of the corner. This gives a set of correlation maps $corr\_map^i$, one for each corner. A set S^i of coordinates is defined by thresholding each correlation map.

$$S^i = \left\{[x, y] \middle| corr\_map^i(x, y) > th\right\}, \tag{2}$$

where i indexes the corners of the marker. These sets are fed into the filter for update (See Section 3.3). Each corner whose feature point is positively detected makes the filter converge to a more stable estimate. Three points are necessary to completely constrain the 6D pose. However, the filter can be updated even with only one feature point. A reliable measurement (FPC) might be unavailable in the following situations:

- P^i is occluded by an object,
- P^i is outside of the FoV,
- the region does not contain the real back-projection (due to a bad region estimation), or
- P^i is inside the region but no point in the correlation map is beyond the threshold (for example, because the viewpoint is drastically changed).

3.3. Data fusion

The goal of the fusion is to obtain a synergy by combining both cues. The individual weaknesses described in previous sections are solved by this combination. Special attention is given to the occlusion and the illumination (contrast) problems of marker detection (weaknesses of the MC) and the recognition of feature points under viewpoint changes (weakness of the FPC). On the one hand, the occlusion and illumination problems are solved by switching to the feature point-based cue which is independent of the recognition of the marker. On the other hand, when the viewpoint drastically changes, the correlation of the templates does no longer provide the localisation of the feature points and, consequently, the filter drifts. As soon as the marker is recognised again, the filter switches to the marker-based cue and it recovers from the incorrect estimate. Details of this combination are described next.

In filter-based camera tracking approaches, a continuous estimation of the transformation T is provided. The filter provides a prediction of T even when no measurement is available. This usually happens when motion is fast (and the image becomes blurred) or when the references are not visible. We target applications where the camera is hand-held or attached to the user's head. Under these circumstances, Kalman filter-based approaches although extensively used for ego tracking, achieve worse performance because the motion is not white nor has Gaussian statistics.[19,20] To avoid the Gaussianity assumption, we have chosen a camera tracking algorithm that uses a particle filter. For more details on particle filters, the reader is referred to.[21]

Each particle n in the filter represents a possible transformation T_n. At initialisation, the value of all particles is set to the transformation provided by the MC. The 3D coordinates of the corners are given by the geometric description of the marker. At each frame, the propagation model used is a random walk (uniform distribution). The update of the weight of each particle n is calculated using its measurement noise (likelihood)

$$w_n = p(Y|T_n), \tag{3}$$

where w_n is the weight of particle n, Y is the measurement, and $p(.|.)$ is the conditional probability. The key role of the fusion filter is to switch between two likelihoods or observation models depending on the measurement type: MC or FPC.

As long as the marker is detected, the system uses the MC measurement ($Y = T_{MC}$) to update the particle filter. In the update step, the weight of each particle n is calculated from the normalised squared Euclidean distance

$$\Sigma(a_j - b_j)^2/\sigma_j^2 \tag{4}$$

between T_{MC} and T_n, where j indexes the elements of the vector T. The normalisation factors σ_j are fixed off-line upon experimentation. The templates associated to the corners of the marker are also updated as long as the marker is detected. On the other hand, when the MC fails to detect the marker, the system relies on the FPC ($Y = \bigcup_i S^i$). First, the regions where the corresponding feature points may lie are computed. For each particle, the 2D back-projection $[p_{n,x}, p_{n,y}]$ of a 3D point $[P_X, P_Y, P_Z]$ can be computed using the transformation T_n and the calibration of the camera:

$$p_n = \left[\begin{array}{c} p_{n,x} \\ p_{n,y} \end{array} \right] = K \cdot [R_n|t_n] \cdot P = K \cdot [R_n|t_n] \left[\begin{array}{c} P_X \\ P_Y \\ P_Z \end{array} \right], \qquad (5)$$

where K is the calibration matrix (computed off-line), R is the rotation matrix formed using rot_X, rot_Y and rot_Z, and $t = [t_X, t_Y, t_Z]^T$ is the translation vector. For each 3D point P^i, a bounding box is computed containing all the back-projections of P^i given all particle's transformations T_n. This bounding box is the region r^i. Then, the correlation maps are obtained (See Section 3.2). Once the sets S^i are defined, the weights can be calculated. For each particle n, the back-projection of P^i gives an image point (p^i_n). The weight of the particle n is proportional to the Euclidean distances from p^i_n to any element in S^i.

