Real-Time Bayesian 3-D Pose Tracking

Qiang Wang, Weiwei Zhang, Xiaoou Tang, *Senior Member, IEEE*, and Heung-Yeung Shum, *Fellow, IEEE*

***Abstract*—In this paper, we propose a novel approach for real-time 3-D tracking of object pose from a single camera. We formulate the 3-D pose tracking task in a Bayesian framework which fuses feature correspondence information from both previous frame and some selected key-frames into the posterior distribution of pose. We also developed an inter-frame motion inference algorithm which can get reliable inter-frame feature correspondences and relative pose. Finally, the maximum *a posteriori* estimation of pose is obtained via stochastic sampling to achieve stable and drift-free tracking. Experiments show significant improvement of our algorithm over existing algorithms especially in the cases of tracking agile motion, severe occlusion, drastic illumination change, and large object scale change.**

***Index Terms*—Bayesian fusion, real-time vision, 3-D pose tracking.**

I. Introduction

REAL-TIME 3-D object pose tracking is often required in many computer vision applications such as human computer interaction (HCI) and augmented reality (AR). The problem of estimating a rigid pose transformation relating one 2-D image to known 3-D geometry has been intensively studied. Closed form solutions [1] three or four 2-D–3-D point correspondences to estimate the pose. Since the solution is based on the root of high degree polynomial equations and does not use the redundancy in the data, the estimation result is susceptible to noise. The nonlinear optimization based methods [2] apply Gauss–Newton or Levenberg–Marquardt optimization to the pose estimation problem. This approach relies on a good initial guess to converge to the correct solution and is generally slow to converge. The iterative linear method has been developed by employing the specific geometric structure of the pose estimation problem during optimization [3], [4]. Such methods require little computational cost, which is appealing for real-time processing. All the above methods are based on point correspondence which is critical for pose tracking.

For the temporal pose tracking problem, existing methods can be divided into two groups. The first kind of method estimates the incremental pose changes between neighboring frames by registering the model with the image directly, which either presupposes that there are known model features whose image projection can be determined [5] or there is a template image with known pose and the registration between the template and the current image can be carried out [6], [7]. The main problem for this kind of method is that fixed model features can be unstable in cases of occlusion, non-rigid deformation and illumination change, thus the registration between the template and the current image is difficult. The second kind of method is differential tracking which estimates the incremental pose change via incremental motion estimation between neighboring frames. It can make use of arbitrary features on the model surface and does not need to model complex global appearance changes. But the main problem for this kind of method is that the differential technique suffers from accumulated drift which limits their effectiveness when dealing with long sequences. So key-frames are used to reduce the motion drift [8]. More recently, Vacchetti *et al.* [9] proposed an algorithm which fuses the online and offline key-frame information to achieve stable real-time tracking which achieves state-of-the-art performance. However there are still some limitations. First, in the case of agile motion, feature point matching between neighboring frames becomes unreliable and can thus cause the tracker to fail. Second, when the key-frames are also obtained on-line, they can also have drift that can possibly propagate. Third, the fusion of the prior online information with that of one key-frame is performed in a heuristic manner and cannot guarantee optimal performance in the cases of image and model uncertainties. For example, such a fusion technique becomes unstable when there is a considerable difference between the object's real 3-D model and the 3-D model used for tracking.

The above analysis motivates our work. We formulate the key-frame based differential pose tracking problem in a general Bayesian tracking framework. Our contribution lies in two aspects. 1) Based on the Bayesian tracking framework, we obtain a pose posterior distribution which fuses feature correspondence information from both the previous frame and the key-frame, where each feature correspondence pair contributes independently to the posterior distribution and small errors in each feature correspondence (due to image or model uncertainty) can possibly reduce each other. Furthermore our new posterior distribution can integrate multiple key-frames simultaneously, which is difficult for the fusion technique described in [9]; we can efficiently get the maximum *a posteriori* (MAP) estimation of pose from the proposed posterior distribution. 2) We also develop an inter-frame motion inference algorithm which can get reliable feature correspondences and relative pose estimation in difficult cases such as agile motion and severe occlusion, and the algorithm can get reasonable results even when only 10% of initial feature correspondences are good. Extensive experiments clearly show the advantage of our algorithm over another current state-of-the-art technique [9].

II. Bayesian Differential Pose Tracking With Key-Frames

In a dynamical system, the state of the object and image observation at time t are represented as X_t and I_t, respectively. The sequence of states and observations up to time t are denoted by $\mathbf{X}_t = \{X_1, \ldots, X_t\}$ and $\mathbf{I}_t = \{I_1, \ldots, I_t\}$, and the tracking problem at time t can be regarded as an inference problem of the

Manuscript received March 1, 2006; revised August 21, 2006. This paper was recommended by Associate Editor P. Frossard.

The authors are with Microsoft Research Asia, Beijing, 100080 China (e-mail: qiangwa@microsoft.com).

Color versions of Figs. 3-7 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2006.885727

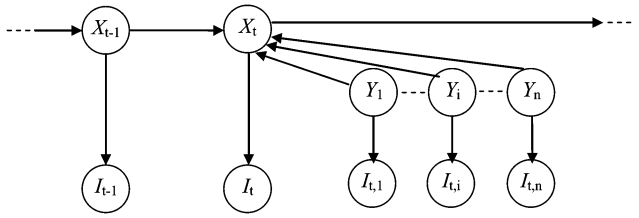

Fig. 1. Graphical model at time t.

posterior distribution $P(X_t|\mathbf{I}_t)$. In the differential pose tracking problem, the incremental pose between the neighboring images is measured by the relative motion between them. The advantage of differential pose tracking is that it does not require semantic point correspondences between the object 3-D model points and their 2-D image projections. However, it is well known that the differential techniques suffer from accumulated drift. The solution is to select some key-frames from the past estimated frames and anchor the current frame to them. By fusing the key-frame information and previous neighboring frame information, we can obtain a pose estimate which exhibits much less drifting error. In this paper, we formulate such a key-frame based differential pose tracking problem in a Bayesian framework, and the developed algorithm achieves real-time robust performance.

