Improvement of Accuracy for 2D Marker-Based Tracking Using Particle Filter

Yuko Uematsu Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
{yu-ko, saito}@ozawa.ics.keio.ac.jp

Abstract

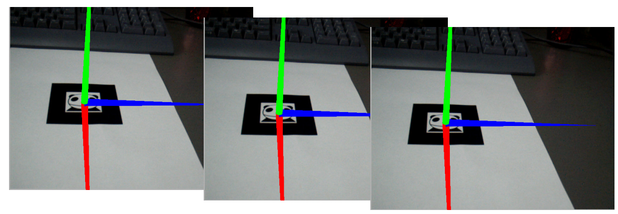
This paper presents a method for improving accuracy of marker-based tracking using a 2D marker for Augmented Reality. We focus on that tracking becomes unstable when the view direction of the camera is almost perpendicular to a marker plane. Especially, tracking of Z axis which is perpendicular to the marker plane (X-Y) becomes unstable. For improving tracking accuracy in this case, we search rotation parameters that are the fittest to projected pattern based on the particle filter. By using particle filtering technique, then, our method can correctly estimate rotation parameters of the camera which are important to track the 3D coordinate system and improve the accuracy of the 3D coordinate system. This method can reduce jitters between frames, which is a big problem in AR. In the experiment, we demonstrate that our method can improve the tracking accuracy of the 3D coordinate system compared with just using ARToolkit.

1. Introduction

Augmented Reality(AR) is a technique for overlaying virtual objects onto the real world, which is achieved by superimposing 3D CG objects onto 2D images of the real world. We can see the virtual objects as if they really exist in the real world. Therefore AR can provide users very effective views by giving some additional information [1].

In Augmented Reality, one of the most important issues for mixing the real and virtual world smoothly is geometrical registration between a 3D coordinate system in the real world and a 2D coordinate system of the input images. To overlay the virtual objects onto input images of the real world correctly, we have to track the camera motion and align the virtual object according to the motion frame by frame.

A 2D (planar) rectangular marker like ARToolkit [3] is very popular tool for the camera tracking in AR. Marker-based method can estimate rotation and translation of the camera in real-time by extracting the contour of the rectangular from the input images and estimating position and pose of the marker.



(a) Images captured from an angle. 3D coordinate system is easily tracked from these images.

(b) Images captured in which marker plane is almost perpendicular to optical axis of a camera. It's difficult to track 3D coordinate system, especially Z axis.

Figure 1. Example images on tracking 3D coordinate system.

When estimating the rotation and translation parameters of the camera by using such markers, it is important to detect the change of 2D appearance of the marker in the input images. However detection errors of 2D features like edges or corners of the marker may affect camera tracking accuracy depending on the camera position and pose with respect to 3D coordinate system fixed on the marker plane. Especially when tracking the Z axis which is perpendicular to the marker plane (X-Y plane), the angle of the camera relative to the marker plane extremely affects the tracking accuracy.

0-7695-3056-7/07 $25.00 © 2007 IEEE
DOI 10.1109/ICAT.2007.16

If the camera captures the marker from an angle as shown in Fig. 1(a), for example, the 3D coordinate axes fixed on the marker can be correctly tracked over frames. On the other hand, when the image plane of the camera is almost parallel to the marker as shown in Fig. 1(b), directions of axes are not stable even though the camera hardly moves between frames. This is because slight difference of detection of 2D features such as edges or corners between the frames significantly affects computation of 3D position and pose of the camera.

In this paper, we introduce a 2D marker-based tracking method to improve the estimation accuracy of rotation parameters of a camera which represent 3D pose of the camera with respect to a 3D coordinate system fixed on the marker plane. From the image sequence as shown in Fig. 1(b), it is difficult to obtain accurate rotation parameters only by extracting 2D edges of the marker. The best way in terms of accuracy is full search of all parameter candidates, however, it is quite unreasonable for an on-line system from a computational perspective.

Therefore we employ an algorithm of particle filtering [2]. Our method searches the best parameters by comparing the captured image and an appearance generated by each parameter candidate, while a number of parameter candidates are kept based on particle filter. Since we do not estimate the parameters by detecting 2D feature points, but search the best parameters by comparing actual images and generated images. Therefore Z axis can be correctly tracked and estimation errors of the rotation parameters such as jitters between frames can be reduced.

