

Head Motion Prediction in Augmented Reality Systems Using Monte Carlo Particle Filters

Fakhr-eddine Ababsa, Jean Yves Didier, Malik Mallem, David Roussel

Laboratoire Systèmes Complexes. CNRS FRE 2494
40, Rue du Pelvoux, CE1455-Courcouronnes 91020 Evry Cedex France
[ababsa | didier | mallem | roussel] @iup.univ-evry.fr

Abstract

A basic problem with Augmented Reality systems using Head-Mounted Displays (HMDs) is the perceived latency or lag. This delay corresponds to the elapsed time between the moment when the user's head moves and the moment of displaying the corresponding virtual objects in the HMD. One way to eliminate or reduce the effect of system delays is to predict future head locations. Actually, the most used filter to predict head motion is the extended Kalman filter (EKF). However, when dealing with non linear models (like head motion) in state equation and measurement relation and a non Gaussian noise assumption, the EKF method may lead to a non optimal solution. In this paper, we propose to use sequential Monte Carlo methods, also known as particle filters to predict head motion. These methods provide general solutions to many problems with any non linearities or distributions. Our purpose is to compare, both in simulation and in real task, the results obtained by particle filter with those given by EKF.

Key words: Augmented Reality, HMD, latency, dynamic registration, particle filter.

1. Introduction

In Augmented Reality systems using Head-Mounted Displays (HMDs), one of the major problem is the end-to-end system delay (or latency). This delay exists because the head tracker, scene generator, and communication links require time to perform their tasks, causing a lag between the measurement of head location and the display of the corresponding images inside the HMD [1,2,3,4,16]. Therefore, those images are displayed later than they should be, making the virtual objects appear to "lag behind" the user's head movements. This is annoying to users and tends to destroy the illusion that the virtual objects are co-existing with the real objects. One way to compensate for the delay is to predict future head locations. If the system can somehow determine the future head position and orientation for the time when the images will be displayed, it can use that future location to generate the graphic images, instead of using the measured head location. Inertial sensors, such as linear accelerometers and angular rate gyroscopes are widely

used for head motion tracking [12,13,18,20,21]. A filter structure is used to fuse inertial data and then predict the head pose. Actually, the most used filter to estimate head motion is the extended Kalman filter (EKF) [19]. This filter is based upon the principle of linearizing the measurements and evolution models using Taylor series expansions. The series approximations in the EKF algorithm can, however, lead to poor representations of the nonlinear functions and probability distributions of interest. Besides, when dealing with non linear models (like head motion) in state equation and measurement relation and a non Gaussian noise assumption, the EKF method may lead to a non optimal solution. The sequential Monte Carlo methods, or particle filters [5,9,10,14,15], provide general solutions to many problems where linearizations and Gaussian approximations are intractable or would yield too low performance. In this paper, we apply the classical particle filter Bayesian bootstrap [14], to predict head motion. This method allows for a complete representation of the posterior distribution of the states, so that any statistical estimates can be easily computed. It can therefore, deal with any non linearities or distributions. Our purpose is to compare, both in simulation and in real task, the results obtained by particle filter estimator with those given by EKF estimator. Typical head motion trajectories are considered and error prediction for the two estimators are examined.

The remainder of this paper is organized as follows. Section 2 is devoted to the principle of particle filter and to the implementation details of the Bayesian bootstrap filter [14,15]. Section 3 describes the head motion model, including the state representation and system dynamics. Section 4 presents results of our experiments. Finally, section 5 provides conclusions and pointers for future work.