The Bayesian key-frame based differential pose tracking problem could be represented by a graphical model as shown in Fig. 1. At time t, a set of key-frames $\{Y_1, \ldots, Y_n\}$ is selected, and $\{I_{t,1}, \ldots, I_{t,n}\}$ is its corresponding image observation. For conciseness, we denote the previous neighboring frame as the 0th key-frame and let $Y_0 = X_{t-1}$, $I_{t,0} = I_{t-1}$, with the posterior distribution of X_t specified by

$$P(X_t|\{Y_i\},\{I_{t,i}\},I_t) = \frac{P(I_t,\{I_{t,i}\}|X_t,\{Y_i\})P(X_t|\{Y_i\})P(\{Y_i\})}{P(\{Y_i\},\{I_{t,i}\},I_t)}. \quad (1)$$

In (1), we assume that the pose states of key-frames are deterministic since we select key-frames from those past frames which have high confidence pose estimations. We will justify this assumption in Section V. So (1) can be simplified as

$$P(X_t|\{Y_i\},\{I_{t,i}\},I_t) \propto P(I_t,\{I_{t,i}\}|X_t,\{Y_i\})P(X_t|\{Y_i\}). \quad (2)$$

It is reasonable to assume the conditional independence between key-frames, so (2) becomes

$$P(X_t|\{Y_i\},\{I_{t,i}\},I_t) \propto \frac{\prod_{i=0}^{n} P(I_t,I_{t,i}|X_t,Y_i)P(X_t|\{Y_i\})}{P(I_t|X_t)^n}. \quad (3)$$

$P(I_t|X_t)$ is the texture consistency likelihood which can be evaluated given the known 3-D textured model, however it is so unstable due to illumination change, partial occlusion, image noise, motion blur, etc., that we simply assume that it has a uniform distribution. Let $\delta_i^t = X_t \sim Y_i$ be the pose change between pose state Y_i and X_t, where $\sim$ is the pose difference operator whose form depends on the parameterization of the pose. In this paper, we use the quaternion representation of rotation and the details will be described in Section V. We also define the pose composition operator $\circ$ such that $(X_t \sim Y_i) \circ Y_i = X_t$. Since Y_i is assumed deterministic, (3) can be written as

$$P(X_t|\{Y_i\},\{I_{t,i}\},I_t) \approx c \cdot \prod_{i=0}^{n} P\left(I_t, I_{t,i}|\delta_i^t, Y_i\right) P(X_t|\{Y_i\}) \quad (4)$$

where c is a constant. From (4), we can see that the right side has two terms: one is the product of pose likelihood functions, and the other is the prediction density of X_t given key-frame pose Y_i. We assume that the prediction density has a Gaussian distribution, with its mean determined by the previous frame's pose estimate $\hat{Y}_0$ and corresponding relative pose estimate $\hat{\delta}_0^t$, and its covariance denoted as Σ

$$P(X_t|\{Y_i\}) = N\left(\hat{\delta}_i^t \circ \hat{Y}_0, \Sigma\right). \quad (5)$$

The above model generalizes the pose tracking problem by fusing the information from a previous frame and a key-frame in a principled way. Now the problem is how to model the pose likelihood function and how to get the relative pose estimate $\hat{\delta}_0^t$. We will first introduce the pose likelihood model in Section III, and then introduce the inter-frame motion inference algorithm in Section IV. The whole Bayesian inference algorithm and implementation will be introduced in Section V.

III. Pose Likelihood Model

We define the pose likelihood model $P(I_t, I_{t,i}|\delta_i^t, Y_i)$ based on the point matches between frame I_t and $I_{t,i}$. First, we detect a set of strongest interest points in frame $I_{t,i}$ and denote it as $\mathbf{u}_i = \{u_i^1, \ldots u_i^m\}$; its correspondence points in frame I_t are denoted as $\mathbf{v}_i = \{v_i^1, \ldots v_i^m\}$. Since we know the 3-D model pose Y_i in frame $I_{t,i}$, we can back project $\mathbf{u}_i$ to the 3-D model to get the corresponding 3-D points $\mathbf{U}_i = \{U_i^1, \ldots U_i^m\}$. Given the 2-D–3-D correspondences between the point set $\mathbf{U}_i$ and $\mathbf{v}_i$, the relative pose between frames I_t and $I_{t,i}$ can be calculated [3], [4], thus we define the likelihood of the relative pose δ_i^t as follows:

$$P\left(I_t, I_{t,i}|\delta_i^t, Y_i\right) \approx P\left(\mathbf{U}_i, \mathbf{v}_i|\delta_i^t\right) \propto \exp\left(-\sum_{j=1}^{m} \rho\left(\frac{e_{ij}^2}{2\sigma^2}\right)\right) \quad (6)$$

where $\rho(\cdot)$ is a robust function [10]

$$\rho(r) = \begin{cases} r, & r < T \\ 2T, & r \geq T. \end{cases} \quad (7)$$

T is a threshold, and e_{ij} is the position difference between 2-D point v_i^j and the image projection of 3-D point U_i^j

