Recursive Camera Resectioning with Unscented Particle Filter in Image Sequences: Application to Video-based Augmented Reality

Jongsung Kim, Kisang Hong
Pohang University of Science & Technology, Pohang, 790-784 Korea
{kimjs,hongks}@postech.ac.kr

Abstract

In this paper, we propose a new recursive framework for camera resectioning and apply it to off-line video-based augmented reality. Our algorithm is based on an unscented particle filter, which deals with non-linear dynamic systems without local linearization, and leads to more accurate results than other non-linear filters. The proposed approach has some desirable properties. It does not rely on closed-form solutions. It is fairly accurate and is easy to implement as compared with other non-linear approaches. As a result, the proposed algorithm outperforms the standard camera resectioning algorithm. We verify this through experimentation using real image sequences.

Key words: Augmented Reality, Camera Resectioning, Unscented Particle Filter

1. Introduction

Computing a camera matrix from known or reconstructed 3D structure and corresponding image locations is called a linear camera calibration or camera resectioning in the vision community [1, 4, 15, 18]. Camera resectioning frequently utilized in frame-sampling based applications, e.g. structure and motion analysis [7, 11, 15, 20], and off-line video-based augmented reality [21, 23] etc. The frame-sampling based approaches are mainly composed of two parts, the Euclidean reconstruction from key-frames and the camera motion estimation of all frames from the reconstructed 3D structure, i.e. camera resectioning.

For projective camera resectioning, linear least-squares methods give reasonable solutions if data are appropriately preprocessed [4]. However, in the Euclidean case, although we have closed-form solutions for camera resectioning [1, 18], the intrinsic parameters computed by these methods do not always satisfy the intrinsic parameter constraints, i.e. zero skew, unit aspect ratio and fixed optical center. A general remedy for this problem is to fix the intrinsic parameters with known values and then re-estimate the unfixed intrinsic and extrinsic parameters for the consistency of estimation results. For this purpose, we should minimize a non-linear error cost function (e.g. mean squares re-projection error) by using iterative optimization techniques [18]. However, this approach has a drawback in that it only works when the initial solutions are close to true ones. Our research focuses on this problem.

Contrary to other approaches, we consider a recursive framework which efficiently uses the latest information for prediction [3, 19]. Our algorithm is based on the unscented particle filter [6, 14, 16, 22], which was presented as an alternative to extended Kalman filter and achieves a better level of accuracy at a comparable level of complexity.

The organization of this paper is as follows. Section 2 introduces our image sequence analysis system. Section 3 presents our dynamic state space model for the camera motion estimation problem. Section 4 describes our recursive framework for camera resectioning. Experimental results are given in Section 5 and our conclusion in Section 6.

2. Image Sequence Analysis System

2.1. System Overview

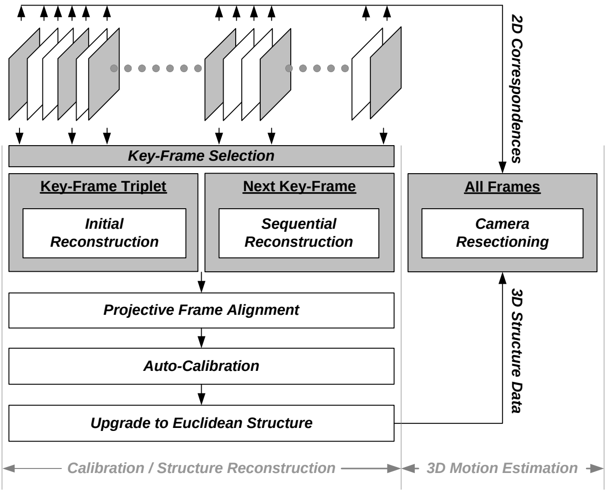


Fig. 1 Frame-sampling based auto-calibration, structure and motion analysis system in image sequences

Our system for auto-calibration, structure and motion analysis in uncalibrated image sequences is presented in Fig. 1. Similar systems have been introduced by D.

Nister [16] and B. Georgescu et al. [20]. Our system is composed of two main parts; Euclidean structure estimation from key-frames and motion estimation by camera resectioning.

