STABLE POSE ESTIMATION WITH A MOTION MODEL IN REAL-TIME APPLICATION

Po-Chen Wu[1], *Jui-Hsin Lai*[2], *Ja-Ling Wu*[2], *Shao-Yi Chien*[1]

[1]Media IC and System Lab
Graduate Institute of Electronics Engineering and Dept. of Electrical Engineering
[2]Communications and Multimedia Lab
Graduate Institute of Networking and Multimedia and Dept. of Computer Science and Information Eng.
National Taiwan University
pcwu@media.ee.ntu.edu.tw, {larrylai, wjl}@cmlab.csie.ntu.edu.tw, sychien@cc.ee.ntu.edu.tw

ABSTRACT

Estimation of a object pose from camera is a well-developing topic in computer vision. In theory, the pose from a calibrated camera can be uniquely determined. But in practice, most of the real-time pose estimation algorithms suffer from pose ambiguity due to low accuracy of the target object. We think that pose ambiguity—two distinct local minima of the according error function—exist because of the phenomenon of geometric illusions. Both of the ambiguous poses are plausible. After obtaining the solution of two minima (pose candidates), we develop a real-time algorithm for stable pose estimation of a target objects with a motion model. In the experimental results, the proposed algorithm diminish the significance of pose jumping and pose jittering effectively. To the best of our knowledge, this is the first work to solve the pose ambiguity problem with motion model in real-time application.

Index Terms— Pose estimation, pose ambiguity, pose stabilization.

1. INTRODUCTION

The target of pose estimation is to calculate position and orientation of target object from a calibrated camera. Augmented reality (AR) [1], which synthetic objects are inserted into a real scene in real-time, is a prime candidate system for this topic. After obtaining the pose computed with some geometric information, the system could render computer generated images (CGI) according to the pose on the display. ARToolkit [2], for example, is such a system for AR application and have been widely used. The target object in AR system is usually the planar fiducial marker , which used for navigation and localization frequently.

The information available for solving the pose estimation problem is usually a set of point correspondences. They are composed of a 3D reference point expressed in object coordinates and its 2D projection expressed in image coordinates.

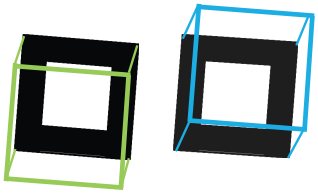

Fig. 1. Illustration of pose ambiguity. It is a geometric illusion: There seems to be more than one 3D geometrical explanations obtained from the same perspective projected marker on the image plane.

Using object space collinearity error, Lu et al. [3] derived an iterative algorithm which directly computes orthogonal rotation matrices. Instead of using iterative algorithm, Ansar et al. [4] developed a framework which allows for a set of linear solutions to the pose estimation problem, and it's for both points and lines. These online pose estimation works calculated the unique pose for each frame without considering the pose ambiguity problem.

Pose ambiguity, as shown in Fig. 1, is the main cause of pose jumping. The derived pose would be random one of the ambiguous poses frame by frame, and it causes pose jumping. From our experiences, several state-of-the-art pose algorithms suffer from pose jumping. These pose ambiguity problems have been discussed by previous works [5], [6]. Oberkampf et al. [5] give a straightforward interpretation for the case of orthographic projection. They develop their algorithm for planar targets, which uses scaled orthographic projection at each iteration step. Schweighofer et al. [6] extended to tackle the general case of perspective projection and develop algorithm for a unique solution to pose estimation. But even with these algorithms, the problem of pose jumping still appears occasionally.

In order to reduce the significance of pose jumping, we propose an algorithm to derive the pose of target object with motion model. The motion model will update through

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.176

Kalman filter [7]. The Kalman filter provides an efficient computational means to estimate the true poses with computing a weighted average of the measured pose and the predicted pose from motion model. From our observation, one of the two ambiguous poses with distinct local minima of error function are the correct one. So every time obtaining the two ambiguous poses, we would choose the pose which is more similar to the predicted pose. If the predicted pose is realistic, then we can almost ensure that the chosen pose is the proper one.

The main contributions of this work are shown as follows,

1. We can solve the problem of pose jumping effectively since we consider the proper pose from two ambiguous poses with motion model.