Our method can be applied to the images like Fig. 1(a) as well as the images like Fig. 1(b) with the same algorithm.

2. Related Work

The tracking method using filtering technique is often used for object tracking. Since it is performed by probabilistic estimation based on the previous status, it can be robust against the random noise in the input images. In recent computer vision area, a lot of tracking method such as human position tracking, head pose tracking, face tracking, etc. are studied by using filtering methods. Especially, particle filtering that is one of the filtering methods is highly used for the tracking [7, 4, 5]. Oka et al. use the state vector which is composed of 6 parameters representing 3D human head position and pose. Then they achieved accurate tracking of human head by adaptively controlling diffusion of particles [7].

On the other hand, the particle filtering is also used for camera tracking [8, 6]. Pupilli et al. proposed a camera tracking method using some feature points in the real scene. They applied the particle filter to track the feature points, however, the tracking is based on the template matching without template renewal, so the tracking is not robust against the change in appearance of the feature points. Although the initial feature points are natural feature points, their 3D positions and poses have to be known like using markers.

As a close method to our method, Marimon et al. applied the particle filtering to marker-based tracking [6]. They focus on a problem that the marker is occluded when the whole marker is not completely captured in the image frame. They employ particle filter to combine feature point-based tracking with marker-based tracking only for tracking when the marker can not be completely seen, however their method cannot improve the accuracy when the marker can be seen. Therefore their method is essentially different from our method because the purpose of our method is improving the tracking accuracy of marker-based tracking itself.

3. Camera Tracking Method with Particle Filtering

In this section, we introduce our proposed method that uses particle filtering technique and estimates camera rotation and translation by tracking a 3D coordinate system fixed on a marker plane from input images captured by a moving camera.

The overview of our method is shown in Fig. 2. We use the algorithm of ARToolkit to extract a marker from input images and estimate the translation parameters of the camera. Then the rotation parameters are estimated by using particle filter. The probability density function (PDF) of the particle filter consists of a set of discrete hypotheses (particles) of rotation parameters and corresponding weight values at every frame. For computing the weight values, the particle which can generate the fittest pattern to the actual input image is searched from the candidates. In particular, we project the contour of the marker on the input image by each particle and compute distances from the sampled points on the contour to the nearest edge in the input image. Then the particle which has minimum distance is selected.

In 2D marker-based camera tracking, detection errors of 2D features like edges or corners of the marker may affect

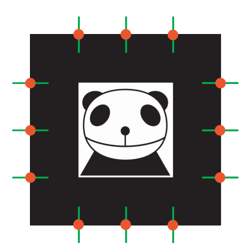


Figure 3. Sampled K points. ($K = 12$)