2.2. Auto-Calibration and Euclidean Structure Reconstruction from Key-Frames

In our system, selecting and tracking image features from image sequences is conducted by the Kanade-Lucas-Tomasi algorithm [2]. To automatically select key-frames we use the frame decimation algorithm [16] or key-frame selection algorithms [21, 23]. In our experiments we manually select key-frames. From the first three key-frames, we reconstruct the initial 3D structure and three projective cameras by using the trifocal tensor constraint [15]. We sequentially merge the next key-frame to the first three key-frames using the reconstructed projective structure. After completing the projective reconstruction from key-frames in image sequences, we minimize the estimation error with the projective bundle adjustment, and then we upgrade the projective reconstruction to the Euclidean reconstruction by using the auto-calibration technique. We apply the Euclidean bundle adjustment to minimize the calibration error. Refer to Pollefeys et al. [10, 11] and Triggs et al. [12] for details on *auto-calibration* and *bundle-adjustment*, respectively.

2.3. 3D Motion Estimation of Moving Camera by Camera Resectioning

From the 2D correspondences and the 3D structure computed in the first step, we estimate the 3D motion of moving camera in all frames. This procedure is called linear calibration or camera resectioning. The estimated camera parameters are used in the off-line augmented reality system developed in our laboratory.

Linear and non-linear estimation techniques for camera intrinsic and extrinsic parameters have been introduced in many vision materials [1, 15, 18]. Most of conventional approaches are based on linear least squares methods. However, linear solutions are not adequate for the augmented reality system where estimated cameras should satisfy some constraints, i.e. zero skew, unit aspect ratio, and fixed optical center. In the previous work this problem has never been directly considered [21, 23]. A general solution for this problem is to fix the intrinsic parameters with known values and then re-estimate the unfixed intrinsic parameters and extrinsic parameters. For the consistency of the estimation results, we should reduce the estimation error arising from the blind parameter fixing by using iterative optimization techniques [18]. However, this approach has a drawback that it works only when the linear solutions are close to true ones. Our camera resectioning algorithm solves this problem by using the state space model and the recursive estimation.

3. Problem Formulation

We adopt a dynamic state-space model with parameters to represent the camera motion. The global rotation $\Omega \in SO(3)$ and the global translation T of the camera are defined as the system states, and written by

$$\mathbf{x} = \{\Omega, T\}. \tag{1}$$

Associated to each motion Ω , T , there are time-varying parameters, i.e. angular velocity ω , linear velocity V , angular acceleration $\dot{\omega}$, and linear acceleration $\dot{V}$. We use $\boldsymbol{\theta}$ as the notation for the system parameters, and define as

$$\boldsymbol{\theta} = \left\{ f, \omega, \dot{\omega}, V, \dot{V}, \Sigma_{\dot{\omega}}, \Sigma_{\dot{V}}, X^1, \ldots, X^N \right\} \tag{2}$$

where f is a focal length, $\Sigma_{\dot{\omega}}$, $\Sigma_{\dot{V}}$ static parameters for the covariance matrix of $\dot{\omega}$, $\dot{V}$, respectively, and $X^1, \ldots, X^N$ 3D points of the static scene reconstructed in the first step of our image sequence analysis system. The time evolution model for the system states and the system parameters is given by

$$f_{t+1} = f_t \tag{3}$$

$$X_{t+1}^{(i)} = X_t^{(i)} \qquad i = 1, \ldots, N \tag{4}$$

$$\Omega_{t+1} = \log_{SO(3)}\left(e^{\hat{\omega}_t} e^{\Omega_t}\right) \tag{5}$$

$$T_{t+1} = e^{\hat{\omega}_t} T_t + V_t \tag{6}$$

$$\omega_{t+1} = \omega_t + \dot{\omega}_t \qquad \dot{\omega}_t \sim N\left(0, \Sigma_{\dot{\omega}}\right) \tag{7}$$

$$V_{t+1} = V_t + \dot{V}_t \qquad \dot{V} \sim N\left(0, \Sigma_{\dot{V}}\right) \tag{8}$$

where $\hat{\omega}$ is the skew symmetric matrix of angular velocity ω , and $\log_{SO(3)}$ the inverse of Rodrigues' formula [19]. The measurement equation is given by

$$\tilde{\mathbf{y}}_t = \left(x_{t1}, y_{t1}, \ldots, x_{tn}, y_{tn}\right)^T = \mathbf{h}\left(\mathbf{x}_t, \boldsymbol{\theta}_t\right) + \mathbf{n}_t \tag{9}$$

where the measurement noise $\mathbf{n}_t \sim N\left(0, \Sigma_{\mathbf{n}_t}\right)$ and $\mathbf{h}(\cdot)$ is the $2N$ vector of corresponding non-linear equation of the perspective camera projection, defined by