Once the update is concluded either using the MC or the FPC, T is obtained from a weighted sum of T_n.

4. EXPERIMENTS

In order to assess the performance of our system, we compare the 6D pose of the camera estimated by the system proposed by Pupilli and Calway,[7] which uses a particle filter updated through feature points (identified here as *feature point-based camera tracker* FPCT), the MC and the estimate of our data fusion approach. A custom video sequence is used for this purpose. In this sequence, the camera moves around a marker, and several occlusions occur, either manually produced (by covering the pattern) or happening when the marker is partially outside of the FoV. The FPCT is initialised once, at the beginning, using the estimate of the MC. Figure 1 shows the estimation of the FPCT and our approach (one realisation). Note that the rotation is expressed in Euler angles converted from the rotation quaternion.

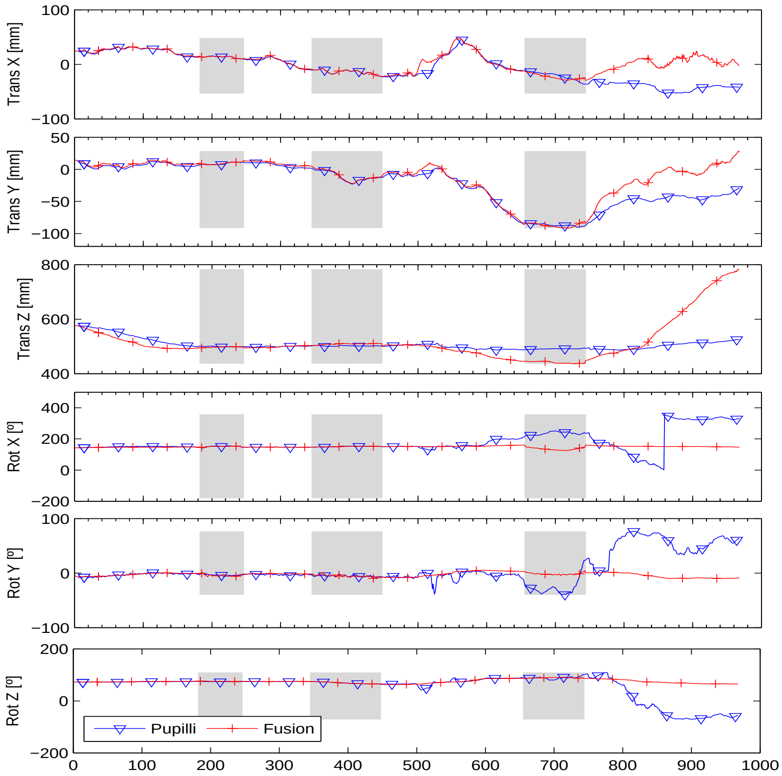
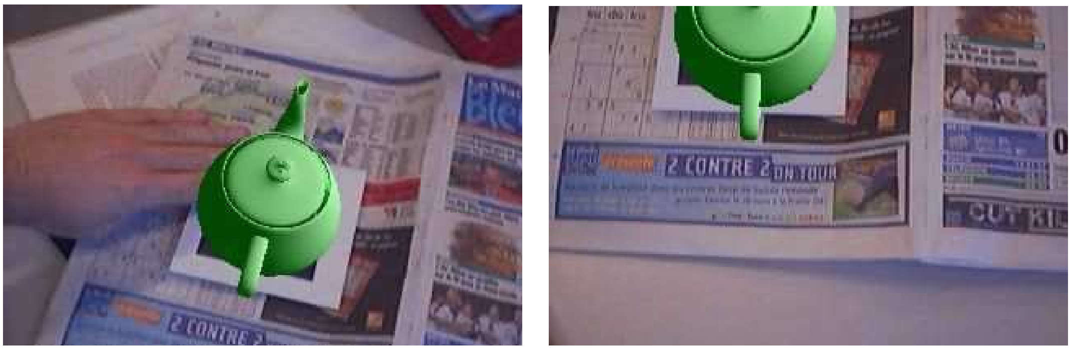