$$e_{ij}^2 = \|v_i^j - A[R|T]U_i^j\|^2 \quad (8)$$

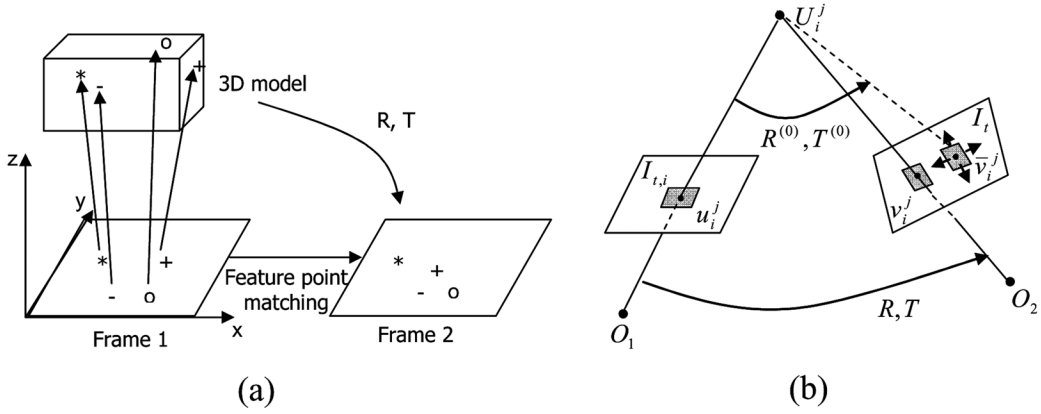

Fig. 2. (a) Relative pose from feature correspondence. (b) Geometrically constrained feature matching.

where A is the camera's internal parameter matrix which can be obtained offline in advance, and R, T is the rotation matrix and translation vector of the relative pose δ_i^t, respectively.

The goodness of the likelihood model defined in (6) depends on the correctness and accuracy of the correspondences between $\mathbf{U}_i$ and $\mathbf{v}_i$. There are two causes of correspondence error: the error of the 2-D feature matches between I_t and $I_{t,i}$, and the deviation between the provided 3-D object model and the real object model. For the first error, we propose to use an iterative inter-frame motion inference algorithm to get refined matching and reject those features which violate the 3-D model constraint or are in the background scene; this will be described in Section IV. For the second error, we can reduce it by Bayesian fusing of multiple key-frames with many point correspondences. Some related synthetic experiments will be described in Section VI.

IV. Inter-Frame Motion Inference

In this section, we will describe the algorithm to calculate good 2-D feature matches between I_t and $I_{t,i}$. We employ the coupling effects of the inter-frame feature correspondence and the relative pose, as shown in Fig. 2. Fig. 2(a) indicates how to get the relative pose from feature correspondences and Fig. 2(b) indicates that given a rough relative pose estimate, we can narrow the baseline of the two feature windows and make the feature matching easier and more reliable. The estimation of inter-frame feature matching and relative pose is actually a chicken-and-egg problem, so we will formulate this problem and jointly estimate the feature correspondence and relative pose iteratively. The key point is that through iterative estimation, we can gradually get better relative pose estimation and thus get more accurate 2-D feature correspondences.

A. Joint Distribution of Feature Correspondence and Relative Pose

We consider the joint distribution of feature correspondence and relative pose between frame $I_{t,i}$ and I_t. It is denoted as

$$P(\mathbf{v}_i, R, T | I_{t,i}, I_t, \mathbf{U}_i) \tag{9}$$

where the meaning of symbols $\mathbf{v}_i, R, T, \mathbf{U}_i$ is as described in Section III. The above joint distribution is difficult to directly handle due to its high dimensionality and nonlinearity, but its two conditional distributions can be effectively modeled.

The first conditional distribution for (9) is $P(R, T | I_{t,i}, I_t, \mathbf{U}_i, \mathbf{v}_i)$, which is the distribution of relative pose given the 2-D–3-D correspondences between matched image features and model points. We ignore its prior distribution and keep the likelihood since we do not have any prediction on the relative pose, so it can be written as

$$P(R, T | I_{t,i}, I_t, \mathbf{U}_i, \mathbf{v}_i) \approx P(\mathbf{U}_i, \mathbf{v}_i | R, T) \propto \exp\left(-\sum_{j=1}^{m} \rho\left(\frac{e_{ij}^2}{2\sigma^2}\right)\right) \tag{10}$$

where the right side of (10) has the same form as that in (6).

The second conditional distribution for (9) is $P(\mathbf{v}_i | I_{t,i}, I_t, \mathbf{U}_i, R, T)$, which is the distribution of feature correspondences $\mathbf{v}_i$ in image I_t given its 3-D model points $\mathbf{U}_i$ and relative pose estimation R, T. It can be modeled as

$$P(\mathbf{v}_i | I_{t,i}, I_t, \mathbf{U}_i, R, T) \propto \exp\left(-\sum_{j=1}^{m} e_{ij}^2 - \lambda \sum_{j=1}^{m} f_j^2\right). \tag{11}$$

The above distribution accounts for both the geometric constraint from the 3-D model and the appearance constraint from image intensity consistency. e_{ij} is the geometric constraint term as defined in (8), and λ is a weight coefficient. In practice, we tune λ to balance the appearance constraint to be about five times larger than the geometric reprojection error constraint. f_j is the appearance constraint term defined as follows:

$$f_j^2 = \sum_k \left\| c_2^{(j)} * I_{t,i}\left(W_j\left(v_i^{(j,k)}\right)\right) - c_1^{(j)} * I_t\left(v_i^{(j,k)}\right)\right\|^2. \tag{12}$$

$W_j(\cdot)$ is a 2-D projective warping which can be directly determined by the relative pose R, T, 3-D points U_i^j and its corresponding mesh normal [11]. $v_i^{(j,k)}$ is the coordinate of the kth pixel in a window centered at v_i^j. This window is used for image

feature matching. $c_1^{(j)}$ and $c_2^{(j)}$ are, respectively, the averaged intensity level of correlation windows in $I_{t,i}$ and I_t for illumination compensation.