2. Particle filtering method

Particle filtering has been a growing research area lately due to improved computer performance. A rebirth of this type of algorithms came after the seminal paper of Gordon et al. [14], showing that particle filtering methods could be used in practice to solve the optimal estimation problem. The head motion tracking can be

considered as a single target non linear discrete time system:

$$\begin{aligned} x_{t+1} &= f(x_{t-1}, v_t) \\ y_t &= h(x_t) + e_t \end{aligned} \quad (1)$$

The particle filters provide an approximative Bayesian solution to discrete time recursive problem by updating an approximative description of the posterior filtering density. Let $x_t \in \mathfrak{R}^n$ denote the state of the observed system and $Y_t = \{y_i\}_{i=0}^t$ be the set of observations until present time. Assume independent process noise v_t and measurement noise e_t with densities p_{v_t} respective p_{e_t} . The initial uncertainty is described by the density p_{x_0} . The particle filter approximates the probability density $p(x_t|Y_t)$

by a large set of N particles $\{x_t^{(j)}\}_{j=1}^N$, where each particle has an assigned relative weight, $w_t^{(j)}$, such that all weights sum to unity. The location and weight of each particle reflect the value of the density in the region of the state space. The particle filter updates the particle location and the corresponding weights recursively with each new observation. The non-linear prediction density $p(x_t|Y_{t-1})$ and filtering density $p(x_t|Y_t)$ for the Bayesian inference are given by:

$$p(x_t|Y_{t-1}) = \int_{\mathfrak{R}^n} p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1})dx_{t-1} \quad (2)$$

$$p(x_t|Y_t) \propto p(y_t|x_t)p(x_t|Y_{t-1}) \quad (3)$$

The main idea is to approximate $p(x_t|Y_{t-1})$ with

$$p(x_t|Y_{t-1}) \approx \frac{1}{N} \sum_{i=1}^N \delta(x_t - x_t^{(i)}) \quad (4)$$

where δ is the discrete Dirac function. Inserting equation (4) into equation (3) yields a density to sample from. This can be done by using the Bayesian bootstrap or Sampling Importance Resampling (SIR) algorithm from [14], given in Table (1).

The estimate for the particle filter can be calculated as :

$$\hat{x}_t^N = \sum_{i=1}^N w_t^{(i)} x_t^{(i)} \quad (5)$$

Table 1. Bayesian bootstrap (SIR) algorithm

- | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Set $t=0$, generates N samples $\{x_0^{(j)}\}_{j=1}^N$ from the initial distribution $p(x_0)$. 2. Compute the weights $w_t^{(j)} = p(y_t x_t^{(j)})$ and normalize, i.e, $\tilde{w}_t^{(j)} = w_t^{(j)} / \sum_{j=1}^N w_t^{(j)}, j=1, \dots, N$. 3. Generate a new set $\{x_t^{(j*)}\}_{j=1}^N$ by resampling with replacement N times from $\{x_0^{(j)}\}_{j=1}^N$, where $Pr(x_t^{(j*)} = x_t^{(j)}) = \tilde{w}_t^{(j)}$. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 4. Predict new particles, i.e, $x_{t+1}^{(j)} = f(x_t^{(j*)}, v_t)$, $i = 1, \dots, N$ using different noise realization for the particles. 5. Increase t and iterate to item 2. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3. Head motion model and system dynamics

Our AR tracking system consists of a robust vision landmark tracker and an inertial measurement unit (IMU) providing 3D linear acceleration and 3D rate of turn (rate gyro). The 3D coordinate frames used in our study, as illustrated in figure (1), are: {W} the world coordinate system, {C} the camera-centered coordinate system, and {I} the inertial-centered coordinate system. Since the inertial sensor is rigidly mounted to the camera, the transformation between the camera frame {C} and the inertial frame {I} is known.

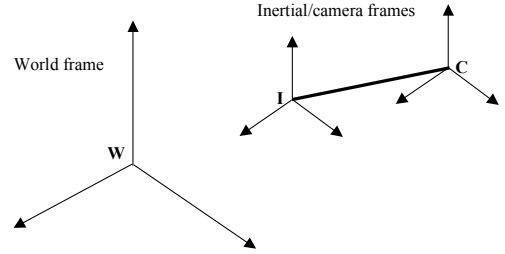


Fig. 1 Frames of reference

We used head motion model proposed by Chai et al. [6,7], where head motion is represented by 15×1 vector:

$$X_{head} = (\theta, \omega, x, \dot{x}, \ddot{x}) \quad (6)$$

where θ is the orientation of the camera frame with respect to the world (we use Z-Y-X Euler angles), ω is the angular velocity, and x, \dot{x}, \ddot{x} are the position, velocity, and acceleration of the camera with respect to the world. With these states, the discretized system dynamics are given by [6]:

$$\begin{bmatrix} \theta_{k+1} \\ \omega_{k+1} \\ x_{k+1} \\ \dot{x}_{k+1} \\ \ddot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} \theta_k + \Delta T \cdot W(\theta) \omega_k + v_k^1 \\ \omega_k + v_k^2 \\ x_k + \Delta T \cdot \dot{x}_k + \frac{1}{2} \Delta T^2 \cdot \ddot{x}_k + v_k^3 \\ \dot{x}_k + \Delta T \cdot \ddot{x}_k + v_k^4 \\ \ddot{x}_k + v_k^5 \end{bmatrix} \quad (7)$$

where ΔT is the sampling period, $W(\theta)$ is the Jacobian matrix that relates the absolute rotation angle to the angular rate [8], and v_k^i are the system random distribution noise.

Otherwise, sensors have associated with them an output equation, which maps the states to the sensor output. Since the goal of our study is to compare performances of a particle filter estimator and an EKF estimator, we suppose that observation equations measure directly the

system state. So the vision system measures the camera pose (θ_m, x_m) in world coordinate system. The gyroscope produces angular velocity measurements ω_m , and the accelerometers produce the acceleration \ddot{x}_m . The sensors measurements can then be represented by a measurement vector as follows:

$$y_m = [\theta_m, \omega_m, x_m, \ddot{x}_m] \quad (8)$$

the measure equation is then given by :

$$Y_k = H \cdot X_k + e_k \quad (9)$$

where

$$H = \begin{bmatrix} I_{3 \times 3} & 0 & 0 & 0 & 0 \\ 0 & I_{3 \times 3} & 0 & 0 & 0 \\ 0 & 0 & I_{3 \times 3} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{3 \times 3} \end{bmatrix} \quad (10)$$

and $I_{3 \times 3}$ is a 3×3 identity matrix, and e_k is the measurements noise.

We used a particle filter as fusion filter [11,17,20,21] to estimate the head pose parameters of (6) from the measurements of the vision and inertial sensors.

4. Results

In order to evaluate the performance of our fusion particle filter, we first implement it with purely synthetic data. So, a synthetic position and orientation of user's head are described (figure 2), and synthetic accelerometer, gyro, and camera data are generated from it. A synthetic white Gaussian random sequence is added to the measurements with variance 0.01 for gyros, 1300 for accelerometers and 0.016 for the vision tracker. Furthermore, the process and measurement noises, v_k and e_k respectively, are assumed gaussian, i.e $v_k \in N(0, Q)$ and $e_k \in N(0, R)$. The data rate of the sensors is $\Delta T = 15.6$ ms.

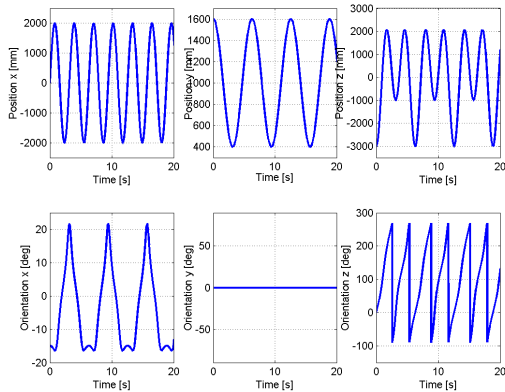


Fig. 2 Synthetic position and orientation data

The filter parameters, i.e. measurement noise matrix R , process noise matrix Q and initial state error matrix P_0 , where chosen as:

$$P_0 = \text{diag} \left(\begin{bmatrix} 100I_{3 \times 3} & 10I_{3 \times 3} & 10^2I_{3 \times 3} & 10^4I_{3 \times 3} & 10^6I_{3 \times 3} \end{bmatrix} \right)$$