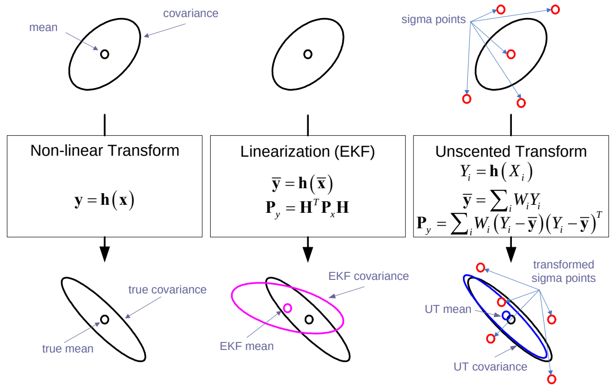
$$\left(x_t^{(i)}, y_t^{(i)}\right)^T = \left(f \frac{\left[X_t^{(i)\prime}\right]^1}{\left[X_t^{(i)\prime}\right]^3}, f \frac{\left[X_t^{(i)\prime}\right]^2}{\left[X_t^{(i)\prime}\right]^3} \right)^T \tag{10}$$

where $X_t^{(i)\prime} = e^{\Omega_t} X_t^{(i)} + T_t$ and $[\cdot]^i$ i th element. In our problem formulation, we assume that the camera intrinsic parameters are fixed with known values. Contrary to the conventional approaches, this assumption makes no error in our recursive camera resectioning algorithm.

4. Recursive Camera Resectioning

4.1. Propagating Mean and Covariance of Camera System State by Unscented Kalman Filter

Unscented transform [6] is a method for propagating mean and covariance with second order accuracy in a nonlinear system. This transform was applied to the extended Kalman filter and called as *unscented Kalman filter* (UKF) by E. A. Wan et al. [14]. We include Fig. 2 to visually illustrate the idea of the unscented transform and to support the understanding the UKF-based part of our algorithm. In this algorithm, the mean and the covariance of the n -dimensional state is represented with $2n+1$ weighted samples, called sigma points.



mean
covariance
sigma points
Non-linear Transform
$\mathbf{y}=\mathbf{h}(\mathbf{x})$
Linearization (EKF)
$\bar{\mathbf{y}}=\mathbf{h}(\bar{\mathbf{x}})$
$\mathbf{P}_y=\mathbf{H}^T\mathbf{P}_x\mathbf{H}$
Unscented Transform
$Y_i=\mathbf{h}(X_i)$
$\bar{\mathbf{y}}=\sum_i W_i Y_i$
$\mathbf{P}_y=\sum_i W_i(Y_i-\bar{\mathbf{y}})(Y_i-\bar{\mathbf{y}})^T$
true covariance
true mean
EKF covariance
EKF mean
transformed sigma points
UT mean
UT covariance

Fig. 2 Mean and covariance propagation: non-linear transform (left), extended Kalman filter (center), unscented transform (right)

Through UKF algorithm we predict the mean $\bar{\mathbf{x}}_t^{UKF}$ and covariance Σ_t^{UKF} at time t of the camera system state $\mathbf{x}_t$ in (1). The UKF-based prediction algorithm can be described in the following way:

Calculate $2n+1$ sigma points $\left\{X_{t-1}^{(0)},\ldots,X_{t-1}^{(2n+1)}\right\}$ of the camera system state using the mean and the covariance at time $t-1$:

$$X_{t-1}^{(0)}=\bar{\mathbf{x}}_{t-1} \qquad W^{(0)}=k/(n+k) \quad (11)$$

$$X_{t-1}^{(i)}=\bar{\mathbf{x}}_{t-1}+\left(\sqrt{(n+k)\Sigma_{t-1}}\right)^{(i)} \qquad W^{(i)}=1/2(n+k) \quad (12)$$

$$X_{t-1}^{(i+n)}=\bar{\mathbf{x}}_{t-1}-\left(\sqrt{(n+k)\Sigma_{t-1}}\right)^{(i)} \qquad W^{(i+n)}=1/2(n+k) \quad (13)$$

where $\left(\sqrt{A}\right)^{(i)}$ is i th singular vector of the matrix A and $k=2$ and $n=6$.