Given the above two modeled conditionals, we propose to get the MAP estimation of $\mathbf{v}_i, R, T$ via iterative conditional modes (ICM), which is a flexible inference algorithm. The framework of the algorithm is as follows.

1) Initialize $\mathbf{v}_i$ through generic feature matching, set $l = 1$

2) $(R^{(l)}, T^{(l)}) \leftarrow \arg\max_{R,T} P\left(R, T | I_{t,i}, I_t, \mathbf{U}_i, \mathbf{v}_i^{(l-1)}\right)$

3) $\mathbf{v}_i^{(l)} \leftarrow \arg\max_{\mathbf{v}_i} P\left(\mathbf{v}_i | I_{t,i}, I_t, \mathbf{U}_i, R^{(l)}, T^{(l)}\right)$

4) If not converged, set $l = l + 1$, go to 2.

The ICM iteration starts with an initial guess of feature matching obtained by a multiscale block matching with illumination compensation. We do not require such matching to be good since the iteration step of relative pose estimation can get an approximately good pose even if only 10% of feature matches are good. Given this approximately good pose, we can warp frame $I_{t,i}$ to more closely match frame I_t, then the iteration step of feature matching on the warped image can be improved significantly. The ICM iteration improves feature matching performance especially in the cases of large inter-frame motion which include translation and rotation. In the cases of feature matching between key-frame and current frame, since we already have an inter-frame motion estimation result, we can use a technique as that described in [9] for key-frame image warping: we warp the key-frame image to the position at the estimated current frame position to replace the image $I_{t,i}$ and do multiscale block matching to get feature matches between the key-frame and current frame. The details on how to perform the two ICM estimation steps are described in Section IV-B and C.

B. Relative Pose Optimization

Maximizing the probability in (10) (step 2 of ICM) is equivalent to minimizing a cost function which is simply the negative log of the posterior in (10)

$$C(R,T) = \sum_{j=1}^{m} \rho\left(\frac{e_{ij}^2}{2\sigma^2}\right). \tag{13}$$

We employ a stochastic optimization approach extended from RANSAC [12] to optimize (13). Note the relationship between $\mathbf{v}_i$, $\mathbf{U}_i$, R, and T as specified in (8); we generate a number of samples from the feature pairs set $\mathbf{v}_i, \mathbf{U}_i$, where each sample is generated by randomly selecting a minimum set of feature pairs that can recover the relative pose R, T. The cost function in (13) can thus be evaluated and the $[R|T]$ associated with the sample of minimum cost is the optimization result. The POSIT algorithm [3] can be used to recover the relative pose from 2-D–3-D feature pairs. The minimum number of feature pairs is 4 in order to recover the pose. Now we discuss some issues in relative pose estimation.

Convergence of POSIT: It has been pointed out in [3] that the POSIT algorithm is not guaranteed to be convergent. Although it converges extremely fast in many camera-object configurations, we did observe that it may converge very slowly or even diverge in some real cases. Since the convergence depends on the camera-object configuration, in our algorithm implementation we choose to first transform object coordinates to a reference coordinate system centered at one of the object points. Then we apply the POSIT algorithm to obtain the transformation between the camera coordinate system and the reference coordinate system, and then compensate back the first transformation to get the result. Although we still cannot guarantee the convergence of this modified algorithm, for example in the case of erroneous 2-D–3-D feature correspondences, we did achieve much faster and more stable convergence compared with the original algorithm in all experiments.

Combination of Pose Estimation Algorithms: Although we cannot guarantee the convergence of the modified POSIT algorithm, we want to utilize its extremely fast convergence speed in most cases, so we propose to use a combination of the pose estimation algorithms: we first minimize the cost function in (13) based on the modified POSIT algorithm, then we obtain all the inlier feature pairs given the estimated pose; the final pose is then estimated by applying the orthogonal iteration method [4] on all the inlier feature pairs, which is guaranteed to converge.

Rejection of Bad Feature Matches: The robustness of relative pose estimation depends on the percentage of out-lier feature matches. If we can reject as many bad feature matches as possible, we can increase the possibility of getting a reliable relative pose estimate. We reject the bad feature matches by the following steps. 1) Reject feature matches which are not consistent with the image intensity constraint, such feature matches with large intensity error as specified in (12). 2) Reject feature matches which lie in the background scene; since the background features normally appear at positions near the object boundary, they can easily get matched since the background usually does not move. The acquired background feature matches have very high intensity consistency and they are structured outliers, which can significantly decrease the robustness and accuracy of the pose estimation. We propose to reject background features as follows: first we calculate the average feature motion between $\mathbf{u}_i$ and $\mathbf{v}_i$. If there is enough motion, then for each feature, we calculate the block difference between the position u_i^j in image $I_{t,i}$ and the same position u_i^j in image I_t. Again if such block difference is small, then jth feature is regarded as a background feature and can be rejected. The basic idea of the above process is that if we conclude that the object moves across frames, then the image window intensity at an object feature position should change, otherwise the feature lies in the background.

This stochastic optimization approach is very robust with respect to feature correspondence outliers, and can even recover an approximately good pose when initially 90 of 100 feature pairs are outliers.