Figure 1. First experiment. Translation and rotation in X,Y and Z axis. Shaded regions represent occlusions (Manually produced: frames 182-246, 345-448. Marker partially outside of the FoV: frames 655-744).

The measurement of the MC is not shown for better clarity. This measurement is very close to the filtered output of our approach, except during the occlusions, where no measurement is produced at all. The FPCT (inverted triangles) is capable of tracking despite the first two occlusions. However, it looses track after the third occlusion. This could be due to an incomplete update step (only two 3D points are available while the marker is partially outside of the FoV, frames 655-744), leading to a bad prediction.

In our approach (crosses) a continuous estimate is provided even during occlusions. This shows that the fusion system keeps track of the marker along the whole sequence thanks to the FPC. As this experiment illustrates, the fusion solves the main drawback of the MC: occlusions. Snapshots from several frames of the sequence are shown in Figure 2. A virtual teapot is placed on the marker to show correct alignment in front of occlusions.



(a) manual occlusion

(b) the marker is escaping the field of view

Figure 2. First experiment. Snapshots of the sequence inserting a virtual teapot on the marker. Fusion tracker using FPC measurements.

As stated before (see Section 3.1) the MC uses a fixed threshold for binarisation and further marker identification. When the illumination changes considerably, the contrast becomes too low in the contour of the marker and the detection algorithm fails. On the other hand, the FPC is illumination-invariant because the templates are normalised with respect to their luminance means. Another test has been conducted to analyse the tracking performance in case of changes in illumination. A second custom video sequence has been used. It features larger rotations and a varying offset of ±100 on the RGB channels.

Figure 3 shows the evolution of the estimation for the MC and our approach (one realisation). The ground truth is obtained, separately, with the MC tracking the marker in the original sequence (without illumination changes). The analysis of the evolution shows that the MC provides an estimate only when the offset is between -50 and +50 (approximately). It can be seen that our fusion deviates from the ground truth in FPC mode (see, for instance, Rot X between frame 240 and 280). The necessary recovery from this drift is provided by the MC once the marker is detected again. This experiment illustrates two benefits of the fusion. On the one hand, the fusion solves the main drawback of the FPC: viewpoint changes. On the other, the fusion solves the illumination problem of the MC using the invariance to light conditions of the FPC. Snapshots from several frames of the sequence are shown in Figure 4. As for Fig. 2, a virtual teapot is added to show the correct alignment. This experiment shows that our fusion can face illumination variations successfully.

All the experiments were carried out on a 1700 MHz processor. Mean frame rates achieved for a 320x240 pixels video stream (acquisition at 30 Hz) are shown in Table 1. Although the fusion tracker slows down the individual frame rates of each tracker, the difference is negligible.

5. CONCLUSIONS

We have presented a tracking system that fuses a marker-based cue (MC) and a feature point-based cue (FPC). In this framework, measurements from both cues are fed into a particle filter in order to track the camera position and orientation. Their complementary performance is exploited. Indeed, the measurements depend on the availability of the reference or on the capacity of the tracker to recover from a biased estimate.

