COMBINATION OF VIDEO-BASED CAMERA TRACKERS USING A DYNAMICALLY ADAPTED PARTICLE FILTER

David Marimon and Touradj Ebrahimi
Signal Processing Institute
Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland

Keywords: Sensor fusion, video camera tracking, particle filter, adaptive estimation.

Abstract: This paper presents a video-based camera tracker that combines marker-based and feature point-based cues in a particle filter framework. The framework relies on their complementary performance. 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, feature point tracking can still provide estimates given a limited number of feature points. However, these tend to drift and usually fail to recover when the reference reappears. Therefore, we propose a combination 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 marker is available whereas the feature point-based cue is selected otherwise. The feature points tracked are the corners of the marker. Evaluations on real cases show that the fusion of these two approaches outperforms the individual tracking results.
Filtering techniques often suffer from the difficulty of modeling the motion with precision. A second related topic presented is an adaptation method for the particle filer. It achieves tolerance to fast motion manoeuvres.

1 INTRODUCTION

Combination of tracking techniques has proven to be necessary for some camera tracking applications. To reach a synergy, techniques with complementary performance have first to be identified. Research on camera tracking has concentrated on combining sensors within different modalities (e.g. inertial, acoustic, optic). However, this identification is possible within a single modality: video trackers. Video-based camera tracking can be classified into two categories that have compensated weaknesses and strengths: bottom-up and top-down approaches (Okuma et al., 2003). For the first category, the six Degrees of Freedom (DoF), 3D position and 3D orientation, estimates are 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* (Zhang et al., 2002) can be classified in the first group. Although they have a high detection rate and estimation speed, they still lack tracking robustness: the marker(s) must be always visible thus limiting the user actions. In contrast to bottom-up approaches, top-down techniques such as *filter-based camera tracking* allow track continuation when the reference is temporarily unavailable (e.g. due to occlusions). They use predictive motion models and update them when the reference is again visible (Davison, 2003; Koller et al., 1997; Pupilli and Calway, 2005). Their weakness is, in general, the drift during the absence of a stable reference (usually due to feature points difficult to recognise after perspective distortions).

Filtering techniques often suffer from the difficulty of modelling the motion with precision. More concretely, the errors in the system of equations are usually modelled with noise of fixed variance, also called *hyper-parameter*. In practice, these variances are rarely constant in time. Hence, better accuracy is achieved with filters that adapt, or self-tune, the hyper-parameters online (Maybeck, 1982).

In this paper, we present a particle-filter based

Marimon D. and Ebrahimi T. (2007).
COMBINATION OF VIDEO-BASED CAMERA TRACKERS USING A DYNAMICALLY ADAPTED PARTICLE FILTER.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 363-370

camera tracker that dynamically adapts its hyper-parameters. The main purpose of this framework is to take advantage of the complementary performance of two particular video-trackers. The system combines the measurements of a marker-based cue (MC) and a feature point-based cue (FPC). The MC tracks a square marker using its contour lines. The FPC tracks the corners of the marker. Besides this novel combination, an adaptive estimator is also proposed. The process noise model is adapted online to prevent deviation of the estimate in case of fast manoeuvres as well as to keep precise estimates during slow motion stages.

The paper is structured as follows. Section 2 describes similar works. The techniques involved in the combination 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 discussed.

2 RELATED WORK

In hybrid tracking, systems that combine diverse tracking techniques have shown that the fusion obtained enhances the overall performance (Allen et al., 2001).

