Video-based camera tracking using rotation-discriminative template matching

David Marimon and Touradj Ebrahimi

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Switzerland
`david.marimon@gmail.com, touradj.ebrahimi@epfl.ch`

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. Feature points are dynamically found in scene and used for further tracking. Evaluations on real cases show that the fusion of these two approaches outperforms the individual tracking results. A critical aspect of the feature point-based cue is to robustly recognise the feature points depite rotations of the camera. A novelty of the proposed framework is the use of a rotation-discriminative method to match feature points.

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 [1]. 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* [2] 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 [3, 4]. Their weakness is, in general, the drift during the absence of a stable reference (usually due to features difficult to recognise after perspective distortions). Filter-based camera tracking generally uses available data such as feature points to correct the filtered state. The problem with feature points is to reliably recognise them. Most techniques use descriptors based on the grey-level or colour histogram or directly the intensity (templates) of their neighbourhood [3, 4]. Feature points change their appearance at consecutive frames due to camera motion. Therefore, methods that robustly recognise feature points despite those changes have to be employed.

In this paper, we present a particle-filter based camera tracker. 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 feature points found in the scene. The proposed framework extends the camera traking system presented in [5]. In this previous work, only the corners of the marker are used and the method to recognise feature points is very sensible to rotations of the camera. We propose a novel use of the rotation discriminative template matching (RDTM) method described in [6]. More precisely, this method is employed here to recognise feature points despite large rotations.

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 [7].

The commonly developed fusions are inertial-acoustic and inertial-video [7]. 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. The advantage of using a bottom-up approach such as a marker-based tracker is that drift is automatically reduced each time the detection occurs. Several works have combined marker-based approaches with inertial sensors [8, 9]. [9] 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, [10] and [11] stand out for their robustness to illumination changes and partial occlusions. [10] 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. [11] 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 [1, 12]. Among these, [1] 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 [4]. Nonetheless, this framework takes limited advantage of the filtering framework and still needs the assistance of an inertial sensor.

Recent works have addressed the problem of robustly identifying feature points in camera tracking frameworks [13, 15]. In both cases, the application of invariant descriptors for correct feature point matching has brought important improvements. Sim *et al.* [13] use SIFT features [14], which have high scale and rotation invariance enabling accurate tracking. However, the extraction and description of SIFT features makes the mapping of the scene more complicated. Indeed, the data association of feature points between frames cannot be used in a straightforward manner because the descriptors are scale invariant and hence the features have many different scales. Therefore, the association is done by traversing all the list of feature descriptors. This process has a large computational cost and the overall system runs at 11.9 seconds per frame. Chekhlov *et al.* [15] propose a multi-resolution descriptor based also on SIFT. The approach differs from [13] in that the extraction of feature points is done at a fixed scale. In order to be scale invariant, several SIFT descriptors at different scales are stored for each feature. At runtime, the scale is selected according to camera pose and 3D feature position. Once the scale is selected, the validation can be computed.

Those descriptors differ from the descriptor presented in [6] mainly in the fact that rotation information is lost. We propose to exploit this information during the filter update by associating it to the estimated camera rotation.

3   SYSTEM DESCRIPTION

This section describes the parameters of the 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.

3.1   Particle Filter equations

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 [16]. 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 [17].

Each particle n in the filter represents a possible camera pose

$$T_n = [t_X, t_Y, t_Z, \text{rot}_W, \text{rot}_X, \text{rot}_Y, \text{rot}_Z]_n, \tag{1}$$

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

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

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

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

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)

$$w_n = p(Y|T_n), \tag{4}$$

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.

3.2   Marker-based Cue (MC)

We use the marker-based system provided by [18] 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.

Fig. 1: Square marker used for the MC.

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 [18]. 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 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,

$$p = K \cdot [R|t] \cdot P, \tag{5}$$

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.