C. Geometrically Constrained Feature Matching

Maximizing the probability in (11) (step 3 of ICM) is equivalent to minimizing a cost function which is simply the negative log of the posterior in (11)

$$C(\mathbf{v}_i) = \sum_{j=1}^{m} e_{ij}^2 + \lambda \sum_{j=1}^{m} f_j^2. \tag{14}$$

The above minimization can be viewed as a constrained feature matching process, as shown in Fig. 2(b). With known pose R, T, the first image $I_{t,i}$ can be prewarped to the position of the second image I_t, then block matching starting from $\bar{v}_i^j$ is performed using (12) as the matching cost. Since the multiscale block matching can be done approximately over integer image coordinates, no image interpolation is required and the resulting matching algorithm is extremely efficient especially for real-time processing.

V. Bayesian Estimation of Pose

In this section, we will describe the inference of the current pose for the model specified by (4). In Section IV, we have described a method to obtain a set of good 2-D–3-D point matches between a key-frame and current frame pair, and thus the pose likelihood model can be constructed. In the same time, the relative pose δ_0^t is also estimated and can be used to specify the prediction prior of the current pose. In this section, we will describe how to select key-frames from past frames for the current frame's pose estimation and how to get the MAP estimation of the current pose.

A. Selection of Key-Frames

The selection of key-frames is important since it directly affects the pose likelihood model. Remember the assumption that we made in Section II: the key-frame's pose state is deterministic at current time t and its distribution is a delta function. So we should pick key-frames from previously tracked frames which have sharp posterior distributions specified by (4). However, such an assumption still causes some deviation since even a sharp posterior distribution is not actually a delta function. As a result, if we always choose the most recently acquired key-frames as the key-frames for the current pose estimation, the deviation on key-frame poses will accumulate and finally corrupt the tracking. Fig. 3 shows the pose trajectory during tracking. On one hand we should choose a key-frame which is far away from current frame along the time axis, because such a key-frame's deviation is unlikely to be cumulative and is likely to be independent of the estimated previous frame's deviation due to the long time interval. On the other hand, we should choose a key-frame which has similar pose as the current frame's pose, because when the pose difference is very large, the observed object images have little overlap and feature matching between them is impossible. As an example, in Fig. 3, the best

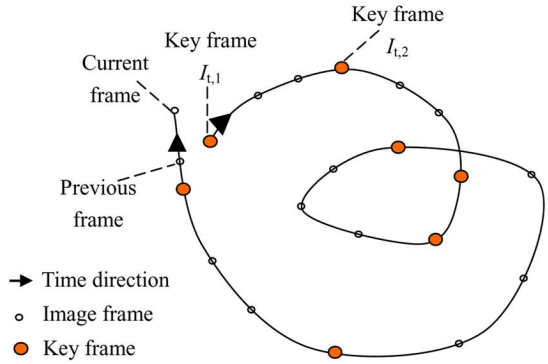

Fig. 3. Key-frame selection during tracking.

two key-frames for the current frame should be $I_{t,1}$, $I_{t,2}$. We define a measure to reflect the above requirements

$$m(Y) = \exp\left(\frac{-d(Y, \hat{X}_t)}{\sigma^2}\right) \exp\left(\frac{-n_0}{\min(n_0, n_x - n_y)}\right) \tag{15}$$

where $d(\cdot,\cdot)$ is a distance function defined in pose space and will be described in Section V-B, $\hat{X}_t$ is the estimated pose of current frame, n_x and n_y are the temporal frame index of the current frame and the key-frame, respectively, and n_0 is a parameter to control the temporal difference between the selected key-frame and the current frame. We first construct a set which contains key-frames with largest measures or key-frames which is within the specified pose distance from current pose, then we select several key-frames from the above set which have maximal temporal distance to the current frame. In our synthetic experiment, the key-frames number varies and in the real experiment, the key-frame number is set to 1 for computational efficiency.

B. MAP Estimation of Current Pose

Before introducing the inference algorithm, we first define the pose difference operator, composition operator, and the distance measure on pose. Suppose that we use a quaternion representation of rotation [13], and denote $X_1 = (\mathbf{q}, \mathbf{t}) = (q_0, q_1, q_2, q_3, t_1, t_2, t_3)$, $X_2 = (\mathbf{r}, \mathbf{s}) = (r_0, r_1, r_2, r_3, s_1, s_2, s_3)$, where $\mathbf{q}$, $\mathbf{r}$ is the quaternion representation of rotation and $\mathbf{t}$, $\mathbf{s}$ is the translation vector. Then we define

$$X_1 \circ X_2 = (\mathbf{q} \wedge \mathbf{r}, R(\mathbf{q}) \cdot \mathbf{s} + \mathbf{t}) \tag{16}$$

$$X_1 \sim X_2 = (\mathbf{q} \wedge \bar{\mathbf{r}}, \mathbf{t} - R(\mathbf{q} \wedge \bar{\mathbf{r}}) \cdot \mathbf{s}) \tag{17}$$

$$d(X_1, X_2) = \frac{\|\gamma\|^2}{\sigma_r^2} + \frac{\|\mathbf{t} - \mathbf{s}\|^2}{\sigma_t^2} \tag{18}$$

where $\wedge$ is the quaternion multiplication operator, $\bar{\mathbf{r}}$ is the conjugate of $\mathbf{r}$, $R(\mathbf{q})$ is the rotation matrix represented by the quaternion $\mathbf{q}$, γ is the vector part of $\mathbf{q} \wedge \bar{\mathbf{r}}$ and σ_r, σ_t are parameters to normalize the dimension size of rotation and translation, respectively.