2. The significance of pose jittering will be reduced because of using Kalman filter. We can estimated the pose that tend to be closer to the true pose than the measured pose. The sequences of estimated poses would be also much smoother because the poses are much more consistent with their previous ones.

3. This is the first work of pose estimation combined with motion model. Even if the target object is miss-detected in some frames of long sequence, we can just use the predicted pose with motion model as the final pose to prevent from discontinues sequence of poses.

The remainder of this article is organized as follows. First, we describe the formulation of the pose estimation problem more formally in Sec. 2. Then we interpret the pose ambibuity and show how to develop the two poses with local minima of the according error function in Sec. 3. In Sec. 4, we describe the details of our stable pose estimation algorithm. In Sec. 5, we show the results of pose estimation and compare the performance with other competitive pose algorithms, and conclusions are drawn in Sec. 6.

2. PROBLEM FORMULATION

The main problem of camera pose estimation is to find out the six degrees of freedom, which are parameterized by the orientation and the position of the target object with respect to a calibrated camera (with known interior parameters), as shown in Fig. 2. Given a set of noncollinear 3D coordinates of reference points $\mathbf{p}_i = (x_i, y_i, z_i)^t, i = i, ..., n, n \geq 3$ expressed in an object-space coordinates and a set of camera-space coordinates $\mathbf{q}_i = (x_i', y_i', z_i')^t$, the transformation between them can be formulated as:

$$\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}, \tag{1}$$

where

$$R = \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{pmatrix} \in SO(3) \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \in R(3) \tag{2}$$

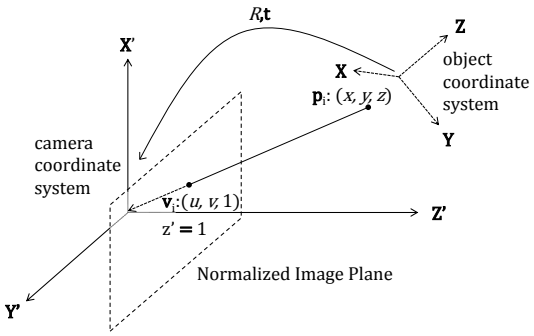

Fig. 2. The coordinate systems between camera and target objects in the pose estimation problem.

are a rotation matrix and a translation vector, respectively.

We introduce the *normalized image plane located* at $z' = 1$ as the camera reference frame. In such a normalized image plane, we define the image point $\mathbf{v}_i = (u_i, v_i, 1)^t$ to be the projection of $\mathbf{p}_i$ on it. Under the idealized pinhole camera model, $\mathbf{v}_i$, $\mathbf{q}_i$ and the center of projection are collinear. We can express this relationship by the following equation:

$$u_i = \frac{\mathbf{r}_1^t \mathbf{p}_i + t_x}{\mathbf{r}_3^t \mathbf{p}_i + t_z}, \quad v_i = \frac{\mathbf{r}_2^t \mathbf{p}_i + t_y}{\mathbf{r}_3^t \mathbf{p}_i + t_z}. \tag{3}$$

Given the observed image points $\hat{\mathbf{v}}_i = (\hat{u}_i, \hat{v}_i, 1)^t$, the pose estimation algorithm has to find values for R and $\mathbf{t}$ that minimize an according error function. In principle, there are two choices, *image-space error*, as used by [5],

$$E_{is}(R, \mathbf{t}) = \sum_{i=1}^{n} \left[(\hat{u}_i - \tfrac{\mathbf{r}_1^t \mathbf{p}_i + t_x}{\mathbf{r}_3^t \mathbf{p}_i + t_z})^2 + (\hat{v}_i - \tfrac{\mathbf{r}_2^t \mathbf{p}_i + t_y}{\mathbf{r}_3^t \mathbf{p}_i + t_z})^2 \right] \tag{4}$$

and *object-space error*, as used by [3], [6],

$$E_{os}(R, \mathbf{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(R\mathbf{p}_i + \mathbf{t}) \right\|^2, \quad \hat{V}_i = \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^t}{\hat{\mathbf{v}}_i^t \hat{\mathbf{v}}_i}. \tag{5}$$

The E_{is} is more heuristic, but the E_{os} is easier to parameterize, and we derive results for E_{os} in the remainder of this paper.