Natural feature points in unprepared environments appear in objects at unknown locations. Hence, the 3D location of feature points in the world coordinate frame is generally unavailable. However, the combination framework proposed here admits a certain preparation of the environment, this is, a marker is available. Since the world coordinate frame is fixed to the marker and the real size of the marker is known, the 3D location of any point in the marker is known. We take advantage of this fact and propose to use the corners as feature points in the scene.

Although we have proved in our previous work that these points provide a reliable measurement for camera tracking [5], they might not always be available. For instance, because a corner is occluded by an object or it is outside of the FoV. In this case, it is interesting to have other feature points to rely on. As explained before, in order to constrain the camera pose, the 3D position of a feature point must be available. However, the inverse procedure can also be

done. Indeed, from Equation (5) one deduces that the 3D world coordinates of a point can be computed if the camera pose $[\mathbf{R}|\mathbf{t}]$ is known. Since the filter keeps an estimate of this pose, it is possible to calculate the 3D position of feature points. Once this location is computed, a new feature point can be added to the map of feature points that constrain the camera pose. This process is detailed in [19].

The intensity level and gradient information are chosen as a description of the feature points, for further recognition. Each time a feature point is added to the map the template of its neighbourhood is stored. At this time, rotated versions of this template are generated. The orientation gradient is computed for each of these versions and the information is summarised in a single robust orientation histogram. The final descriptor of a feature point is composed by the histogram and the rotated templates. The amount of rotated versions is proportional to the number of bins in the histogram.

At runtime, the feature points in the map are searched in the video frame. A region is defined around the estimated location of each feature point. Assume, for the moment, that those regions are known. Each region is matched with the corresponding descriptor. The result of this matching is a correlation score together with a bin-wise estimated rotation, for each pixel inside the region. More precisely, the result indicates which rotated version $\Theta(x, y)$ of the template gives the highest correlation $\Psi(x, y)$ at each pixel (x, y). Further details about the descriptor and the RDTM process can be found in [6].

As explained in the next section, the set of correlation scores and estimated rotations is the measurement fed into the filter for update. Each feature point that is positively matched 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 point is occluded by an object; a point is outside of the FoV; the region does not contain the feature point (due to a bad region estimation); or the 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 drift 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 $(Y = \mathrm{MC})$. The likelihood is modelled with a Cauchy distribution centered at the measurement T_{MC}

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

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,
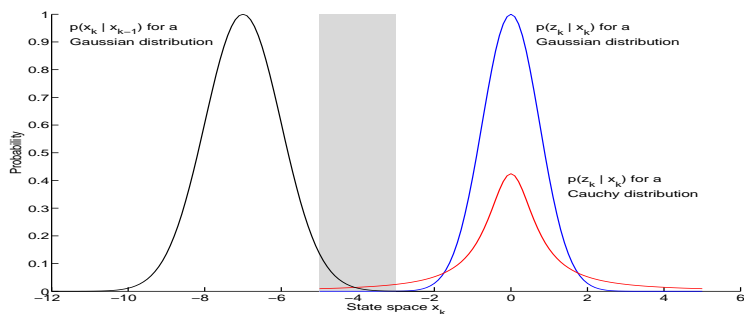

Fig. 2: Overlap between transition prior distribution and the likelihood distribution: modelled with a Gaussian (no overlap) and with a Cauchy distribution (thick line).

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 [20].

On the other hand, when the MC fails to detect the marker, the system relies on the FPC (Y =FPC) and another likelihood is used. As a previous step to looking for the new location of the feature points (see Section 3.3), it is necessary to calculate the *regions* around the estimated location of each feature point. For each feature point, all the back-projections given the transformations T_n are computed (see Eq. 5). The *region* is the bounding box containing all these back-projections. These bounding boxes are fed into the FPC and the matching results are obtained in return. The weights can then be calculated. First, a set of 2D coordinates is obtained by thresholding $\Psi(x, y)$.