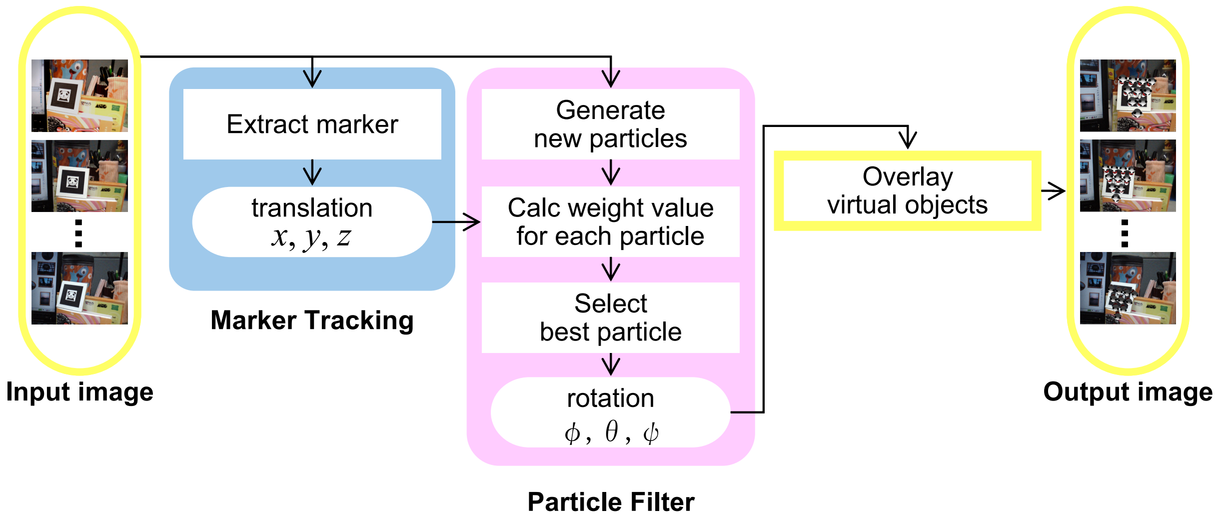


Figure 2. Overview of our tracking method.

camera tracking accuracy depending on the camera position and pose with respect to 3D coordinate system fixed on the marker plane. Since our method evaluates the assumed 3D position and pose of the camera by comparing with the actual input image, the stable tracking can be kept without depending on the camera angle even when the detection errors of 2D features affect tracking accuracy.

3.1. Initialization Step

Our method uses K feature points on a contour of the 2D rectangular marker for computing weight values in the particle filtering. In the experiment described later, we let the number of feature points K=12, where three points are sampled from every side of the contour as shown in Fig. 3(a). Each feature point has a 3D position in a 3D coordinate system fixed on the marker plane (Z=0). We estimate a 3D position and pose of a camera with respect to the 3D coordinate system at every frame.

In the initialization step, 2D corresponding points of K feature points in the initial image frame is computed by rotation and translation parameters which are obtained from ARToolkit. The rotation parameters are also used for generating the initial values of the particles in the tracking step described in next section.

3.2. Tracking Step

In the tracking step, we estimate the camera pose in the t th image frame by applying particle filter to the input images and the initialized K feature points.

In our method translation parameters $(x, y, z)^\top$ of the camera are computed from ARToolkit at every frame. Then the camera pose, that is represented by rotation parameters, is estimated by particle filter at every frame by representing them as a 3D vector $\boldsymbol{p}_t=(\phi_t, \theta_t, \psi_t)^\top$ in a 3D state space $\boldsymbol{S}$ where t represents the frame number. As described before, these rotation and translation parameters represent 3D position and pose of the camera with respect to the 3D coordinate system fixed on the marker plane.

Particle filtering represents the probability density function (PDF) as a set of N discrete hypotheses (particles) $\{\boldsymbol{s}_t^i\}$ in the 3D state space $\boldsymbol{S}$ and the corresponding weight values π_t^i $(i = 1 \cdots N)$. This sample set can approximate an arbitrary PDF.

At the beginning of the tracking, we generate N new samples as $\boldsymbol{s}_t^0$ in neighborhood of the initial values which are rotation and translation parameters obtained at the initialization step. Then a constant value $\pi_0^{(i)} = 1/N$ is given to every particle. In this way the initial values of a set of particles are decided.

After the initial frame, the tracking is performed based on the previous particles and the current input image. The particle set in t th frame $(\boldsymbol{s}_t^i; \pi_t^i)$ is estimated based on the previous assumption set $(\boldsymbol{s}_{t-1}^i; \pi_{t-1}^i)$ and a motion model as following equation.

$$\boldsymbol{s}_t^i = \boldsymbol{s}_{t-1}^i + v_{t-1} + \mu \tag{1}$$

where, v_{t-1} is velocity of the camera and represents the distance from t-2 th frame to t-1 th frame. μ is random noise. In our method, each particle is moved from previous sample based on the concept that the camera moves with uniform motion v_{t-1} and is diffused by adding random noise μ to become the particle in t th frame.

After obtaining N new particles $\{\boldsymbol{s}_t^i\}$, we compute the corresponding weight values π_t^i for $\boldsymbol{s}_t^i$ by evaluation based on the current image frame. The weight values mean the level of confidence for the corresponding particles. Therefore we give larger value to the particle which is closer to the truth. In our method we use the contour of the marker and evaluate how far the contour in the input image from the projected contour by each particle. In particular, we compute distances from the sampled K points on the contour to the edge in the input image. The detail will be described in the next section.

Finally we consider the particle $\boldsymbol{s}_t^i$ which has the maximum weight value as the camera pose $\boldsymbol{p}_t = (\phi_t, \theta_t, \psi_t)^\top$ in t th image.