The commonly developed fusions are inertial-acoustic and inertial-video (Allen et al., 2001). Inertial sensors usually achieve better performance for fast motion. On the other hand, in order to compensate for drift, an accurate tracker is needed for periodical correction. Video-based tracking performs better at low motion and fails with rapid movements. During the last decade, interest has grown on combining inertial and video. Representative works are those presented by Azuma and by You. Azuma used inertial sensors to predict future head location combined with an optoelectrical tracker (Azuma and Bishop, 1995). As presented in (You et al., 1999), the fusion of gyros and video tracking shows that computational cost of the latter can be decreased by 3DoF orientation self-tracking of the former. It reduces the search area and avoids tracking interruptions due to occlusion. Several works have combined marker-based approaches with inertial sensors (Kanbara et al., 2000; You and Neumann, 2001). (You and Neumann, 2001) presented a square marker-based tracker that fuses its data with an inertial tracker, in a Kalman filtering framework. Among the existing marker-based trackers, two recent works, (Claus and Fitzgibbon, 2004) and (Fiala, 2005) stand out for their robustness to illumination changes and partial occlusions. (Claus and Fitzgibbon, 2004) takes advantage of machine learning techniques, and trains a classifier with a set of markers under different conditions of light and viewpoint. No particular attention is given to occlusion handling. (Fiala, 2005) uses spatial derivatives of grey-scale image to detect edges, produce line segments and further link them into squares. This linking method permits the localisation of markers even when the illumination is different from one edge to the other. The drawback of this method is that markers can only be occluded up to a certain degree. More precisely, the edges must be visible enough to produce straight lines that cross at the corners.

However, little attention has been given to fusing diverse techniques from the same modality. Several researchers have identified the potential of video-based tracking fusion (Najafi et al., 2004; Okuma et al., 2003; Satoh et al., 2003). However, (Okuma et al., 2003) is the only reported work to fuse data from a single camera. Their system switches between a model-based tracker and a feature point-based tracker, similar to that of (Pupilli and Calway, 2005). Nonetheless, this framework takes limited advantage of the filtering framework and still needs the assistance of an inertial sensor. Combination of low-level features has been addressed in (Vacchetti et al., 2004). Edge and feature points are used for model-based tracking of textured and non-textured objects.

Online hyper-parameter adaptation has been applied to video tracking with growing interest. (Chai et al., 1999) presented a multiple model adaptive estimator for augmented reality registration. Each model characterises a possible motion type, namely fast and slow motion. Switching is done according to camera's prediction error. Although multiple models may seem an attractive solution, several works have identified their drawbacks. When the state space is large, the number of necessary models becomes untractable and their quantisation must be fine enough in order to obtain good accuracy (Ichimura, 2002). (Ichimura, 2002) and, more recently, (Xu and Li, 2006) have shown the advantages of tuning the hyper-parameters with a single motion model. Both have employed online tuning for 2D tracking purposes. (Ichimura, 2002) presented an adaptive estimator that considers the hyper-parameters as part of the state vector. Whilst providing good results, this technique adds complexity to the filter and moves the problem to the hyper-hyper-parameters that govern change in hyper-parameters. (Xu and Li, 2006) presents a simpler adaptation algorithm that calculates a similarity of predictions between frames and updates the hyper-parameters accordingly. As described in Section 3.5, the adaptation method presented here is closer to this technique.

3 SYSTEM DESCRIPTION

This section describes the particle filter, how the marker-based and the feature point-based cues are obtained, as well as the procedure used to fuse them in the filter. The dynamic tuning of the filter is also exposed.

3.1 Particle Filter

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 motion tracking, lead to a non optimal solution because the motion is not white nor has Gaussian statistics (Chai et al., 1999). To avoid the Gaussianity assumption, we have chosen a camera tracking algorithm that uses a particle filter. More precisely, we have chosen a sample importance resampling (SIR) filter. For more details on particle filters, the reader is referred to (Arulampalam et al., 2002).

Each particle n in the filter represents a possible transformation

$$T_n = [t_X, t_Y, t_Z, rot_W, rot_X, rot_Y, rot_Z]_n, \quad (1)$$

where t are the translations and rot is the quaternion for the rotation. T determines the 3D relation of the camera with respect to the world coordinate system. We have avoided adding the velocity terms so as not to overload the particle filter (which would otherwise affect the speed of the system).