$$S_j = \left\{ [c_x, c_y] \big| \Psi_j(c_x, c_y) > th_{corr} \right\}, \tag{7}$$

where j indexes the feature points mapped from the scene. 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 $[p_{n,x}, p_{n,y}]$

$$S_{n,j} = \left\{ [c_x, c_y] \in S_j \big| dist(c, p_n) < th_{dist} \right\}. \tag{8}$$

Finally, the weight is computed. The weight of the particle n is proportional to the correlation Ψ_j achieved in the subsets $S_{n,j}$. Furthermore, this is refined with the orientation Θ_j estimated by the RDTM process. This orientation should have a rough correspondence with the rotation of the camera about the Z axis. The more perpendicular is the original template to the current pose of the camera, the higher the chances of the estimated orientation being similar to the rotation about the Z axis. We take advantage of this fact. Indeed, the weights are forced to be proportional also to the difference between the orientation Θ_j and the rotation around the Z axis of the corresponding particles's state $\psi_{Z,n}$

$$w_n = exp\left(\sum_{j=1}^{L} \sum_{[x,y] \in S_{n,j}} \Psi_j(x,y) \cdot exp - \left(\frac{(\psi_{Z,n} - \hat{\psi}_{Z,j}) - \Theta_j(x,y) \cdot \Delta}{\alpha \cdot \Delta} \right)^2 \right), \tag{9}$$

where L is the number of feature points, $\Delta = 360/N$ is the quantisation step of the orientation according to the number of bins N (see Section 3.3), $\hat{\psi}_{Z,j}$ is the rotation of the camera at the initialisation of the feature point, and α is a tunable parameter. Weighting the particles according to the correlation gives already a strong validation for the data association between feature points and the point in the image plane where they lie. Reinforcing this validation with the orientation permits to avoid confusion with points with high correlation but unexpected orientation according to the camera's pose. Therefore, α can be tuned to vary this reinforcement of the data association. In our case, this parameter is fixed to a high value ($\alpha = N/2$) as the perpendicularity of the camera with respect to the template of a feature point cannot be assured a priori. It is also possible to make this parameter vary according to the angle of rotation in X and Y axes, for instance $\alpha \propto \sum |\psi_{X,n} - \hat{\psi}_{X,j}| + |\psi_{Y,n} - \hat{\psi}_{Y,j}|$. This option is not considered for simplicity purposes.

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.

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.

Algorithm 1 Combination procedure

---

loop
  $vframe \leftarrow$ getVideoFrame()
  $marker \leftarrow$ detectMarker($vframe$)
  if $pattern$.correspondsTo($marker$) **then**
    $T_{MC} \leftarrow$ MC.calcTransformation($marker$)
    $\widehat{T} \leftarrow$ filter.updateFromMC(T_{MC})
  else
    $reg \leftarrow$ filter.calcRegions()
    for $j = 1$ to NumberOfFeaturePoints **do**
      $[\Theta_j, \Psi_j] \leftarrow$ RDTM(reg , $vframe$, descriptors$_j$)
    end for
    $\widehat{T} \leftarrow$ filter.updateFromFPC($\Theta_{j=1...L}$, $\Psi_{j=1...L}$)
  end if
  filter.findNewFeaturePoints($vframe$)
end loop

---

4 EXPERIMENTS

In order to assess the performance of the camera tracking system, we have performed several experiments. Two sequences are used. The first one is generated synthetically. The second one is recorded with a hand-held camera. For the first one, the ground truth is known whereas for the second one a qualitative measure is used. When the camera position with respect to the world coordinate frame is known, it is possible to add virtual objects at a 3D position in the world coordinate space. This is generally known as Augmented Reality. If the alignment between a virtual object and the real scene is fixed, the object should move accordingly to the cameras motion as if it was placed in the real world. A qualitative measure is found by observing how static a fixed virtual object is with respect to the real world.