3. POSE AMBIGUITY INTERPRETATION

Pose ambiguity denotes situations where the error function have several local minima for a given configuration. The cause of pose ambiguity is the low accuracy of the reference points extraction, and it's almost inevitable in general cases. Fig. 1. shows the illustration of pose ambiguity.

Most of recent pose estimation algorithms working in real-time suffer from pose ambiguity. Schweighofer et al. [6] found that in the case which coplanar points $\mathbf{p}_i = (p_{i_x}, p_{i_y}, 0)$

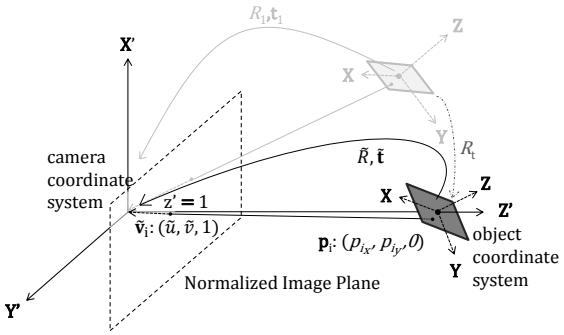

Fig. 3. The transformed coordinate system.

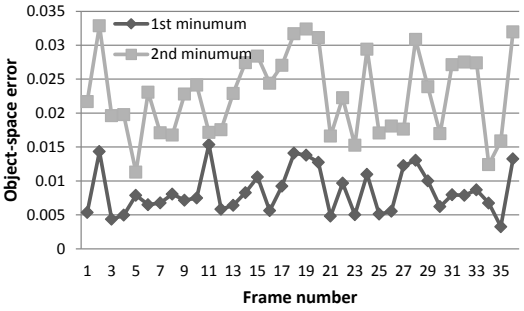

Fig. 4. The object-space errors E_{os} from a sequence video with a planar target. The pose which has the lowest error E_{os} (the dark plot) is the final pose in each frame [6].

viewed by a perspective camera, it typically delivers two distinct minima according to E_{is} and E_{os}. And we derive the two poses with minima of E_{os} by method mentioned in [6].

3.1. Derivation of Poses With Local Minima

Begin with a known pose $(R_1, \mathbf{t}_1)$ got from any pose estimation algorithm, which the iterative algorithm proposed by [3] had been used in our experiences. Then use this first guess of pose to estimate a second pose, which also has a minimum of E_{os}.

Assume reference points $\mathbf{p}_i$. which are measured in the image as $\hat{\mathbf{v}}_i$ such that

$$\hat{\mathbf{v}}_i \approx \mathbf{v}_i \propto R_1\mathbf{p}_i + \mathbf{t}_1, \tag{6}$$

Multiply both sides of (6) with R_t to get a transformed system such that $R_t\mathbf{t}_1 = [0\ 0\ \|\mathbf{t}_1\|]^t$ (see Fig. 3). Let

$$\tilde{\mathbf{v}}_i = R_t\hat{\mathbf{v}}_i,\ \ \tilde{R} = R_tR_1,\ \ \tilde{\mathbf{t}} = R_t\mathbf{t}_1, \tag{7}$$

and the pose $(\tilde{R}, \tilde{\mathbf{t}})$ minimizes

$$E_{os}(\tilde{R}, \tilde{\mathbf{t}}) = \sum_{i=1}^{n} \left\|(I - \tilde{V}_i)(\tilde{R}\mathbf{p}_i + \tilde{\mathbf{t}})\right\|^2. \tag{8}$$