Predict the mean and the covariance of the system state from $2n+1$sigma points using the time evolution model in (3) ~ (8) (we denote a symbol $\mathbf{g}$ to represent the evolution model), and the measurement equation in (9):

$$X_{t|t-1}^{(i)}=\mathbf{g}\left(X_{t-1}^{(i)},\boldsymbol{\theta}_t\right) \qquad i=1,\ldots,2n+1 \quad (14)$$

$$\bar{\mathbf{x}}_{t|t-1}=\sum_{i=0}^{2n+1}W^{(i)}X_{t|t-1}^{(i)} \quad (15)$$

$$\Sigma_{t|t-1}^{xx}=\sum_{i=0}^{2n+1}W^{(i)}\left(X_{t|t-1}^{(i)}-\bar{\mathbf{x}}_{t|t-1}\right)\left(X_{t|t-1}^{(i)}-\bar{\mathbf{x}}_{t|t-1}\right)^T \quad (16)$$

$$Y_{t|t-1}=\mathbf{h}\left(X_{t|t-1}^{(i)},\boldsymbol{\theta}_t\right)+\mathbf{n}_t \quad (17)$$

$$\bar{\mathbf{y}}_{t|t-1}=\sum_{i=0}^{2n+1}W^{(i)}Y_{t|t-1}^{(i)} \quad (18)$$

Update the predicted mean and the predicted covariance by innovation information:

$$\Sigma_{t|t-1}^{yy}=\sum_{i=0}^{2n+1}W^{(i)}\left(Y_{t|t-1}^{(i)}-\bar{\mathbf{y}}_{t|t-1}\right)\left(Y_{t|t-1}^{(i)}-\bar{\mathbf{y}}_{t|t-1}\right)^T \quad (19)$$

$$\Sigma_{t|t-1}^{xy}=\sum_{i=0}^{2n+1}W^{(i)}\left(X_{t|t-1}^{(i)}-\bar{\mathbf{x}}_{t|t-1}\right)\left(Y_{t|t-1}^{(i)}-\bar{\mathbf{y}}_{t|t-1}\right)^T \quad (20)$$

$$K_t=\Sigma_{t|t-1}^{xy}\left(\Sigma_{t|t-1}^{yy}\right)^{-1} \quad (21)$$

$$\bar{\mathbf{x}}_t^{UKF}=\bar{\mathbf{x}}_{t|t-1}+K_t\left(\mathbf{y}_t-\bar{\mathbf{y}}_{t|t-1}\right) \quad (22)$$

$$\Sigma_t^{UKF}=\Sigma_{t|t-1}^{xx}-K_t\Sigma_{t|t-1}^{yy}K_t^T \quad (23)$$

This procedure enables to generate samples from the predicted modes of the proposal distribution, which is called *UKF proposal distribution* [16].

4.2. Bayesian Filtering of Camera System State by Unscented Particle Filter with Independent Metropolis-Hastings Chain

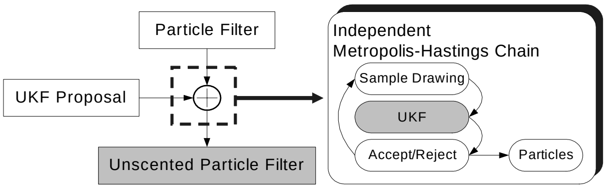


Fig. 3 Unscented particle filter with independent Metropolis-Hastings chain sampling

In Fig. 3, we show the block diagram of our filter design for the Bayesian filtering of camera system state. *Unscented particle filter* (UPF) [16] is a particle filter based on *sequential importance sampling* [8] and the unscented Kalman filter. R. van der Merwe et al. [14] showed that UPF outperforms standard particle filtering and other non-linear filtering methods.

For dynamic systems, the importance proposal can be modeled with a mixture of Gaussian distributions and are obtained by a bank of unscented Kalman filters [22]. Because a moving camera is a dynamic system, we adopt the Gaussian mixture proposal distribution. This approach allows for a reduction in the number of particles. Our importance proposal distribution, a mixture of M Gaussian distributions is given by