3.3. Computation of Weight Value for Each Particle

As described in the previous section, the weight values are the level of confidence for the corresponding particles. Therefore we give big value to the particle which seems to be close to the truth; give smaller value to the particle which seems to be far from the truth.

For computing weight value of each particle, K feature points shown in Fig. 3(b) are projected onto the image plane by using the parameters of each particle. Then the distance between each projected point j and the nearest edge in the input image is computed as d_j $(1 \leq j \leq K)$. The nearest edge is searched along the perpendicular line to the contour as shown in Fig. 4. In particular, the searching is started from the projected point to both sides of the line. The sum of the distance d_j for every feature points obtained by the parameters of the particle i is normalized between -1 and 1. The score value is considered as $c_t^{(i)}$.

$$c_t^{(i)} = 1 - \frac{2\sum_{j=1}^{K} d_j}{max} \qquad (-1 \leq c_t^{(i)} \leq 1) \tag{2}$$

where, max is a distance from the projected point to the end of the searching line. Therefore the score value $c_t^{(i)}$ of the particle which has closer rotation parameters to the true parameters is closer to 1, conversely, the score value of the particle which is far away from the truth is close to -1.

After computing the score values $c_t^{(i)}$ for N particles, weight value $\pi_t^{(i)}$ is computed by gaussian function as following equation.

$$\pi_t^{(i)} \propto e^{-\frac{\left(1-c_t^{(i)}\right)^2}{2\sigma^2}} \tag{3}$$

Where, σ is standard deviation of Gaussian function. In our experiment, we let $\sigma = 3.0$. Each $\pi_t^{(i)}$ is normalized so that the sum of all the $\pi_t^{(i)}$ become 1.0. Therefore the particle which has closer to the true will obtain bigger weight value $\pi_t^{(i)}$.

In this way, the particle which has the largest weight value is selected as the rotation parameters in the current frame, after computing the weight values for all the particles. Because of evaluating parameters by checking how fit each particle is to the actual input image, we can always search the best parameters at every frame.

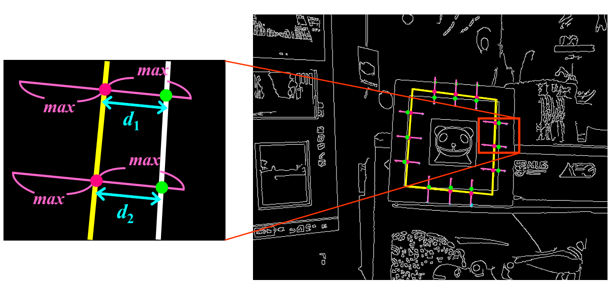


Figure 4. Distance between the projected contour and the actual contour.

4. Experimental Results

We have implemented some experiments to evaluate our estimation method. Our system consists of a PC (OS: Windows XP, CPU: Intel Pentium 4.3 GHz) and a USB camera whose resolution is 640×480 pixel. The size of a 2D marker rectangular marker is 80×80 mm. The number of particles is set to 300. Under this condition the frame rate is 15 fps.

First, the resulting images of computing weight values in particle filter are shown in Fig. 5. Yellow rectangular in the image is the marker's contour projected by the parameters of each particle. The nearest edge is searched along each perpendicular line to the contour. Fig. 5(a) is the result image with maximum weight value. You can see that the projected yellow contour is extremely fit to the actual contour. In contrast, in Fig. 5(h) which is the result image with minimum weight value, since the projected yellow contour is deformed, the contour is not fit to the actual contour at all. From these resulting images, therefore, you can find that our method can correctly assign the weight values according to the difference of appearance between the actual input image and the projected contour from the particles.