$$Q = \text{diag} \left(\begin{bmatrix} 0.01I_{3 \times 3} & 0.1I_{3 \times 3} & 10I_{3 \times 3} & 10^5I_{3 \times 3} & 10^6I_{3 \times 3} \end{bmatrix} \right)$$

$$R = \text{diag} \left(\begin{bmatrix} 0.001I_{3 \times 3} & 0.01I_{3 \times 3} & I_{3 \times 3} & 100I_{3 \times 3} & 10^4I_{3 \times 3} \end{bmatrix} \right)$$

The noise matrices were chosen empirically in order to achieve the best performance of the filter.

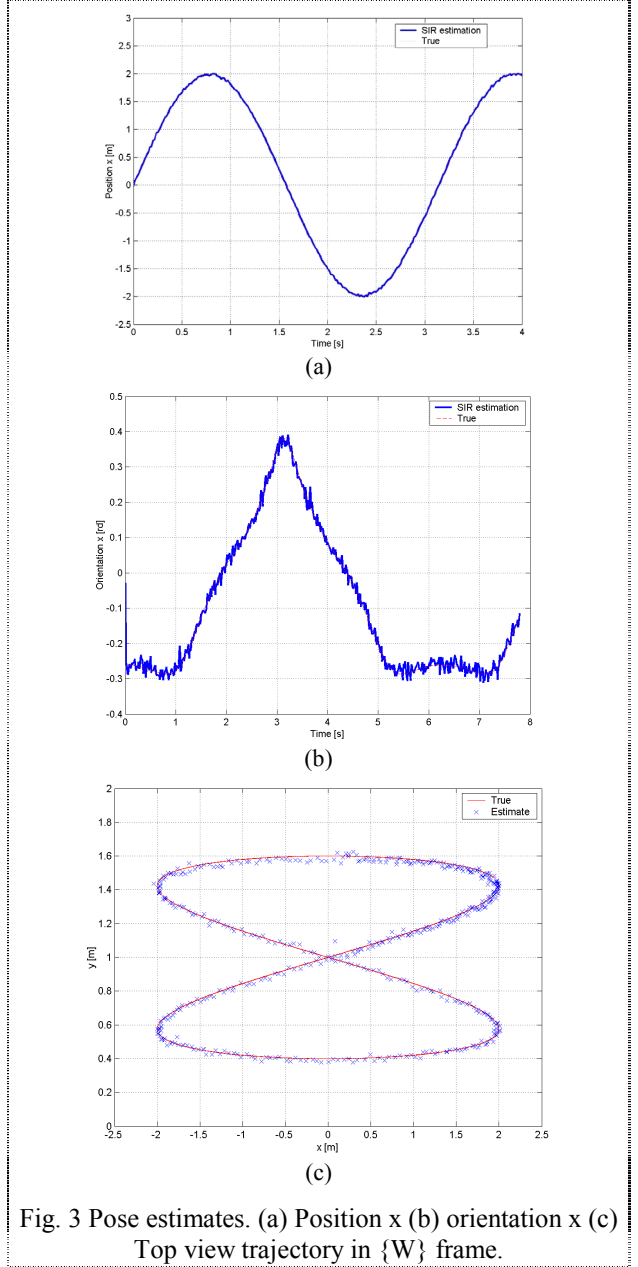


Fig. 3 Pose estimates. (a) Position x (b) orientation x (c) Top view trajectory in $\{W\}$ frame.

First, we compare the SIR method with an EKF filter, implemented using equations (7) and (9). We don't include the extended Kalman filter equations since they can be found in most related textbook, e.g., [19]. In figure (3) results of head motion prediction using SIR filter are presented. The true position and orientation are as a solid line, where the estimates are drawn in dashed lines. Figure (3-a) illustrates the estimated x position, and figure (3-b) is the estimated x orientation. The particle filter used $N = 100$ samples. We note that the implemented particle filter succeeds to follow the true

trajectories very closely, which implies a robust head motion prediction. Figure (3-c) shows a top view of the prediction process in the $\{W\}$ frame. The solid line represents the true location of the camera in the $\{W\}$ frame, whereas the "x" represents the predicted one.