For each video frame, the filter follows two steps: prediction and update. The probabilistic motion model for the prediction step is defined as follows. The process noise (also known as transition prior $p(T_n(k)|T_n(k-1))$) is modelled with a Uniform distribution centred at the previous state $T_n(k-1)$ (frame $k-1$), with variance q (process noise's -also called system noise- vector of hyper-parameters). The reason for this type of random walk motion model is to avoid any assumption on the direction of the motion. This distribution enables faster reactivity to abrupt changes. The propagation for the translation vector is

$$T_n(k)\big|_{t_X,t_Y,t_Z} = T_n(k-1)\big|_{t_X,t_Y,t_Z} + u_t \quad (2)$$

where u_t is a random variable coming from the uniform distribution, particularised for each translation axis. The propagation for the rotation is

$$T_n(k)\big|_{rot} = u_{rot} \times T_n(k-1)\big|_{rot} \quad (3)$$

where $\times$ is a quaternion multiplication and u_{rot} is a quaternion coming from the uniform distribution of the rotation components. In the update step, the weight of each particle n is calculated using its measurement noise (likelihood)

$$w_n = p(Y|T_n), \quad (4)$$

where w_n is the weight of particle n and Y is the measurement. The key role of the combination filter is to switch between two sorts of likelihood depending on the type of measurement that is used: MC or FPC. Once the weights are obtained, these are normalised and the update step of the filter is concluded. The corrected mean state $\widehat{T}$ is given by the weighted sum of T_n. $\widehat{T}$ is used as output of the camera tracking system.

Figure 1: Square marker used for the MC.

3.2 Marker-based Cue (MC)

We use the marker-based system provided by (Kato and Billinghurst, 1999) to calculate the transformation T between the world coordinate frame and that of the camera (3D position and 3D orientation). As explained in Section 3.4, this transformation is the measurement fed into the filter for update.

At each frame, the algorithm searches for a square marker (see Figure 1) 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 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 only 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 (Kato and Billinghurst, 1999). When this information is not available, no pose can be calculated. This occurs in the following cases: markers are partially or completely occluded by an object; markers are partially or completely out of the FoV; or not all lines can be detected (e.g., due to low contrast).

3.3 Feature Point-based Cue (FPC)

In order to constrain the camera pose estimation, the back-projection of salient or feature points in the scene can be used. For this purpose, both the 3D location of the feature point P and the 2D back-projection p is needed. In homogeneous coordinates,

$$p = K \cdot [R|t] \cdot P, \quad (5)$$

where K is the calibration matrix (computed off-line), R is the rotation matrix formed using the quaternion rot and $t = [t_X, t_Y, t_Z]^T$ is the translation vector.

Among the available feature points in the scene, we have selected the corners of the marker because their 3D location in the world coordinate frame is known. The intensity level information is chosen as a description of these feature points, for further recognition. A template of each corner is defined with the 16x16 pixel patch around the corner's back-projection in the image plane. This template is stored at initialisation.

At runtime, these feature points are searched in the video frame. A region is defined around the estimated location of each feature point. Assume, for the moment, that these regions are known. Each region is cross-correlated with the template of the corresponding feature point, thus resulting in a correlation map.

As explained in the next section, the set of correlation maps is the measurement fed into the filter for update. Each template that is positively correlated makes the filter converge to a more stable estimate. Three points are necessary to robustly determine the six DoF. However, the filter can be updated even with only one feature point. A reliable feature point might be unavailable in the following situations: a corner is occluded by an object; a corner is outside of the FoV; the region does not contain the feature point (due to a bad region estimation); or the feature point is inside the region but no correlation is beyond the threshold (e.g., because the viewpoint is drastically changed).

3.4 Cues Combination

The goal of the system is to obtain a synergy by combining both cues. Individual weaknesses previously described are thus lessened by this combination. Special attention is given to the occlusion and illumination problems in the MC and the viewpoint change in the FPC.