Here we introduce a rotation matrix $\tilde{R}_z$ to let (8) be

$$E_{os}(\tilde{R}, \tilde{\mathbf{t}}) = \sum_{i=1}^{n} \left\|(I - \tilde{V}_i)(\tilde{R}\underbrace{\tilde{R}_z\tilde{R}_z^{-1}}_{I}\mathbf{p}_i + \tilde{\mathbf{t}})\right\|^2, \tag{9}$$

where rotation matrix $\tilde{R}_z^{-1}$ rotates the planar model $\mathbf{p}_i$ only about its z-axis. The rotation matrix $\tilde{R}\tilde{R}_z$ can be decomposed into the product of three rotation matrices $\tilde{R}\tilde{R}_z = R_z(\tilde{\gamma}_1)R_y(\tilde{\beta}_1)R_x(\tilde{\alpha}_1)$, where $R_i(\phi)$ describes a rotation of ϕ degrees about axis i. By selecting $\tilde{R}_z$ such that $\tilde{\alpha}_1 = 0$, we obtain another transformed system

$$\tilde{\mathbf{v}}_i \approx R_z(\tilde{\gamma})R_y(\tilde{\beta})\tilde{\mathbf{p}}_i + \tilde{\mathbf{t}} \tag{10}$$

with the corresponding error function

$$E_{os}(\tilde{\beta}, \tilde{\gamma}, \tilde{\mathbf{t}}) = \sum_{i=1}^{n} \left\|(I - \tilde{V}_i)(R_z(\tilde{\gamma})R_y(\tilde{\beta})\tilde{\mathbf{p}}_i + \tilde{\mathbf{t}})\right\|^2. \tag{11}$$

Since $\tilde{\mathbf{t}} = [0\ 0\ \|\mathbf{t}_1\|]^t$ known from (7), we can rewrite (10) as

$$\tilde{\mathbf{v}}_i \approx R_z(\tilde{\gamma})(R_y(\tilde{\beta})\tilde{\mathbf{p}}_i + \tilde{\mathbf{t}}) \tag{12}$$

because $R_z(\tilde{\gamma})\tilde{\mathbf{t}} = \tilde{\mathbf{t}}_1$. Thus, $R_z(\tilde{\gamma})$ is a rotation just around the optical axis (z-axis) of the camera. This rotation leaves the geometric relation between image plane and model plane invariant and just affects image coordinates. Thus, we can just fix $\tilde{\gamma} = \tilde{\gamma}_1$ and search for local minima of E_{os} with respect to $\tilde{\beta}$ [3], [6].

4. STABLE POSE ESTIMATION ALGORITHM

After obtaining the poses with local minima, some previous work decided the final pose which has the lowest error E_{os} [6], as shown in Fig. 4. Unfortunately, it still suffers from pose ambiguity even when choosing the optimal solution for E_{os}. In fact, the correct pose $\hat{P}$ doesn't consist with the pose with lowest error. From our experimental evidence, we deemed the second pose would be the correct one when pose jumping occurs. The result of Fig. 5 consists with our assumption: The two poses with local minima exchange sometimes and one of the them is correct. Based on this observations, we develop our *Stable Pose Estimation Algorithm*. In each time step, the system would generate a predicted pose $\tilde{P}$ according to a motion model. This motion model simulates the orientation of the pose in real condition and updates through Kalman filter in each time step. We choose the pose which is more similar to $\tilde{P}$ from two candidates as the correct pose $\hat{P}$. The final pose is the weighted average of the predicted pose $\tilde{P}$ and the measured pose $\hat{P}$.

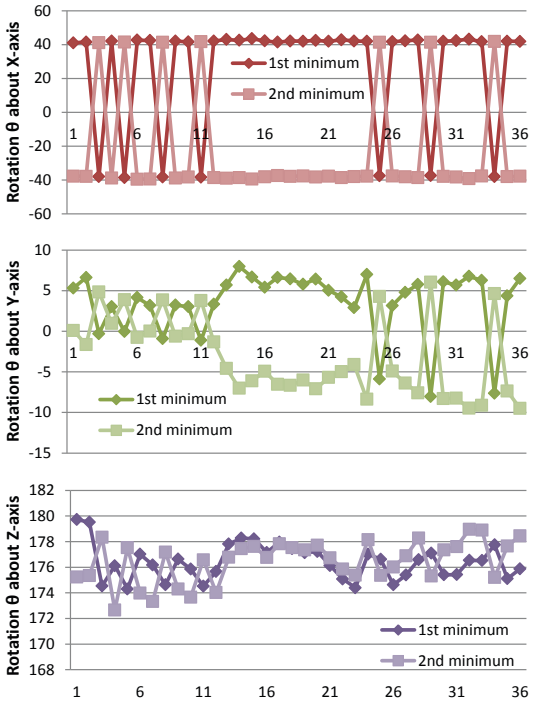

Fig. 5. The rotation angle about X-axis, Y-axis, and Z-axis of the poses with minimum error E_{os}. The value will dramatically changed during some frames.