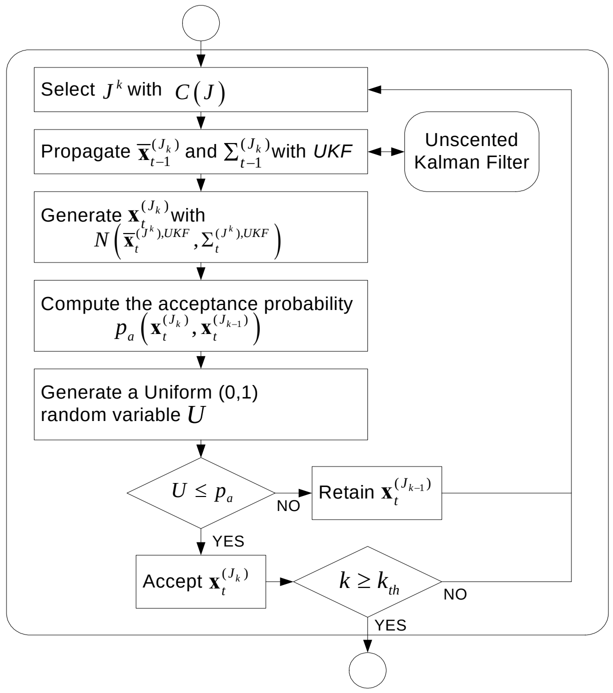

Select J^k with $C(J)$
Propagate $\overline{\mathbf{x}}_{t-1}^{(J_k)}$ and $\Sigma_{t-1}^{(J_k)}$ with *UKF*
Unscented Kalman Filter
Generate $\mathbf{x}_t^{(J_k)}$ with $N\left(\overline{\mathbf{x}}_t^{(J^k),UKF}, \Sigma_t^{(J^k),UKF}\right)$
Compute the acceptance probability $p_a\left(\mathbf{x}_t^{(J_k)}, \mathbf{x}_t^{(J_{k-1})}\right)$
Generate a Uniform (0,1) random variable U
$U \le p_a$
NO
Retain $\mathbf{x}_t^{(J_{k-1})}$
YES
Accept $\mathbf{x}_t^{(J_k)}$
$k \ge k_{th}$
NO
YES

Fig. 4 Independent Metropolis-Hasting Chain algorithm incorporated with unscented Kalman filter

$$g_t\left(\mathbf{x}_t \middle| \mathbf{x}_{0:t-1}\right) = \sum_{j=1}^{M} w_{t-1}^{(j)} N\left(\mathbf{x}_t \middle| \overline{\mathbf{x}}_t^{(j),UKF}, \Sigma_t^{(j),UKF}\right). \quad (24)$$

Drawing a sample according to the proposal in (24) has the following four steps:

Select J th component from the Gaussian mixture proposal distribution in (24) with probability proportional to the weighting factor w_{t-1} , which is represented with the cumulative distribution function (CDF) given by

$$C(J) = \sum_{j=1}^{J} w_{t-1}^{(j)}. \quad (25)$$

Predict the mean $\overline{\mathbf{x}}_t^{(J),UKF}$ and the covariance $\Sigma_t^{(J),UKF}$ by the way described in Section 4.1

Draw a sample according to the proposal distribution written by

$$g_t^{(J)}\left(\mathbf{x}_t \middle| \mathbf{x}_{0:t-1}\right) = N\left(\mathbf{x}_t \middle| \overline{\mathbf{x}}_t^{(J),UKF}, \Sigma_t^{(J),UKF}\right) \quad (26)$$

Accept or Reject the sample using rejection criteria.

This procedure is repeatedly conducted until the generated sample is accepted. This algorithm is called as rejection algorithm. It is well-acknowledged that this algorithm is restrictive and inefficient [8]. For example, in rejection algorithm we should repeat the sampling procedure until one sample is accepted. To improve the efficiency and the convergence of the sampling algorithm, we adopt *independent Metropolis-Hastings chain* (IMHC) [8]:

Generate a sample X *from* $g(\cdot)$
Generate a Uniform (0, 1) random variable U
If $U \le \min\left[1, \dfrac{f(X)g(X')}{f(X')g(X)}\right]$ *accept* X
else set X *equal to* X'

where X' is the previous value of X . This method has many preferable properties. It achieves re-sampling effect automatically and also avoids weight estimation. Re-sampling is necessary to evolve the system for time t to $t+1$ and to prevent the proposal distribution from becoming skewed. Our sampling procedure is illustrated in Fig. 4. Through this procedure, we generate M samples.