4.1 Evaluation of the combination of cues

An experiment is conducted to analyse the tracking performance in front of occlusions of the marker. As stated before, one of our goals is to cope with the loss of track of the MC when the marker is occluded. In our framework, tracking can continue by using the FPC. Two techniques are compared in this case. On the one hand, ARToolkit [18], which is equivalent to use the MC alone. On the other hand, our framework combining MC and FPC.

Snapshots from several frames of the augmented sequence are shown in Fig. 3.

4.2 Evaluation of the RDTM for camera tracking

In [6], the RDTM method to recognise regions is described. This method is tailored to match a template despite of a 2D rotation, as well as detect the

(a) Snapshots of a manual occlusion.

(b) Snapshots of a manual occlusion.

(c) Snapshots while the marker is escaping the field of view.

Fig. 3: Experiment with occlusions. A virtual teapot is placed on the marker to show correct alignment. When the teapot is red, the framework uses the MC, whereas when it is green, the framework relies on the FPC.

rotation that the template has undergone. In this paper, experiments have shown the accuracy of the method on several images rotated over the perpendicular axis (2D rotations). We want to evaluate here the improvement brought by the RDTMwhen compared to a simpler but commonly used method [3–5].

Two feature point-based camera trackers with different matching techniques are compared. In the first one, the recognition is performed with the Normalised Cross-Correlation (NCC) of the templates. In the second one, matching of feature points is done with our RDTM method. The experiment is conducted with the synthetic sequence.

Fig. 4 shows one instance of the absolute error of each axis of the compared techniques. Matching with NCC fails as soon as a large rotation around the Z axis occurs (around frame 50). As a consequence, this tracker looses all references and starts to drift. On the other hand, the rotation-discriminative method allows a continuous track of the feature points and hence accurate camera pose estimation. Indeed, the Root Mean Square Error achieved for the Z axis is very low: 0.79 degrees.

5   CONCLUSION

We have presented a combination of video-based camera trackers within a particle filter framework. The filter uses two cues provided by a marker-based approach and a feature point-based one. We introduce a novel use of a rotation-discriminative template matching (RDTM) method for camera tracking.
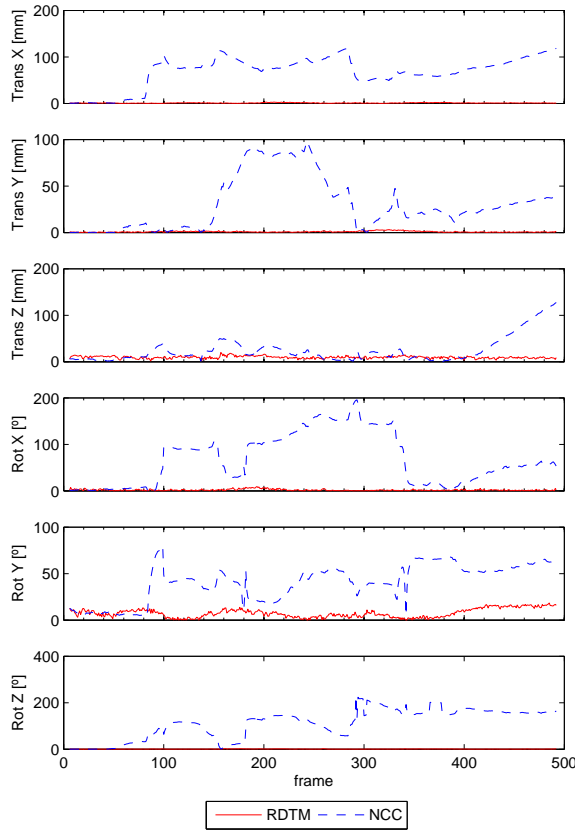
Fig. 4: Experiment with different feature point recognition methods. Comparison between NCC and RDTM. Absolute error of the translation and rotation in X,Y and Z axes (Renens sequence).

Experiments show that the proposed combination produces a synergy. In particular we have shown robustness in front of occlusions of the marker. Moreover, we have demonstrated the convenience of using the RDTM by comparison to other commonly used template matching.