At initialisation, the value of all particles of the filter is set to the transformation estimated by the marker-based cue T_{MC}.

As long as the the marker is detected, the system uses the MC measurement to update the particle filter ($Y = T_{MC}$). The likelihood is modelled with a Cauchy distribution centered at the measurement T_{MC}

$$p(T_{MC}|T_n) = \prod_i \frac{r_i}{\pi \cdot ((T_{n,i} - T_{MC,i})^2 + r_i^2)}, \quad (6)$$

where r is the measurement noise and i indexes the elements of the vectors. This particular distribution's choice has its origin in the following reasoning. In the resampling step of the filter, particles with insignificant weights are discarded. A problem may arise when the particles lie on the tail of the measurement noise distribution. The transition prior $p(T_n(k)|T_n(k-1))$ determines the region in the state-space where the particles fall before their weighting. Hence, it is relevant to evaluate the overlap between the likelihood distribution and the transition prior distribution. When the overlap is small, the number of particles effectively resampled is too small. Figure 2 shows an instance of overlapping region. It must be pointed out that due to computing limits, some values fall to zero even though their real mathematical value is greater than that (the support of a Gaussian distribution is the entire real line). In the example of this

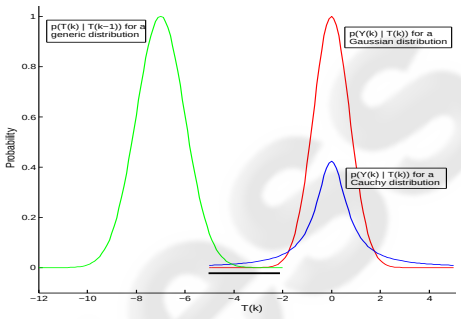

Figure 2: Overlap between transition prior distribution and the likelihood distribution: modelled with a Gaussian (no overlap) and with a Cauchy distribution (thick line).

figure, there is no sufficient computed overlap for the Gaussian distribution (commonly used), whereas the tail of the Cauchy distribution covers the necessary state-space. Therefore, we have chosen a long-tailed density that better covers the state-space, while still being a realistic measurement noise (Ichimura, 2002). Once the weighting is computed, the templates associated to each corner of the marker are also updated. This is done every time the marker is detected. For this purpose the patch around the back-projection of each corner is exchanged with the previous template description. As this template is used by the FPC, it is important that the latest possible description is available. Contrary to what might be thought, template updating does not introduce drift in this case. The patch around the corner is a valid descriptor every time the marker is detected.

On the other hand, when the MC fails to detect the marker, the system relies on the FPC ($Y = correlation\_maps$) and another likelihood is used. As a previous step to obtaining the correlation maps provided by the FPC (see Section 3.3), it is necessary to calculate the *regions* around the estimated location of each feature point. For each corner, all the back-projections given the transformations T_n are computed (see Eq. 5). The *region* is the bounding box contain-

ing all these back-projections. These bounding boxes are fed into the FPC and the correlation maps are obtained in return. The weights can then be calculated. First, a set of 2D coordinates is obtained by thresholding each correlation map.

$$S_j = \left\{ [c_x, c_y] \middle| correlation\_map_j(c_x, c_y) > th_{corr} \right\}, \quad (7)$$

where j indexes the corners of the marker. Second, for each particle, a subset is kept with the points in S_j that are within a certain Euclidean distance from the corresponding back-projection $[p_{n,x}, p_{n,y}]$

$$\widehat{S}_{n,j} = \left\{ [c_x, c_y] \in S_j \middle| dist(c, p_n) < th_{dist} \right\}. \quad (8)$$

The weight of the particle n is proportional to the number of elements $(|.|)$ in the subsets $\widehat{S}_{n,j}$

$$w_n = exp(-\sum_j |S_{n,j}|). \quad (9)$$

As it can be seen, the likelihood for the FPC measurement is much less straightforward to compute than the MC. Nevertheless, the weights can be calculated independently of the number of feature points recognised whereas the likelihood for the MC is available only if the marker is visible.