Then based on (4)–(6), we use stochastic sampling style optimization [14] to get the MAP estimate of X_t in (4). First we generate samples of X_t from distribution $P(X_t|\{Y_i\})$ which is a Gaussian centered at $(\hat{\delta}_0^t \circ Y_0)$, then (4) can be evaluated and

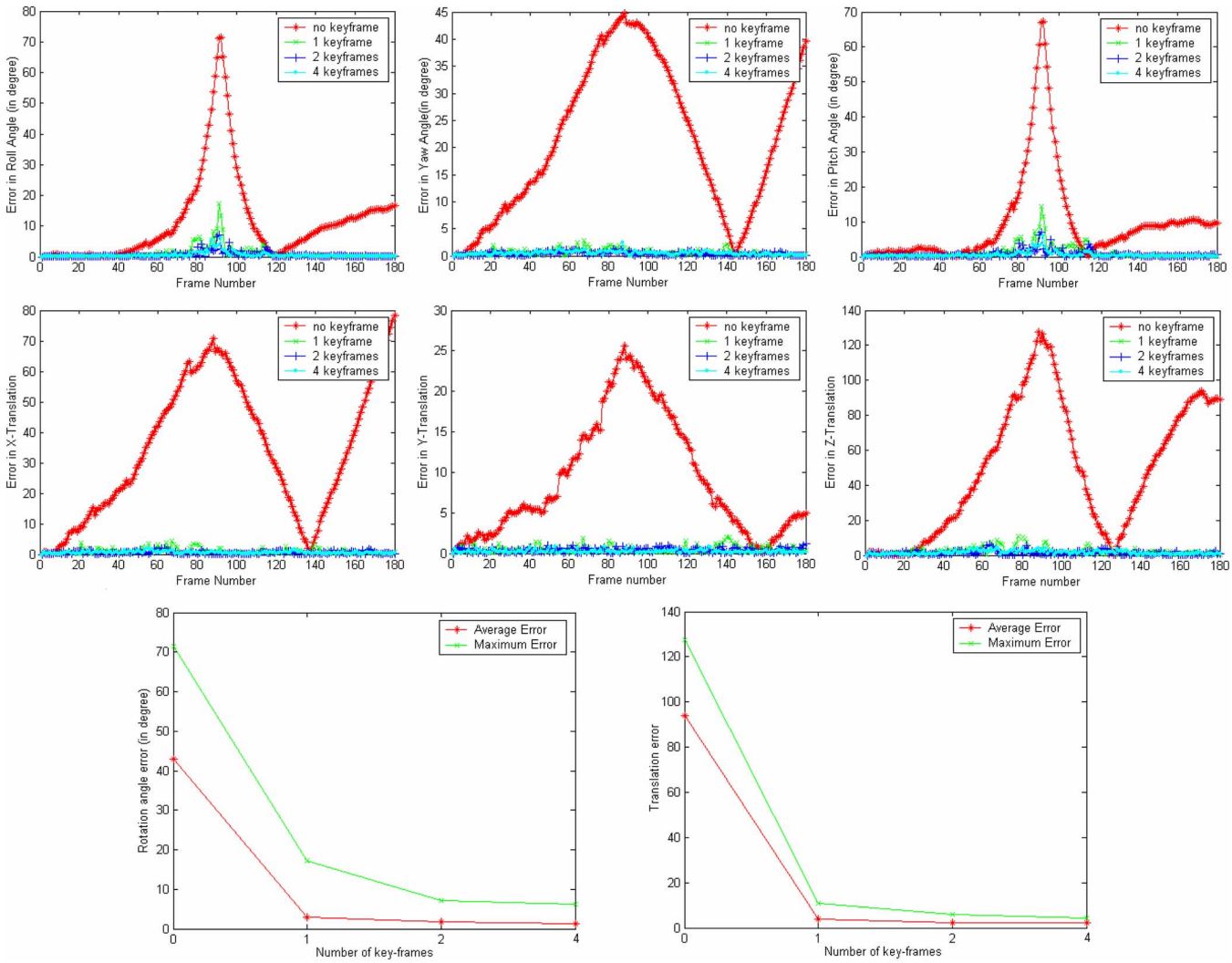

Fig. 4. Error on each pose parameter. The top row is rotational error in degrees; from left to right are errors in roll, yaw, and pitch angles. The middle row is translational error; from left to right are translational errors in X, Y, and Z axis. The bottom row is the average and maximum error with different number of key-frames; from left to right are rotation angle errors and translation errors.

each sample can get a weight proportional to the probability $\prod_{i=0}^{n} P(I_t, I_{t,i}|\delta_i^t, Y_i)$. The sample with the highest weight is output as the MAP estimation result. Since each term in the evaluation function has a very simple form, the estimation process is quite efficient for real-time processing.

VI. Experiments and Results

To test the validity of our proposed approach, both synthetic data and live captured real video were used. The tracking on both faces and a general object such as a toy car are performed. For synthetic data, we can compare the result with the ground truth. The live tracking result reflects the robustness and generality of the algorithm.

A. Synthetic Experiment

In this experiment, we use a generic head model to generate the synthetic test sequence. Since we simulate the feature selection and the point matching process, the main objective of the experiment is to evaluate the Bayesian online key-frame fusion process. The size of the generic model is about 100 (unit) and it contains about 180 vertices and 306 triangles in its mesh. Three kinds of noise are used to simulate real scenarios: the model error, the Gaussian point matching error, and the point mismatching error. We add Gaussian noise of s.t.d 2 (unit) to each model vertex. For point matching, we randomly select 100 model vertices as features, among which 80% have matches with a Gaussian noise of s.t.d 0.5 pixel, and the other 20% are modeled as mismatches which are added with uniform random noise in the range of [0, 50] pixels. The synthetic sequence is generated by rotating the generic face model from frontal view to profile view, and then rotating back. The motion is uniform so the inter-frame motion is 1 degree in yaw angle and there are totally 180 frames. The result is shown in Fig. 4. We can observe that the error accumulation is quite large during online tracking if there is no key-frame constraint. The drift error can be significantly reduced by introducing key-frame constraints.