To evaluate the filter performances a *root mean square error* (RMSE) analysis is performed. The RMSE is given by:

$$RMSE(n) = \frac{1}{n} \sum_{k=1}^n \|x(k) - \hat{x}(k)\|^2 \quad (11)$$

where $x(k)$ is the true trajectory, and $\hat{x}(k)$ the predicted one.

In figure (4), the RMSE values for different times are presented for both SIR and EKF filters. Figures (4-a) and (4-b) illustrate the RMSE for the camera position and orientation according to the x axis of the $\{W\}$ frame, respectively. We note that, in the two cases and under the same conditions, performances of the SIR estimator are better than those of the EKF estimator. This foretells a well efficiency when the SIR filter will be implemented in our AR system.

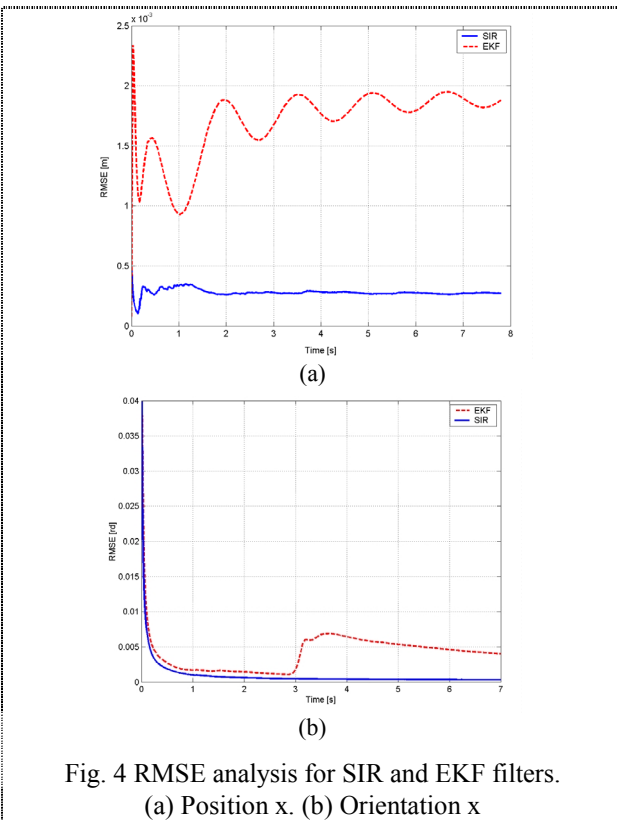


Fig. 4 RMSE analysis for SIR and EKF filters. (a) Position x. (b) Orientation x

In table (2), the results for the two methods are summarized:

Table 2. SIR and EKF RMSE

Estimation	Translation (m)	Orientation (rd)
SIR	0.01642	0.0003
EKF	0.01986	0.0049

In figure (5) the relationship between the SIR performances and the number of particles is presented.

We note that, when the number of particles increases, the RMSE decreases. However, as it is known, the computational burden for the particle filters is dependent on the number of particles [9]. Therefore it is essential to minimize the number of particles used in the estimation step. The number of particles needed is determined so that the computational remained suitable for our real time application, while assuring high performances. So, we have done several experiments to determine the appropriate number of particles, in this case $N = 100$.