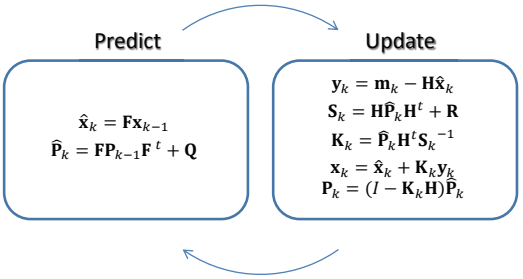

Fig. 6. The detail operations in two phases, "Predict" and "Update" of Kalman filter.

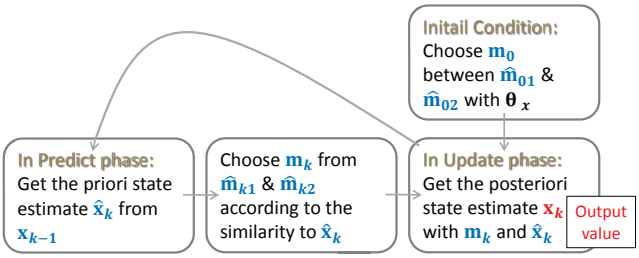

Fig. 7. The system flow of stable pose estimation algorithm.

4.1. Motion Model

Assume the motion model of pose rotation about three axes X, Y, and Z are identical, so here we just discuss the case of rotation about X-axis in the remainder of this paper. The cases of rotation about Y-axis and Z-axis are all the same.

To estimate the following rotation angle with a motion model, the motion model should maintain the current angle value and angular velocity. The angle value and angular velocity re described by the linear state space $\mathbf{x}_k = [x\ \dot{x}]^t$, where $\dot{x}$ is the angular velocity. Assume that between the $(k-1)$ and k time step the system undergoes a constant angular acceleration of a_k, which is normally distributed with mean 0 and deviation σ_a, through Δt seconds. From Newton's laws of motion we conclude that

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k, \tag{13}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}. \tag{14}$$

We rewrite (13) into another form:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k, \tag{15}$$

where

$$\mathbf{w}_k \sim N(0, \mathbf{Q}) \quad \text{and} \quad \mathbf{Q} = \mathbf{G}\mathbf{G}^t\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_a^2. \tag{16}$$

For each time step, we obtain measurements of the rotation angle about X-axis. Let's suppose the measurement noise $\mathbf{v}_k$ is also normally distributed with mean 0 and standard deviation σ_z:

$$\mathbf{m}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \tag{17}$$

where

$$\mathbf{H} = [1\ 0] \quad \text{and} \quad \mathbf{v}_k \sim N(0, \mathbf{R}),\ \mathbf{R} = \mathbf{v}_k\mathbf{v}_k^t = \sigma_z^2. \tag{18}$$

4.2. Predict and Update through Kalman Filter

The operations in two phases of Kalman filter, "Predict" and "Update" are shown in Fig.(6). Because of the pose ambiguity, we'll obtain two measurements, $\hat{\mathbf{m}}_{k1}$ and $\hat{\mathbf{m}}_{k2}$ of the pose in real condition at each time step. Assume that the priori state estimate $\hat{\mathbf{x}}_k$ is very authentic, then we'll check which measurements is more consistent with $\hat{\mathbf{x}}_k$ and thus regard it as the only measurement $\mathbf{m}_k$. After operations of Kalman filter, we can get a new posteriori state estimate $\mathbf{x}_k$, which can be used in the next recursion.