Next, we apply our method to the image sequence in which the marker is captured from the moving camera whose view direction is almost perpendicular to the marker plane as shown in Fig. 1(b). The camera is moved as smooth as possible. Fig. 6 shows a tracking result of rotation parameters about X, Y and Z axes. The direction of Z axis is decided by rotations of X and Y axes. Fig. 7 shows resulting images in which cubes are projected on the marker by using the parameters computed from ARToolkit (green cube) and our method (red cube). The resulting images of

(a) i=77, π=0.003951 (b) i=229, π=0.003726 (c) i=298, π=0.003593 (d) i=59, π=0.003454

(e) i=164, π=0.003322 (f) i=100, π=0.003166 (g) i=232, π=0.002897 (h) i=175, π=0.001704

Figure 5. Resulting images of computing weight values for all particles. i**: index of particles** $(0 \leq i \leq 299)$**;** π**: weight value** $(0 \leq \pi \leq 1)$**.**

the cubes while representative three frames are shown in below of Fig. 6. These frames notably indicate the difference of results between our method and ARToolkit.

Comparing the rotation parameters by our method and ARToolkit, each parameter from our method smoothly changes according to the camera motion. In contrast, the rotation parameters of X and Y from ARToolkit are not stable and rapidly changing, for example in the three frames shown in Fig. 6, in which the green cube projected by ARToolkit is significantly inclined compared to the previous and after frame. As described before, the rotation parameters of X and Y axes decide the direction of Z axis. Therefore accurate estimation of the rotations of X and Y axes is very important to improve tracking accuracy when the view direction of the camera is perpendicular to the marker plane. Our method achieves accurate estimation by using particle filtering. The rapid changes in the rotation parameters also cause some jitters between frames. You can also see the difference between our method and ARToolkit in Fig. 7. The (green) cube projected by ARToolkit is unstable with jitters, however the (red) cube projected by our method is stably aligned with the same position and pose. This is due to accurate estimation of the direction of Z axis.

We also apply the same comparison experiment to an-

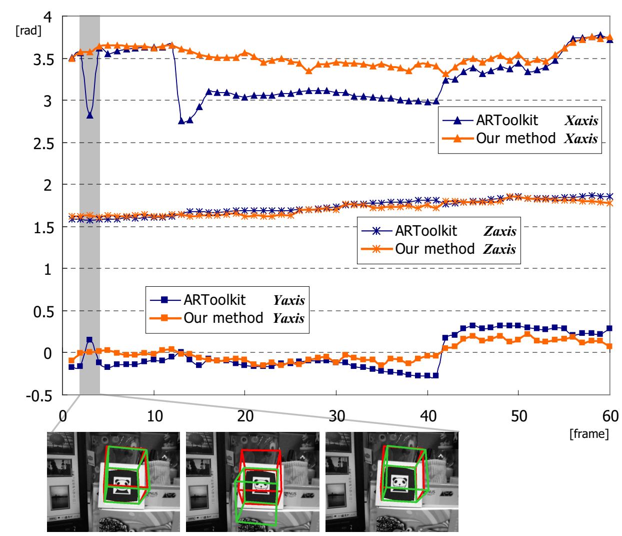


Figure 6. Estimation result of rotation parameters from the images whose view point is perpendicular to the marker plane.