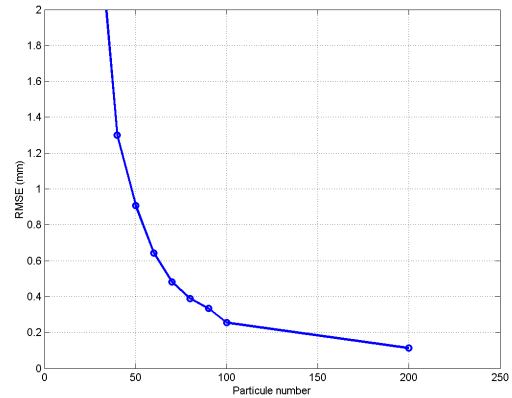


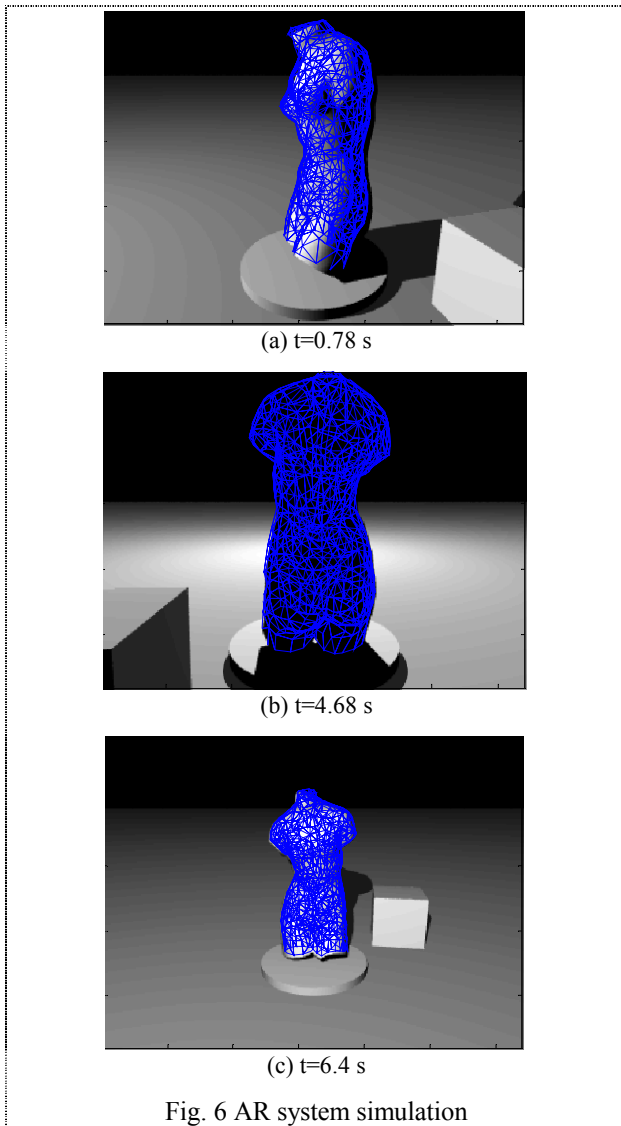
Fig. 5 Effect of the number of particles on SIR performances.

In order to demonstrate the efficiency of our approach, we simulated an AR demonstrated system using our fusion particle filter. We have generated a sequence of images taken by a moving camera. During its motion, the camera observes always the "Venus" object which represents our visual reference object. Since the "Venus" is fixed in the $\{W\}$ frame, it does not move during the process. When the camera pose is predicted by the SIR estimator, we backproject the "Venus" model (the virtual object), on the plane of the corresponding future image. Figure (6) shows the results of this projection for different times in the images sequence. We have noted that, at every time, the virtual model is well superposed to its corresponding scene object. This result shows how well the SIR estimator predict the camera pose.

5. Conclusion

This paper has presented an accurate and an efficient method for AR registration that simultaneously tracks the pose and motion of the user's head. This approach is based on the implementation of a particle filter to integrate inertial and vision subsystems. We have compared our particle filter estimator to a classical EKF estimator implemented under the same conditions. The root mean square error (RMSE) analysis was used to describe performance. The obtained results showed that the particle filter estimator performs better than the EKF estimator. Otherwise, we quantitatively evaluated, in simulation, our AR system under several conditions, and the results showed that the fusion method using particle filter achieves high tracking stability and robustness, and

image projection accuracy is under some pixels over a range of test conditions. Our current investigation is how to tune the filter parameters to achieve optimum performance. Ideally, instead of manually adjusting the parameters based on an intuitive understanding of the process, an automatic procedure would be used such that the optimum performance of the system is guaranteed. One interesting approach is to adaptively adjust the filter parameters based on the observed motion of the user. Indeed, when the person is moving slowly, we have a lower level of process noise than when he is moving rapidly.