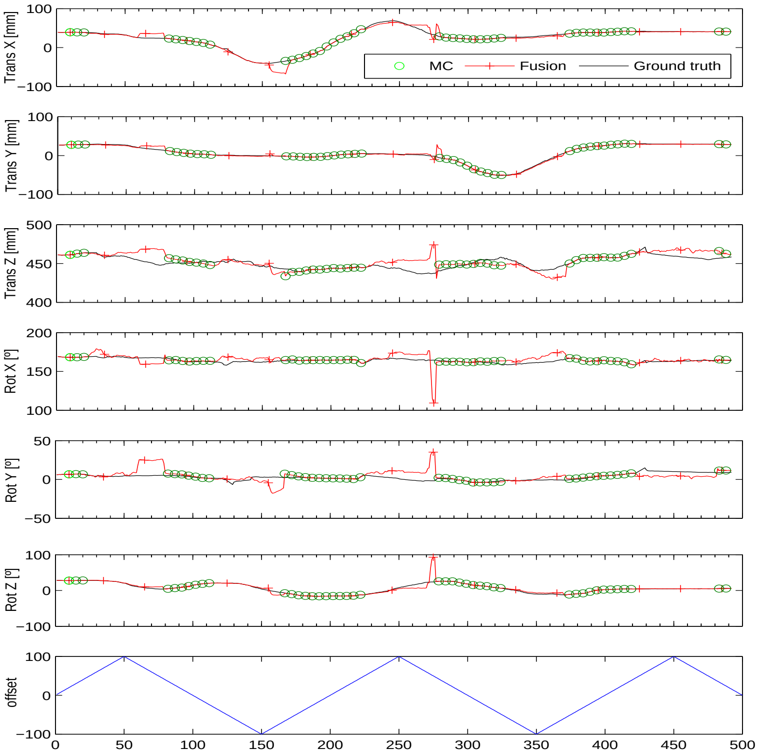
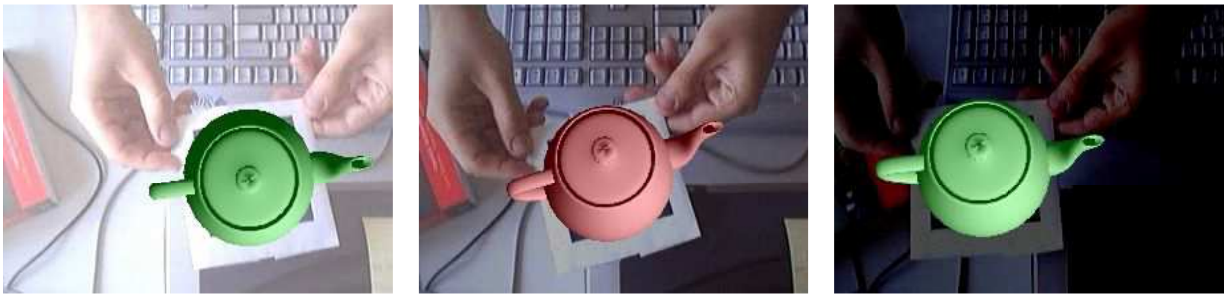


Figure 3. Second experiment. Translation and rotation in X,Y and Z axis. The MC (circles) detects the marker only when the offset is between -50 and +50 (approximately).



(a) saturation (offset +100)

(b) around MC detection threshold (offset 0)

(c) darkness (offset -100)

Figure 4. Second experiment. Snapshots of the sequence inserting a virtual teapot on the marker. Fusion tracker using MC (b) and FPC (a and c).

	frame rate [Hz]
MC	26.9
FPC (1000 particles)	19.1
Fusion (MC mode)	24.6
Fusion (FPC mode)	18.2

Table 1. Mean frame rates achieved for the MC and FPC individually, and for our system relying on one or another tracker.

Experiments on real sequences show that the combination of MC and FPC solves the weaknesses of both approaches. MC is sensible to occlusions and illumination variations, whereas FPC lacks robustness in front of drastic viewpoint changes. A clear improvement in the tracking performance is obtained, providing a synergy of their individual advantages.

Future paths of research will focus on a more advanced merging of information and a superior feature point-based cue that can handle more severe viewpoint changes.