Algorithm 1 expresses the process followed by the combination. It is assumed that the filter has been initialised at the first detection of the marker. The description of the marker is stored in the *pattern* variable.

Algorithm 1 Combination procedure.

```
loop
  vframe ← getVideoFrame()
  marker ← detectMarker( vframe )
  if pattern.correspondsTo( marker ) then
    T_MC ← MC.calcTransformation( marker )
    templates ← extractPatches( marker , vframe )
    T̂ ← filter.updateFromMC( T_MC )
  else
    reg ← filter.calcRegions()
    corr_maps ← FPC.calcMaps( reg , templates )
    T̂ ← filter.updateFromFPC( corr_maps )
  end if
end loop
```

This filtering framework has several advantages. Combination through a filter provides a continuous estimate which is free of jumps that disturb the user's interaction. Frameworks often fall into static solutions giving little opportunity for shaping. The likelihood switching method proposed is generic enough to be used with very different types of cues or sensors such as inertial, etc.

3.5 Dynamic Tuning of the Filter

The goal of the dynamic tuning is to achieve better tracking accuracy together with robustness in front of manoeuvres.

The long-tailed Cauchy-type distribution of the measurement noise is not sufficient to cover the state-space in front of rapid manoeuvres. Figure 3 shows the effect of a large manoeuvre on the probabilistic model assumed. Again, the computing limits play a role on the positive overlap of distributions.

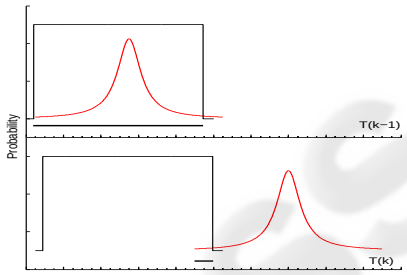

Figure 3: Slow motion (top) and fast motion (bottom). Overlap indicated with a thick line. When a fast manoeuvre occurs, the overlap between transition prior (uniform distribution) and the likelihood (cauchy distribution) is small.

Either the likelihood or the transition prior distributions should broaden in order to face this problem. The measurement noise model is related to the sensor. Hence, the model should only be tuned if a quality value of the measurement provided by the sensor is available. This quality value is usually unrelated to motion and thus of no use to face the manoeuvre problem. On the other hand, the process noise variance q is related to the motion model. We propose to tune the process noise adaptively. As said before, there are six DoF, three for orientation and three for position. Practice demonstrates that motion changes do not necessarily affect all axes in the same manner. Contrary to the adaptive estimators cited before, we propose a tuning that considers each degree of freedom independently. The process variance for each axis q_i is tuned according to the weighted distance from the current corrected mean state $\widehat{T}_i(k)$ to that in the previous frame $\widehat{T}_i(k-1)$

$$\varphi_i = \frac{[\widehat{T}_i(k) - \widehat{T}_i(k-1)]^2}{q_i(k)^2} + \Delta_{min}$$
$$q_i(k+1) = max\left(q_i(k) \cdot min\left(\varphi_i; \Delta_{max}\right); \widetilde{q}_{i,min}\right), \quad (10)$$

where Δ_{min} and Δ_{max} are the minimal and maximal variations, respectively, and $\widetilde{q}_{i,min}$ is the lower bound for the hyper-parameter of axis i. This method permits a large dynamic range for the variance of each axis as it uses the previous q_i to calculate the current value. In

addition, it does not add complexity to the filter state as in (Ichimura, 2002) nor to the system by means of multiple model estimation as the system proposed in (Chai et al., 1999).