To guarantee that the state estimate is reliable at each time step, we have to make sure the state estimate is authentic

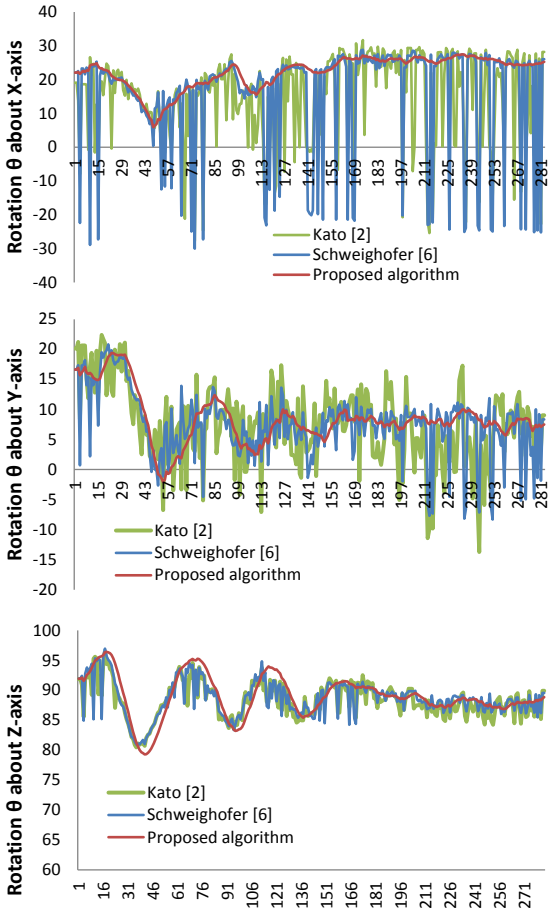

Fig. 9. Comparison of the rotation angle about X-axis, Y-axis, and Z-axis of the poses.

in the beginning. From our experiences, the planar target almost faces upwards in any initial condition, which means the rotation angle of X-axis θ_x is larger than 0, as shown in Fig. 5. Based on this assumption, we choose the first measurement $\mathbf{m}_0$ with larger θ_x than the other. Fig. 7 shows the processing flow of the proposed stable pose estimation algorithm. Finally, we use first element of $\mathbf{x}_k$ to be the output value of pose estimation instead.

5. EXPERIMENTAL RESULTS

In this section, we will discuss about the setting of parameters and show the results of pose estimation. Some video sequences of markers with random rotation angles from the camera are used as the test data. According to the marker pattern in the database provided by [8], we could find the set of point correspondences between object space and image plane as $\mathbf{p}_i$ and $\hat{\mathbf{v}}_i$ in Sec.2. Then we calculated the pose of the planar marker from camera with the set of point correspondences by proposed algorithm and other state-of-the-art algorithms.

5.1. Parameter Settings

The state estimate and estimate covariance matrix, $\mathbf{x}_0$ and $\mathbf{P}_0$, were initialized as following:

$$\mathbf{x}_0 = \begin{bmatrix} \mathbf{m}_0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_0 \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}, \tag{19}$$

where L is a value determined by the variance of the initial state. Larger L means that the initial state estimate is very unreliable and the true value tends to be closer to the measurements. Here we set $L = 10$ in our initial condition.

The other parameters of the motion model in Sec.4.1 to be determined are the deviation of the motion acceleration σ_a, and deviation of the measurement noise σ_z. Larger σ_a means the model has a dramatic acceleration motion, and small σ_z means the measurements are much trustworthy. These two parameters are chosen empirically, where $\sigma_a^2 = 1000$, $\sigma_z^2 = 2$. The experimental results below are generated by these parameters.

5.2. Pose Estimation Result Comparison

We recorded video sequences of a marker which has random rotation angle from the camera. Fig. 8 shows the pose estimation results which are compared with state-of-the-art algorithms. In every condition and every time step, our algorithm provides a solution for real-time pose estimation with high stability. The first row in Fig. 8 is the continues raw image sequences with marker. The second and third rows are resulted by other algorithms, and the forth row are results by our proposed algorithm. Even with low resolution and noisy images, we can still derive a pose sequences without pose jumping, and it is such a difference from others.

Fig. 9 shows the rotation angle of the marker from camera. When the pose jumps during the video sequences, the value of rotation angle would vary dramatically. The most obvious example is the first chart in Fig. 9. With the proposed algorithm, we can almost avoid the situation of pose jumping.