ACKNOWLEDGMENTS

The first author is partially supported by the Swiss National Science Foundation, under grant number 200021-113827, and by the European Commission under contract FP6-027026-K-SPACE and contract FP6-038398-VISNET2.

REFERENCES

1. A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Intl. Conf. on Computer Vision (ICCV)*, 2003.
2. J. Oliensis, "A critique of structure-from-motion algorithms," *Computer Vision and Image Understanding: CVIU* **80**(2), pp. 172–214, 2000.
3. Y. Abdeljaoued, D. Marimon, and T. Ebrahimi, *3D Videocommunication*, ch. Tracking and User Interface for Mixed Reality. WILEY & SONS, 2005.
4. T. Okuma, T. Kurata, and K. Sakaue, "Fiducial-less 3-d object tracking in AR systems based on the integration of top-down and bottom-up approaches and automatic database addition," in *IEEE and ACM Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, p. 260, 2003.
5. X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in AR systems: A comparative study," in *Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 97–106, 2002.
6. D. Koller, G. Klinker, E. Rose, D. Breen, R. Wihtaker, and M. Tuceryan, "Real-time vision-based camera tracking for augmented reality applications," in *ACM Virtual Reality Software and Technology*, 1997.
7. M. Pupilli and A. Calway, "Real-time camera tracking using a particle filter," in *British Machine Vision Conference*, pp. 519–528, BMVA Press, 2005.
8. ARToolkit. www.hitl.washington.edu/artoolkit/.
9. B. Allen, G. Bishop, and G. Welch, "Tracking: Beyond 15 minutes of thought," in *Course Notes, Ann. Conf. Computer Graphics and Interactive Techniques (Siggraph)*, 2001.
10. R. Azuma and G. Bishop, "Improving static and dynamic registration in an optical see-through HMD," in *Proc. of SIGGRAPH*, pp. 197–204, ACM Press, 1995.
11. S. You, U. Neumann, and R. Azuma, "Hybrid inertial and vision tracking for augmented reality registration," in *Proc. of IEEE Virtual Reality (VR)*, pp. 260–267, Mar 1999.
12. M. Kanbara, H. Fujii, H. Takemura, and N. Yokoya, "A stereo vision-based augmented reality system with an inertial sensor," in *IEEE and ACM Int. Symp. on Augmented Reality (ISAR)*, pp. 97–100, 2000.
13. S. You and U. Neumann, "Fusion of vision and gyro tracking for robust augmented reality registration," in *Proc. of IEEE Virtual Reality (VR)*, pp. 71–78, 2001.
14. L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Proc. of the Int. Symp. on Mixed and Augmented Reality*, pp. 27–36, Sep–Oct 2002.
15. K. Satoh, S. Uchiyama, H. Yamamoto, and H. Tamura, "Robot vision-based registration utilizing bird's-eye view with user's view," in *Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 46–55, 2003.
16. H. Najafi, N. Navab, and G. Klinker, "Automated initialization for marker-less tracking: a sensor fusion approach," in *IEEE and ACM Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 79–88, 2004.
17. D. Wagner and D. Schmalstieg, "First steps towards handheld augmented reality," in *Proc. Seventh IEEE International Symposium on Wearable Computers*, pp. 127–135, Oct 2003.
18. H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Intl. Workshop on Augmented Reality (IWAR)*, pp. 85–94, 1999.

19. L. Chai, K. Nguyen, B. Hoff, and T. Vincent, “An adaptive estimator for registration in augmented reality,” in *Intl. Workshop on Augmented Reality (IWAR)*, pp. 23–32, 1999.

20. F. Ababsa, M. Mallem, and D. Roussel, “Comparison between particle filter approach and kalman filter-based technique for head tracking in augmented reality systems,” in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA '04)*, **1**, pp. 1021–1026, April-May 2004.

21. M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. on Signal Processing* **50**, pp. 174–188, Feb. 2002.