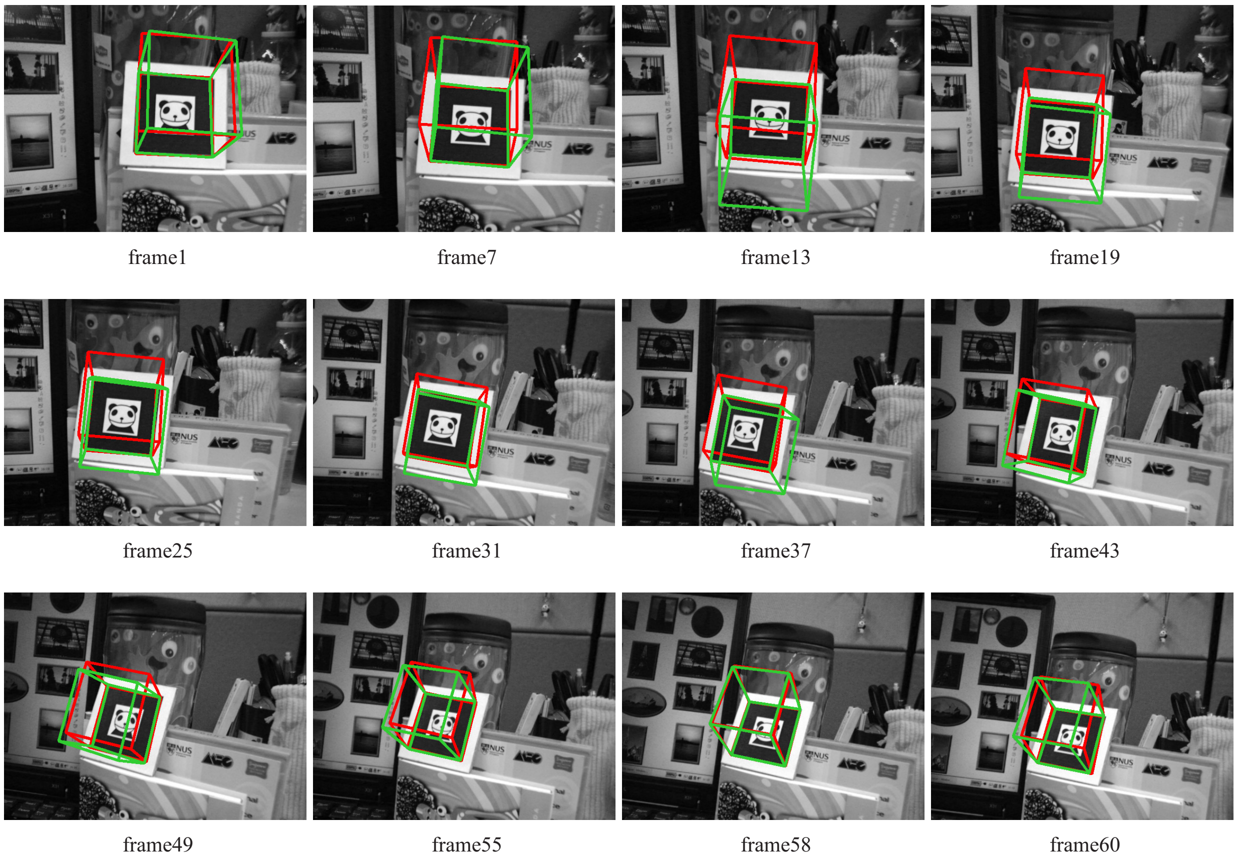


Figure 7. Resulting images of overlaying cubes by ARToolkit and our method. Green cube: by ARToolkit; Red cube: by our method.

other image sequence, in which the marker is captured from a slanted view point as shown in Fig. 1(a). It is known that ARToolkit can correctly estimate each parameter from this kind of image sequence. Here, therefore, we demonstrate that our method can also correctly estimate the rotation parameters without changing the algorithm. Fig. 8 also shows a tracking result of rotation parameters about X, Y and Z axes, and Fig. 9 shows the resulting images of projected cubes: by ARToolkit (green cube) and our method (red cube).

Comparing the results in Fig. 8, both of the results of our method and ARToolkit are quite similar transitions. In the same way, the both cubes in Fig. 9 are projected on almost the same position and pose. Therefore the rotation parameters can be correctly estimated by our method as well as ARToolkit. This result indicates that our method using the particle filtering can also be applied to this kind of image sequence without changing the algorithm. It is effective to apply our method to a lot of marker-based AR approaches.

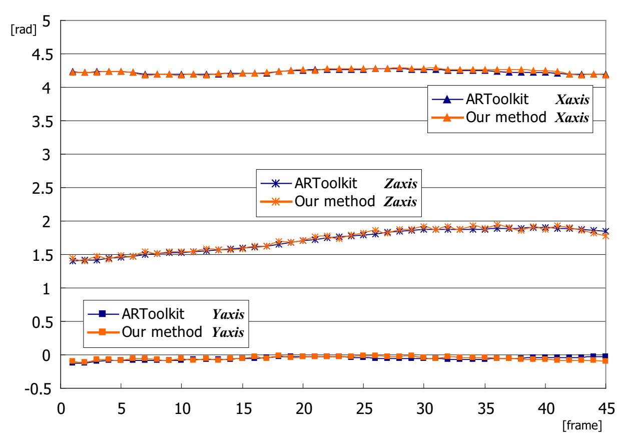


Figure 8. Estimation result of rotation parameters from the images captured from a slanted viewpoint.