Another thing observed from Fig. 4 is that the error drift depends on the direction of motion. For example, in the left figure of the top row of Fig. 4, the error in roll angle peaks at the profile view and decreases as the head rotates back. However, it starts

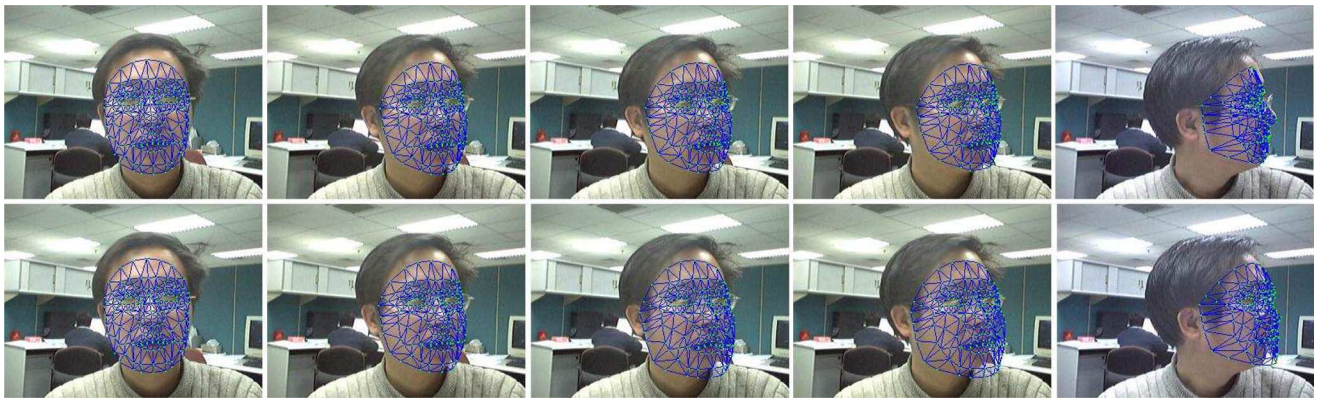

Fig. 5. Result comparison on agile rotation. From left to right: frame 156, 159, 160, 161, and 170. The top row is the Bayesian inter-frame motion estimation. The bottom row is the cross correlation point matching based inter-frame motion estimation as used in [9].

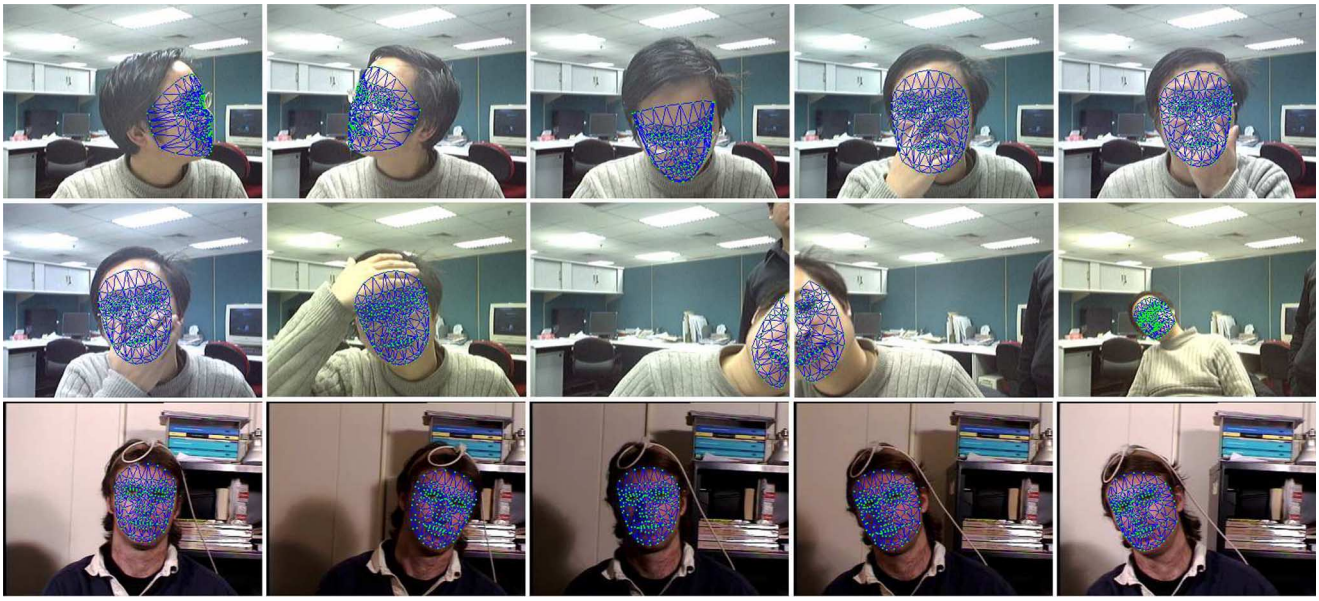

Fig. 6. Results on various situations. The first and second rows are extracted from a captured face tracking sequence. From left to right is frame 13, 35, 65, 133, 144, 158, 166, 189, 205, 225. There are severe occlusions, large object scale change and the motion is agile. The third row is extracted from a standard test sequence "jal8" from Boston University [6]. From left to right is frame 18, 72, 105, 122, and 139. Note that there is significant illumination change and we do not need the trained illumination basis as used in [6].

accumulating again after frame 120. The translational error in the z axis is much larger than that in the x and y axes.