4 EXPERIMENTS

In order to assess the performance of our fusion, we compare the 6D pose of the camera estimated by the system proposed by (Pupilli and Calway, 2005), which we will call *feature point-based camera tracker* FPCT, ARToolkit, which we will call *marker-based camera tracker* MCT, and the corrected mean estimate $\widehat{T}$ of our data fusion approach. A custom video sequence is used as input 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 MCT. Figure 4 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. The estimate of the MCT (green dots)

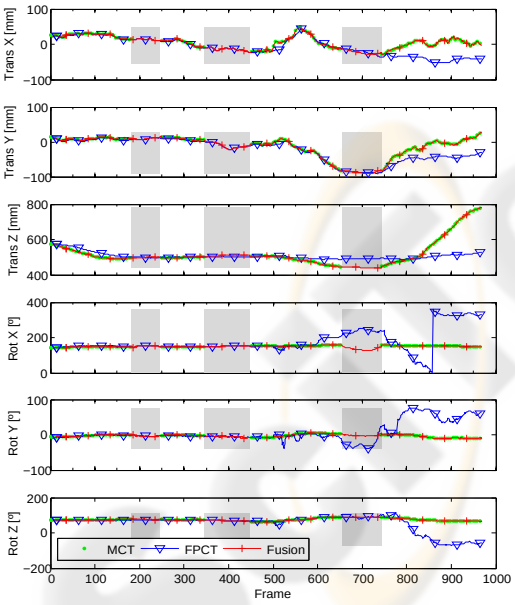

Figure 4: First experiment. Translation and rotation in X,Y and Z axes. Shaded regions represent occlusions (Manually produced: frames 182-246, 345-448. Marker partially out of FoV: frames 655-744).

is difficult to observe because it is very close to the filtered estimate of our approach, except during the occlusions, where no output 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 the FoV, frames 655-744), leading to a bad prediction. As this experiment illustrates, using the MC to update the filter and the templates of the FPC produces better results. In addition, using the FPC enables the fusion to provide an estimate throughout the whole sequence. The fusion addresses the main drawbacks of each component: partial occlusions for the MC, and the viewpoint change for the FPC. Snapshots from several frames of the augmented scene during occlusions are shown in Figure 5.

(a) Occlusion (b) Escaping the FoV

Figure 5: First experiment. A virtual teapot placed on the marker to show correct alignment.

In a second experiment, the adaptive filtering proposed is compared to similar work. Xu (Xu and Li, 2006) presented an adaptation method for 2DoF that can be extended in terms of Eq. (10) to 6DoF as follows

$$\varphi = exp\left(-0.5\sum_i \frac{[\widehat{T}_i(k)-\widehat{T}_i(k-1)]^2}{\widetilde{q}_i^2}\right)$$

$$q_i(k+1) = max\left(min\left(\widetilde{q}_i\cdot\sqrt{1/\varphi};\widetilde{q}_{i,max}\right);\widetilde{q}_{i,min}\right), \quad (11)$$

where $\widetilde{q}_{max}$ and $\widetilde{q}_{min}$ are the upper and lower bound vectors and $\widetilde{q}_i$ is the nominal value for axis i. Note that $\widetilde{q}_i$, $\widetilde{q}_{i,max}$ and $\widetilde{q}_{i,min}$ are fixed offline to a single value for the three rotation axes and another single value for the three translation axes. Remark that the prediction error in one axis affects all other hyper-parameters, as φ is unique for all the axes. Consequently, the model of process noise in one axis may grow even when the real error in that axis is small. Moreover, the dynamic range of the variance of each axis is limited offline whereas our proposed dynamic tuning has only a lower bound to insure stability when motion is almost static. In order to compare the proposed method to this approximation of Xu's approach, a second custom video sequence with several abrupt manoeuvres is used. The upper bound is fixed to the maximal value achieved by our technique in this particular sequence. The lower bound is the same for both. The variances in our method are initialised with the nominal values $\widetilde{q}_i$. The variation bounds Δ_{min} and Δ_{max} are

Table 2: Mean frame rates achieved for the MCT and FPCT individually, and for our fusion filter updated from one or the other cue.