We summarize the main part of our recursive camera resectioning algorithm:

<u>Recursive Camera Resectioning Algorithm</u>

Iterate for $J = 1, \dots, M$

Draw $X_t = x_t^{(J)}$ from

$$g_t^{(J)}\left(\mathbf{x}_t \middle| \mathbf{x}_{0:t-1}\right) = N\left(\mathbf{x}_t \middle| \overline{\mathbf{x}}_t^{(J),UKF}, \Sigma_t^{(J),UKF}\right) \quad (27)$$

by UKF and IMHC approach where the rejection probability can be computed as

$$p_a\left(\mathbf{x}_t^{(J)}, \mathbf{x}_t^{(J_k)}\right) = \min\left(1, \frac{f\left(\mathbf{y}_t \middle| \mathbf{x}_t^{(J)}, \boldsymbol{\theta}_t\right) w_{t-1}^{(J_k)}}{f\left(\mathbf{y}_t \middle| \mathbf{x}_t^{(J_k)}, \boldsymbol{\theta}_t\right) w_{t-1}^{(J)}}\right), \quad (28)$$

and the likelihood function as

$$f\left(\mathbf{y} \middle| \mathbf{x}, \boldsymbol{\theta}\right) = \exp\left\{-\left(\mathbf{y} - \tilde{\mathbf{y}}\right)^T \Sigma_n^{-1} \left(\mathbf{y} - \tilde{\mathbf{y}}\right)\right\}. \quad (29)$$

Compute the incremental weight

$$u_t^{(J)} = f\left(\mathbf{y}_t \middle| \mathbf{x}_t^{(J)}, \boldsymbol{\theta}_t\right) q\left(\mathbf{x}_t^{(J)} \middle| \mathbf{x}_{t-1}^{(J)}\right), \quad (30)$$

and let $w_t^{(J)} = u_t^{(J)} w_{t-1}^{(J)}$.

Normalize so that $\sum_j w_t^{(J)} = 1$.

Estimate the mode of the system state as like

$$E\left[X_t \middle| Y_t\right] \approx \sum_{J=1}^{M} w_t^{(J)} \mathbf{x}_t^{(J)} . \quad (31)$$

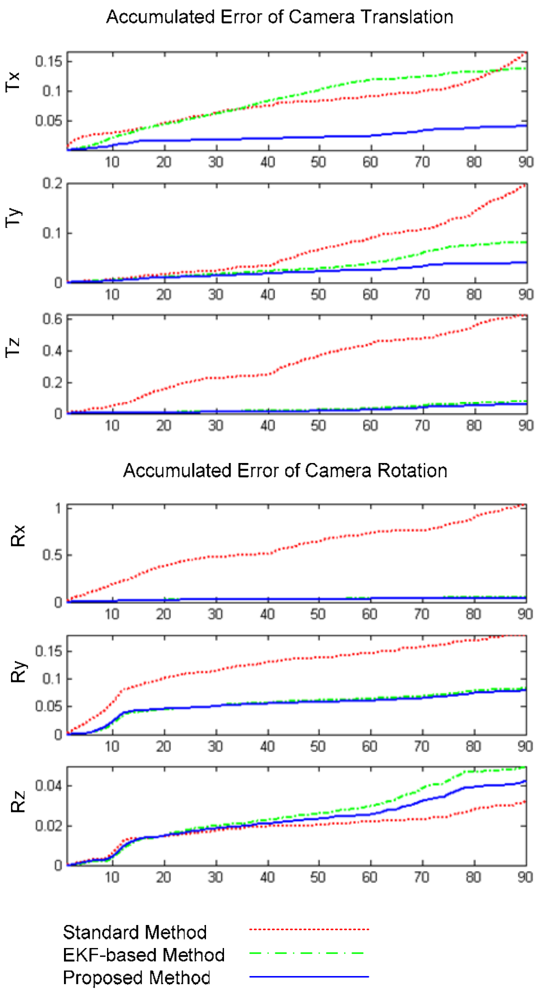


Fig. 5 Comparison of accumulated estimation error: camera translation (top), camera rotation (bottom)

5. Experiments

We tested our algorithm on an image sequence of 90 frames. Three key-frames were selected manually for this experiment. 3D structure points, image correspondences and camera intrinsic parameters were computed by our system presented in Fig. 1. We acquired about 209 features at each frame, 706 3D scene points, and the estimated focal length is 1001. We used 30 particles for UPF, i.e. M =30, 5 iterations for IMHC, i.e. k_{th} =5. We experimentally determined the values of system parameters as $\sigma_{\dot{\omega}}$ =0.0005, $\sigma_{\dot{v}}$ =0.003 and $\sigma_{\mathbf{n}}$ =0.005, assuming that $\Sigma_{\dot{\omega}} = \sigma_{\dot{\omega}}^2 I$, $\Sigma_{\dot{v}} = \sigma_{\dot{v}}^2 I$ and $\Sigma_{\mathbf{n}} = \sigma_{\mathbf{n}}^2 I$. Initial means for camera system state were all zero, i.e. $\overline{\mathbf{x}}_0^{(1)} = \ldots = \overline{\mathbf{x}}_0^{(M)} = \mathbf{0}$ but initial covariance matrices were initialized as $\Sigma_0^{(1)} = \ldots = \Sigma_0^{(1)} = \mathrm{diag}\left(\Sigma_{\dot{\omega}}\ \Sigma_{\dot{v}}\right)$. In Figs. 5 and 8, we compared our method with the standard method described in Section 2, the EKF-based method [3] and the non-linear method [18]. The estimation error depicted in Fig. 5 is the absolute difference of the estimated values between the non-linear method and other methods.