B. Live Captured Video

In this experiment, we track a face captured by a USB camera. The whole tracking process is automatic: in the first frame, the approximately frontal face is automatically detected by a face detector [15] and facial features are automatically extracted using the face alignment algorithm [16]. Then the initial pose in the first frame is estimated using the POSIT algorithm [3] to initialize the pose tracker. Note that the face detector and alignment is never used again in the following tracking process. In all the face tracking experiments, we use the same generic face model as the one used in Section V-A. The USB camera's resolution is set to $320 * 240$ and is calibrated with focal length. 100 feature points are used to achieve the results in this paper and the algorithm's speed is about 30 fps on a P-IV2.8G computer. Fig. 5 compares the performance of the traditional correlation-based feature matching with pose estimation [9] and the proposed Bayesian inter-frame motion estimation algorithm in the case of agile motion. The key-frame fusion part is kept the same in order to make a fair comparison. It shows the advantage of the proposed Bayesian inter-frame motion estimation algorithm. More head tracking results are shown in Fig. 6 where there are severe occlusions, large object scale change, and drastic illumination changes. Note that for the first sequence in Fig. 6, we run the program in debug mode and save the captured video and the online tracking result video to hard disk simultaneously. As a result the video frame-rate is only about 10 fps, which is a more challenging case since the motion between two adjacent frames can be quite large. However, the algorithm still achieves stable result. For the second sequence shown in Fig. 6, we do not need the trained illumination basis and get a result comparable to that in [6]. Finally, we compared our real-time face tracking result subjectively with the state of the art method in [9]. When using a

Fig. 7. Result on toy car tracking. The top row is the original image and the bottom row is the tracking result. From left to right is frame 1, 70, 149, 203, and 251. Note that during rotation, there is significant changes in specular reflection.

normal USB camera, the image quality is relatively low and the illumination may change. Under these conditions, [9] becomes less stable especially for agile motion, large object scale change and the pose may significantly drift when the yaw and tilt angle is large than about 45 deg or more. In addition, when the tracker loses track, it cannot recover the correct pose again. The proposed method has significantly improved performance under the above difficult conditions.[1]

C. Toy Car Tracking

In this experiment, we show the ability of the proposed approach to handle general objects. Fig. 7 shows the results of tracking a toy car. The car model was created according to the photos taken at approximately frontal, top, and side views. It has 1937 vertices and 2532 triangles in its mesh. In the first frame, we need to manually select six points for estimating the car's initial pose, then the subsequent tracking is automatic.

VII. Discussion

In this section, we discuss some issues which can affect the 3-D pose tracking result. The first issue is the camera focal length. The pose error induced by inaccurate focal length depends on the position and shape of the object. This kind of error can be neglected in face tracking but becomes significant if the object's perspective effect is obvious. In our synthetic test for a 3-D cube, doubling the focal length can cause about a 10 deg rotational error in a normal configuration. The second issue is the model error. The algorithm is quite robust to local model error. However the significant error in aspect ratio for a model region with many features can degrade the pose estimation result, while for faces, the aspect ratio between eye–eye distance and eye–mouth distance is almost a constant across different people, so a generic model is enough for face tracking. The third issue is about motion ambiguity from noisy, locally aggregated point matches, in which case the change of 3-D pose does not have significant effects on 2-D point projections. For example, when we use a partial face model and the main observed region is a cheek with little texture, the reliable point matches will be located around the side view eye region. Such ambiguity may sometimes occur but can be immediately corrected in a few frames by using the Bayesian online key-frame fusion approach.

[1]More result tracking video sequences can be found at http://research.microsoft.com/~qiangwa/3DPoseTracking/index.htm.

VIII. Conclusion

A novel approach for real-time 3-D tracking of object pose from a single camera is proposed. The method is based on the probabilistic generative graphical model for pose tracking which can merge the information from 2-D point matching, relative pose estimation, and key-frame pose constraints with uncertainty. The combinatorial representation and inference approach results in an efficient algorithm which can stably track object pose in real-time from videos captured by a low cost camera, especially in the cases of agile motion, significant occlusion, large object scale changes, and drastic illumination changes which cannot be well handled by existing algorithms.

References

[1] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int. J. Comput. Vis.*, vol. 13, no. 3, pp. 331–356, 1994.

[2] D. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 5, pp. 441–450, May 1991.

[3] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 15, no. 1–2, pp. 123–141, 1995.

[4] C. P. Lu, G. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, Jun. 2000.

[5] J. Yao and W. K. Cham, "Efficient model-based linear head motion recovery from movies," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, Jun. 2004, vol. 2, pp. 414–421.

[6] M. L. Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3-D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.

[7] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Int. J. Imag. Syst. Technol.*, vol. 13, pp. 85–94, Sep. 2003.

[8] A. Rahimi, L. P. Morency, and T. Darrell, "Reducing drift in parametric motion tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, Jun. 2001, vol. 1, pp. 315–322.

[9] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1385–1391, Oct. 2004.

[10] P. H. S. Torr and C. Davidson, "IMPSAC: Synthesis of importance sampling and random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 354–364, Mar. 2003.

[11] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *Proc. IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, Jun. 2001, vol. 2, pp. 644–651.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. SIGGRAPH*, New York, 1985, pp. 245–254.

[14] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[15] R. Xiao, L. Zhu, and H. J. Zhang, "Boosting chain learning for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Washington, DC, Oct. 2003, vol. 1, pp. 709–715.

[16] Y. Zhou, L. Gu, and H. J. Zhang, "Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Madison, Wi, Jun. 2003, vol. 1, pp. 109–116.