Tracker	frame rate [Hz]
MCT	26.9
FPCT (1000 particles)	19.1
Fusion (update with MC)	23.8
Fusion (update with FPC)	17.9

5 CONCLUSION

We have presented a combination of video trackers within a particle filter framework. The filter uses two cues provided by a marker-based approach and a feature point-based one. The motion model is adapted online according to the distance between past estimates.

Experiments show that the proposed combination produces a synergy. The system tolerates occlusions and changes of illumination. Independent adaptive tuning for the model of each DoF demonstrates superior performance in front of manoeuvres.

In our future research, we will focus on extending the FPC to feature points beyond the four corners of the marker and enhancing their viewpoint sensibility.

ACKNOWLEDGEMENTS

The first author is supported by the Swiss National Science Foundation (grant number 200021-113827), and by the European Networks of Excellence K-SPACE and VISNET2.

REFERENCES

Allen, B., Bishop, G., and Welch, G. (2001). Tracking: Beyond 15 minutes of thought. In *SIGGRAPH*.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188.

Azuma, R. and Bishop, G. (1995). Improving static and dynamic registration in an optical see-through HMD. In *SIGGRAPH*, pages 197–204. ACM Press.

Chai, L., Nguyen, K., Hoff, B., and Vincent, T. (1999). An adaptive estimator for registration in augmented reality. In *IWAR*, pages 23–32.

Claus, D. and Fitzgibbon, A. (2004). Reliable fiducial detection in natural scenes. In *European Conference on Computer Vision (ECCV)*, volume 3024, pages 469–480. Springer-Verlag.

Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *ICCV*.

Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *CVPR*, volume 2, pages 590–596.

Ichimura, N. (2002). Stochastic filtering for motion trajectory in image sequences using a monte carlo filter with estimation of hyper-parameters. In *ICPR*, volume 4, pages 68–73.

Kanbara, M., Fujii, H., Takemura, H., and Yokoya, N. (2000). A stereo vision-based augmented reality system with an inertial sensor. In *ISAR*, pages 97–100.

Kato, H. and Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IWAR*, pages 85–94.

Koller, D., Klinker, G., Rose, E., Breen, D., Wihtaker, R., and Tuceryan, M. (1997). Real-time vision-based camera tracking for augmented reality applications. In *ACM Virtual Reality Software and Technology*.

Maybeck, P. S. (1982). *Stochastic Models, Estimation, and Control*, volume 141-2, chapter Parameter uncertainties and adaptive estimation, pages 68–158. Academic Press.

Najafi, H., Navab, N., and Klinker, G. (2004). Automated initialization for marker-less tracking: a sensor fusion approach. In *ISMAR*, pages 79–88.

Okuma, T., Kurata, T., and Sakaue, K. (2003). 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 *ISMAR*, page 260.

Pupilli, M. and Calway, A. (2005). Real-time camera tracking using a particle filter. In *British Machine Vision Conference*, pages 519–528. BMVA Press.

Satoh, K., Uchiyama, S., Yamamoto, H., and Tamura, H. (2003). Robot vision-based registration utilizing bird's-eye view with user's view. In *ISMAR*, pages 46–55.

Vacchetti, L., Lepetit, V., and Fua, P. (2004). Combining edge and texture information for real-time accurate 3d camera tracking. In *ISMAR*, Arlington, VA.

Xu, X. and Li, B. (2006). Rao-blackwellised particle filter with adaptive system noise and its evaluation for tracking in surveillance. In *Visual Communications and Image Processing (VCIP)*. SPIE.

You, S. and Neumann, U. (2001). Fusion of vision and gyro tracking for robust augmented reality registration. In *IEEE Virtual Reality (VR)*, pages 71–78.

You, S., Neumann, U., and Azuma, R. (1999). Hybrid inertial and vision tracking for augmented reality registration. In *IEEE Virtual Reality (VR)*, pages 260–267.

Zhang, X., Fronz, S., and Navab, N. (2002). Visual marker detection and decoding in AR systems: A comparative study. In *ISMAR*, pages 97–106.