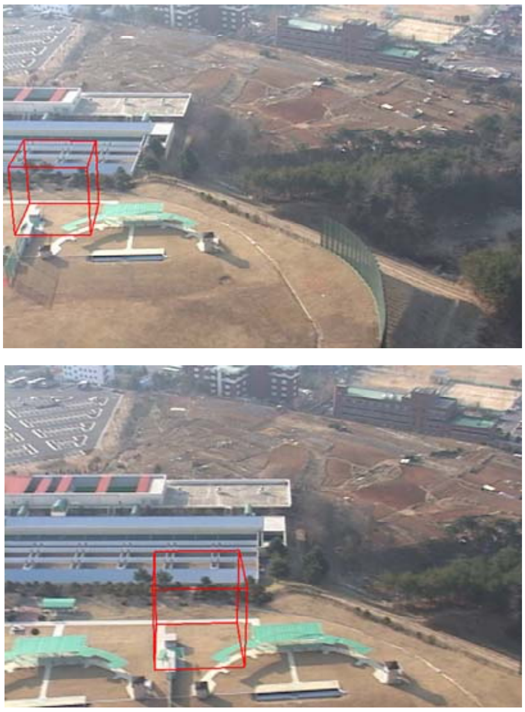


Fig. 6 Video augmentation of bounding box: the first frame (top), the last frame (bottom)

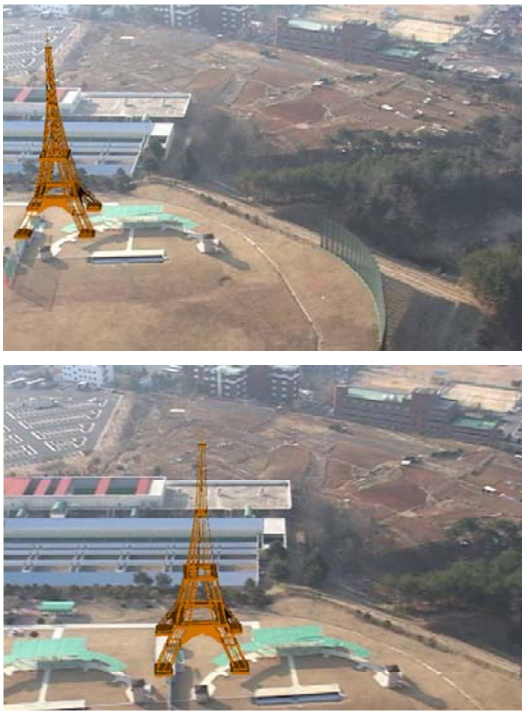


Fig. 7 Video augmentation of graphic object: the first frame (top), the last frame (bottom)

In Figs. 6 and 7, we illustrated the video augmentation results to show that our camera resectioning algorithm was successfully applied to augmented reality. In Figs. 5 and 8, we can see that our method outperforms the standard method and the EKF-based method, and gives results comparable to the non-linear method.

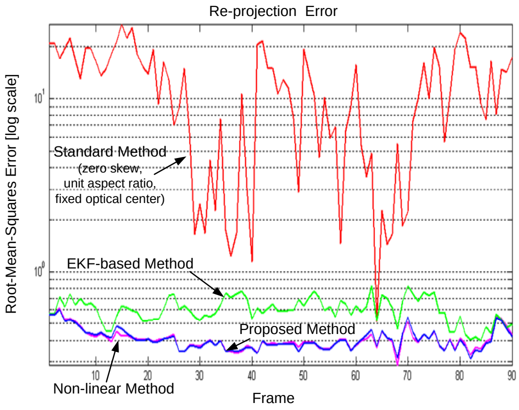


Fig. 8 Comparison of re-projection error: standard method (red), EKF-based method (green), non-linear method (magenta), proposed method (blue)