Furthermore, the pose would be much more stable with maintaining a motion model. People would feel more comfortable if the differential values of rotation angles between two continuous frames about each axis are as small as possible. And pose jittering means that the differential values during video sequences are unsettled. Fig. 9 depicts the pose sequences derived by by our algorithm are much more stable with smaller difference between frames. We have also applied some temporal filters to the other two methods trying to diminish the effects of pose jittering, but the final pose would be badly affected by the ambiguous poses nearby.

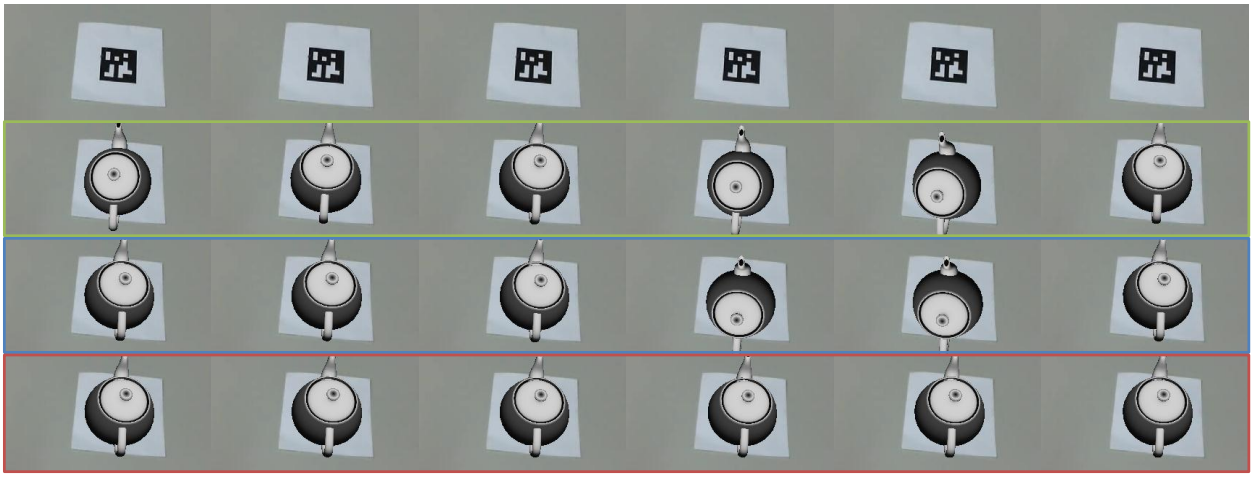

Fig. 8. Pose estimation result comparison. The first raw is the original continues video sequence with a fiducial marker. The second and third raws are the pose estimation results with a CGI of Kato et al. [2] and Schweighofer et al. [6]. The last raw is the results of our algorithm.

6. CONCLUSION

In this work, we propose a stable pose estimation algorithm for real-time application. The proposed concept of motion model can not only be used with proposed algorithm, but other pose estimation algorithms. By this way, the significance of pose jittering can be diminished dramatically. We can even predict the correct pose from two candidate ambiguous poses with the motion model, so the problem of pose jumping can be solved effectively.

To the best of our knowledge, this is the first work which combines pose estimation algorithm with motion model. Because lots of the applications of pose estimation are processed in video form, so we cannot derive the pose with considering information just from one frame. With the implementation of the Kalman filter, the derived pose in each time step would be more consistent with the previous ones. And users of these applications will feel more comfortable with the much smoother pose sequences.

Acknowledgment

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC99-2911-I-002-201, 99R70600, and 10R80800.

7. REFERENCES

[1] Ronald Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 7, pp. 355–385, 1997.

[2] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings of 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, 1999, pp. 85–94.

[3] C.-P. Lu, G.D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, Jun. 2000.

[4] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578 – 589, May 2003.

[5] Denis Oberkampf, Daniel F. DeMenthon, and Larry S. Davis, "Iterative pose estimation using coplanar feature points," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 495 – 511, 1996.

[6] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2024–2030, Dec. 2006.

[7] Greg Welch and Gary Bishop, "An introduction to the kalman filter," 1995, Technical Report TR 95-041, University of North Carolina, Department of Computer Science.

[8] Schmalstieg D. Wagner, D., "Artoolkitplus for pose tracking on mobile devices," in *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, 2007, pp. 139–146.