In our future research, we will focus on extending the application of the RDTM to scale invariance by exploiting the knowledge of the estimated distance between the camera and the feature points.

References

1. Okuma, T., Kurata, T., Sakaue, K.: 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: Proc. Intl. Symp. on Mixed and Augmented Reality (ISMAR). (2003) 260
2. Zhang, X., Fronz, S., Navab, N.: Visual marker detection and decoding in AR systems: A comparative study. In: Proc. Intl. Symp. on Mixed and Augmented Reality (ISMAR). (Sep–Oct 2002) 97–106
3. Davison, A.: Real-time simultaneous localisation and mapping with a single camera. In: Proc. Intl. Conf. on Computer Vision (ICCV). (2003)
4. Pupilli, M., Calway, A.: Real-time camera tracking using a particle filter. In: Proc. British Machine Vision Conference (BMVC). (September 2005) 519–528
5. Marimon, D., Ebrahimi, T.: Combination of video-based camera trackers using a dynamically adapted particle filter. In: 2nd Intl. Conf. on Computer Vision Theory and Applications (VISAPP07). (2007)
6. Marimon, D., Ebrahimi, T.: Efficient rotation-discriminative template matching. In Rueda, L., Mery, D., Kittler, J., eds.: 12th Iberoamerican Congress on Pattern Recognition (CIARP). Volume 4756 of Lecture Notes in Computer Science (LNCS)., Springer-Verlag (2007) 221–230
7. Allen, B., Bishop, G., Welch, G.: Tracking: Beyond 15 minutes of thought. In: Course Notes, Ann. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH). (2001)
8. Kanbara, M., Fujii, H., Takemura, H., Yokoya, N.: A stereo vision-based augmented reality system with an inertial sensor. In: Proc. IEEE and ACM Intl. Symp. on Augmented Reality (ISAR). (Oct 2000) 97–100
9. You, S., Neumann, U.: Fusion of vision and gyro tracking for robust augmented reality registration. In: Proc. IEEE Virtual Reality (VR). (2001) 71–78
10. Claus, D., Fitzgibbon, A.: Reliable fiducial detection in natural scenes. In: Proc. European Conference on Computer Vision (ECCV). Volume 3024., Prague, Czech Republic, Springer-Verlag (May 2004) 469–480
11. Fiala, M.: ARTag, a fiducial marker system using digital techniques. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Volume 2., Washington, DC, USA, IEEE Computer Society (2005) 590–596
12. Satoh, K., Uchiyama, S., Yamamoto, H., Tamura, H.: Robot vision-based registration utilizing bird's-eye view with user's view. In: Proc. Intl. Symp. on Mixed and Augmented Reality (ISMAR). (Oct 2003) 46–55
13. Sim, R., Elinas, P., Griffin, M., Little, J.J.: Vision-based SLAM using the Rao-Blackwellised particle filter. In: Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR), Edinburgh, Scotland (2005) 9–16
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. Intl. Journal of Computer Vision **60**(2) (2004) 91–110
15. Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., Calway, A.: Real-time and robust monocular slam using predictive multi-resolution descriptors. In: 2nd International Symposium on Visual Computing. (November 2006)
16. Chai, L., Nguyen, K., Hoff, B., Vincent, T.: An adaptive estimator for registration in augmented reality. In: Proc. IEEE and ACM Intl. Workshop on Augmented Reality (IWAR). (1999) 23–32
17. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. on Signal Processing **50**(2) (Feb. 2002) 174–188

18. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proc. Intl. Workshop on Augmented Reality (IWAR). (Oct 1999) 85–94
19. Marimon, D.: Advances in top-down and bottom-up approaches to video-based camera tracking. PhD thesis, EPFL (2007)
20. Ichimura, N.: Stochastic filtering for motion trajectory in image sequences using a monte carlo filter with estimation of hyper-parameters. In: Proc. Intl. Conf. on Pattern Recognition (ICPR). Volume 4. (2002) 68–73