6. Conclusion

We proposed a new recursive framework for camera resectioning. The proposed framework is based on a unscented particle filter with independent Matropolis-Hastings chain. This algorithm enables the propagation of the mean and the covariance of the previous camera system state with second order accuracy. In this framework, the proposal distribution is a mixture of a Gaussian distribution. This makes it possible to use a small number of particles. To efficiently generate samples from the proposal, we use an independent Matropolis-Hastings chain sampling algorithm. As a result, the proposed algorithm gives results comparable to non-linear methods. It does not rely on erroneous closed-form solutions. Experimentally, we showed that our algorithm outperforms the standard camera resectioning algorithm.

References

1. O. Faugeras, " Three Dimensional Computer Vision," MIT Press, pp.52-58, 1993.
2. J. Shi and C. Tomasi, " Good Features to Track," In Proc. CVPR, pp.593-600, 1994.
3. A. Azerbayejani and A. Pentland, " Recursive Estimation of Motion, Structure, and Focal Length," IEEE Trans. on PAMI, vol.17, no.6, 1995.
4. R. Hartley, " In Defence of the 8-Point Algorithm," In Proc. ICCV, pp.1064-1070, 1995.
5. P. Beardsley, P. Torr and A. Zisserman, " 3D Model Acquisition from Extended Image Sequences," In Proc. ECCV, pp.683-695, 1995.
6. J. Julier and J. K. Uhlman, "A New Extension of the Kalman Filter to Nonlinear System," In Proc. Of AeroSense: The Int. Symbosium on Aerospace/Defence Sensing, Simulation and Controls, Orlando, Florida, 1997.
7. A. W. Fitzgibbon and A. Zisserman, " Automatic Camera Recovery for Closed and Open Image Sequences, " In Proc. ECCV, pp.311-326, 1998.
8. J. S. Liu and R. Chen, " Sequential Monte Carlo Methods for Dynamic Systems," Journal of the American Statistical Association 93: 1032-1044, 1998.
9. P. Torr, A. Fitzgibbon and A. Zisserman, " The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences," Int'l Journal of Computer Vision, vo1.32, no.1, pp.27-44, 1999.
10. M. Pollefeys, R. Koch, and L. Van Gool, " Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters," Int'l Journal of Computer Vision 32(1):7-25, 1999.
11. M. Pollefeys, " Tutorial on 3D Modeling from Images," ECCV, 2000.
12. B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon, " Bundle Adjustment - A Modern Systhesis," 2000, In Vision Algorithms: Theory and Practice, LNCS, pp. 298-375, 2000.
13. G. Simon, A. Fitzgibbon, and A. Zisserman, " Markerless Tracking using Planar Structures in the Scene," Proc. In Proc. ISAR, 2000.
14. E. A. Wan and R. van der Merwe, " The Unscented Kalman Filter for Nonlinear Estimation," In Proc. Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPC), 2000.
15. R. Hartley and A. Zisserman, " Multiple View Geometry in Computer Vision," Cambrige Univ. Press, 2000.
16. D. Nister, " Frame Decimation for Structure and Motion," In Proc. SMILE 2000, LNCS, vol.2018, pp.17-34, 2001.
17. R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan, " The Unscented Particle Filter," In Proc. NIPS, 2001.
18. D. A. Forsyth and J. Ponce, " Computer Vision: A Modern Approach, " Prentice Hall, 2002.
19. A. Chiuso, P. Favaro, H. Jin and S. Soatto, " Structure from Motion Causally Integrated Over Time," IEEE Trans. on PAMI, Vol.24, No.4, 2002.
20. B. Georgescu and P. Meer, " Balanced Recovery of 3D Structure and Camera Motion from Uncalibrated Image Sequences, " LNCS 2351, pp.294-308, 2002.
21. S. Gibson, J. Cook, T. Howard, R. Hubbold and D. Oram, " Accurate Camera Calibration for Off-line, Video-based Augmented Reality," In Proc. IEEE and ACM ISMAR, 2002.
22. R. van der Merwe and E. Wan, " Gaussian Mixture Sigma-Point Particle Filters for Sequential Probabilistic Inference in Dynamic State-Space Models," In Proc. Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2003.
23. J. Seo, S. Kim, C. Jho, and H. Hong, " 3D Estimation and Keyframe selection for Mach Move," In Proc. of ITC-CSCC, 2003.