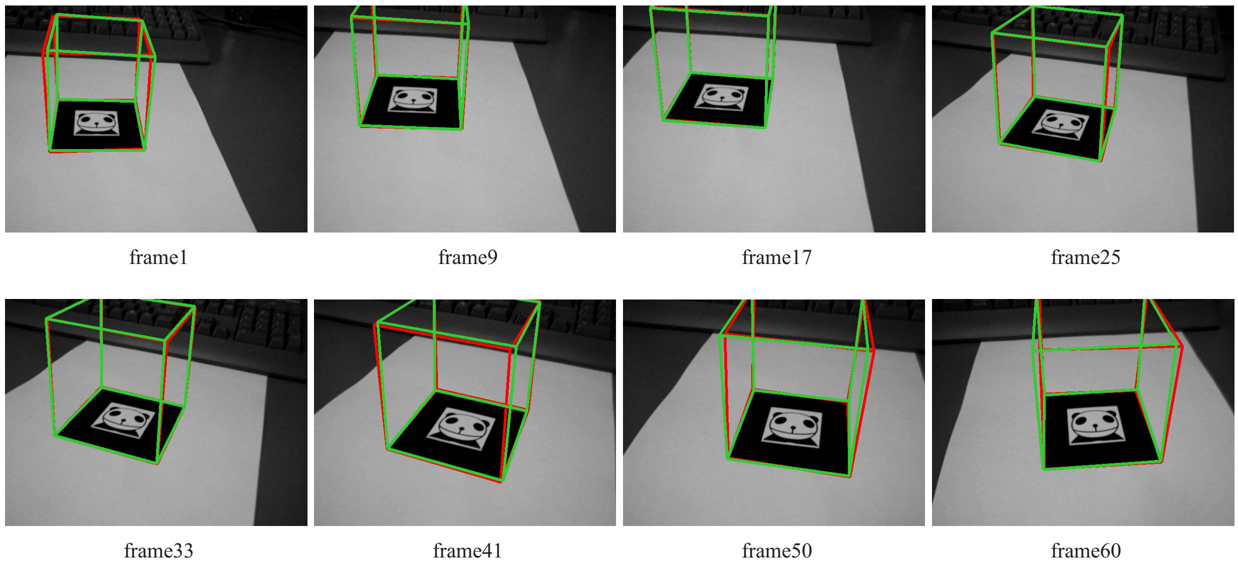


Figure 9. Resulting images of overlaying cubes by ARToolkit and our method. Green cube: by ARToolkit; Red cube: by our method.

5. Conclusion

We have proposed a marker-based camera tracking method which improves the tracking accuracy of 3D camera pose, especially when the camera's image plane is almost parallel to the marker plane. Our method estimates the rotation parameters of the camera by comparing the actual input image and the generated pattern by a lot of hypotheses of the parameters by employing the particle filtering technique. The 3D axes of the coordinate system fixed on the marker plane, especially Z axis, are stably tracked all over the frames, while some jitters are caused by using only ARToolkit. Moreover, our method can be applied not only the images in which marker plane is almost perpendicular to optical axis of a camera, but also the images captured from a inclined view point without changing the algorithm.

6. Acknowledgments

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

References

[1] R. T. Azuma. A survey of augmented reality. *Presence*, pages 355–385, 1997.

[2] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29(1):5–28, 1998.

[3] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proc. of the ISAR*, pages 111–119, 2000.

[4] Y. Kobayashi, D. Sugimura, and Y. Sato. 3d head tracking using the particle filter with cascaded classifiers. In *Proc. of British Machine Vision Conference (BMVC2006)*, pages 37–46, September 2006.

[5] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *Proc. of Computer Vision and Pattern Recognition (CVPR2007)*, June 2007.

[6] D. Marimon, Y. Maret, Yousri, Abdeljaoued, and T. Ebrahimi. Particle filter-based camera tracker fusing marker and feature point cues. In *Proc. of IS&T/SPIE Conf. on Visual Communications and Image Processing*, 2007.

[7] K. Oka, Y. Sato, Y. Nakanishi, and H. Koike. Head pose estimation system based on particle filtering with adaptive diffusion control. In *Proc. of IAPR Conf. Machine Vision Applications (MVA 2005)*, pages 586–589, May 2005.

[8] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proc. of British Machine Vision Conference (BMVC)*, pages 519–528, September 2005.