References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality". In *IEEE Computer Graphics*

[14] N.J. Gordon, "A hybrid bootstrap filter for target tracking in clutter". In *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, 1997, pp. 353-358.

and *Application*, Vol. 21 no. 6, November 2001, pp. 34-47.

[2] R. Azuma and G. Bishop, "Improving Static and Dynamic Registration in an Optical See-through HMD", *Proc. SIGGRAPH 95*, 1995.

[3] M. Bajura, and U. Neumann, "Dynamic Registration Correction in Augmented Reality Systems". In *Proc. of IEEE Virtual Reality Annual International Symposium*, 1995, pp. 189-196.

[4] R. Behringer, "Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors". In *Proc. of IEEE VR'99*, pp. 244-251.

[5] N. Bergman, and A. Doucet, "Markov Chain Monte Carlo data association for target tracking". In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000.

[6] L. Chai, W. Hoff, and T. Vincent, "Three-Dimensional Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality". *Presence: Teleoperators & Virtual Environments*, 2002, pp. 474-492.

[7] L. Chai, K. Nguyen, B. Hoff, and T. Vincent, "An Adaptive Estimator for Registration in Augmented Reality". In *Proc. of IEEE International Workshop on Augmented Reality*, 1999, pp. 23-32.

[8] J. J. Craig, "Introduction to Robotics: Mechanics and Control". Addison-Wesley, 1989.

[9] A. Doucet, N. de Freitas, and N. Gordon, editors, "Sequential Monte Carlo Methods in Practice". Springer Verlag, 2001.

[10] A. Doucet, N.J. Gordon, and V. Krishnamurthy, "Particle Filters for State Estimation of Jump Markov Linear Systems". In *IEEE Trans. on Signal Processing*, Vol. 49 no. 3, March 2001, pp. 613-624.

[11] S. Emura, and S. Tachi, "Multisensor Integrated Prediction for Virtual Reality". *Presence: Teleoperators & Virtual Environments*, Vol 7. no. 4, 1998, pp. 410-422.

[12] E. Foxlin, and L. Naimark, "VIS-Tracker: A Wearable Vision-Inertial Self-Tracker". In *IEEE Conference on Virtual Reality (VR 2003)*. Los Angeles, CA.

[13] E. Foxlin, "Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter". In *Proc. of IEEE Virtual Reality Annual International Symposium*, 1996, pp. 184-194.

[15] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "A novel approach to nonlinear/non Gaussian Bayesian state estimation". In *IEE Proceedings on Radar and Signal Processing*, Vol. 140, 1993, pp. 107-113.

- [16] W. Hoff, K. Nguyen, and T. Lyon, "Computer Vision-Based Registration Techniques for Augmented Reality". In Proc. of IRCV, SPIE, Vol. 2904, 1996, pp. 538-548.
- [17] K. Meyer, H. Applewhite, and F. Biocca, "A Survey of Position Trackers". Presence: Teleoperators & Virtual Environments, Vol. 1 n.o. 2, 1992, pp. 173-200.
- [18] G. Welch, "Hybrid Self-Tracker: An Inertial/Optical Hybrid Three-Dimensional Tracking System". UNC- Chapel Hill, Dept. of C. S., NC, USA TR95-048, 1995.
- [19] G. Welsh, and G. Bishop, "An Introduction to the Kalman Filter". Technical Report TR95-041, University of North Carolina at Chapel Hill, 1995.
- [20] S. You, and U. Neumann, "Fusion of Vision and Gyro Tracking for Robust Augmented Reality Applications". In Proc. of IEEE VR2001, pp. 71-78.
- [21] S. You, U. Neumann, and R. Azuma, "Hybrid Inertial and Vision Tracking for Augmented Reality".