Merging Artificial Objects with Marker-less Video Sequences Based on the Interacting Multiple Model Method

Ying Kin YU, Kin Hong WONG and Michael Ming Yuen CHANG

Abstract—**Inserting synthetic objects into video sequences has gained much interest in recent years. Fast and robust vision-based algorithms are necessary to make such an application possible. Traditional pose tracking schemes using recursive structure from motion techniques adopt one Kalman filter and thus only favour a certain type of camera motion. We propose a robust simultaneous pose tracking and structure recovery algorithm using the Interacting Multiple Model (IMM) to improve performance. In particular, a set of three extended Kalman filters (EKFs), each describing a frequently occurring camera motion in real situations (general, pure translation, pure rotation), is applied within the IMM framework to track the pose of a scene. Another set of EKFs, one filter for each model point, is used to refine the positions of the model features in the 3-D space. The filters for pose tracking and structure refinement are executed in an interleaved manner. The results are used for inserting virtual objects into the original video footage. The performance of the algorithm is demonstrated with both synthetic and real data. Comparisons with different approaches have been performed and show that our method is more efficient and accurate.**

Index Terms: **Augmented Reality, Pose Tracking, Interacting Multiple Model, Kalman filtering**

I. Introduction

Modern movie makers are interested in integrating cartoon characters into real scenes. Originally, producing these types of videos involves tedious work and requires experienced photo editors. With the aid of computers, this process can be automated. The process of mixing synthetically generated objects with image sequences in real-time is known as augmented reality or mixed reality. To produce augmented reality videos, the orientation of the scene with respect to the camera, i.e. the pose information, together with the structure of the scene, should be known. With knowledge of the structure, the virtual object can be placed anywhere in the scene and the pose information allows the motion of the object to be consistent with the background.

Manuscript received June 21, 2004; revised on May 14, 2005. This work was supported by the Research Grant Council of Hong Kong Special Administrative Region. (Project No. CUHK4204/04E)

Y.K.Yu and K.H.Wong are with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Hong Kong. (E-mail: {ykyu, khwong}@cse.cuhk.edu.hk).

M.M.Y.Chang is with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. (E-mail: mchang@ie.cuhk.edu.hk).

A. Previous work

General pose acquisition methods for inserting synthetic objects into images are based on the techniques in structure from motion (SFM) in computer vision [21]. SFM algorithms can estimate both the pose and 3-D structure from a sequence of 2-D images. Popular approaches include multiple view geometry [11] [12] and factorization [2] [13]. Bundle adjustment is also an effective method to recover the motion and model [4]. Its idea is to minimize the re-projection error between the estimated model and the image measurements. The minimization procedure is done in batch by the Newton's method. A branch of bundle adjustments is the interleaved bundle adjustment method [4] [14]. It breaks up the minimization problem into two steps so as to reduce the size of the Jacobian involved, resulting in speeding up the algorithm.

The method mentioned previously tackles the problem in a batch, in which the structure and motion are optimized for all the images at one time. In an interactive application like augmented reality, new measurements from images are acquired continuously from time to time and immediate reactions are required. Recursive techniques that recover the structure and motion sequentially with the images are highly useful, with which image measurements can be processed causally in real-time to give better performance. Most of the recursive approaches are based on Kalman filtering [15]. The series of methods in [5] [6] [7] [8] [20] recover both the structure and motion simultaneously using Kalman filters. The method in [7] is the seminal work in this series of researches. The authors applied a single full covariance iterated extended Kalman filter (IEKF) to recover the structure and pose of an object. Azarbayejani and Pentland described a method in [6] that has significant improvements over [7], where EKF is used as a substitute of IEKF. An extension is made to recover the focal length of the camera in addition to the pose and structure information. In other words, partial auto-calibration is possible. The pointwise structure is represented by one parameter per point with which the degree of freedom for motion, camera and structure becomes (6+1+N), where N is the number of point features. Since there are 2N measurement constraints plus one arbitrary scale constraint in each frame, the computation of parameters can be over-determined when the number of features in each

frame is larger than 7. Such an overdetermination results in better convergence and stability of the filter. The most recent work of recursive structure recovery is by Yu et al [20] [23]. One EKF is used to compute the motion of the scene while a set of EKFs, each corresponding to a feature point in the 3-D space, is applied to estimate the scene structure. The object motion and 3-D structure are calculated in an interleaved manner. Using this method of decoupling, the computation efficiency is increased as a tradeoff in accuracy. This algorithm has been applied to reconstruct 3-D models from 2-D images and insert synthetic objects into video sequences.

B. Our contributions

The Interacting Multiple Model based (IMM-based) method presented in this paper aims to track the pose sequence in videos using the techniques in SFM. In short, the Interacting Multiple Model (IMM) algorithm [10] is a suboptimal hybrid state filter that has been widely used as a tool for tracking maneuvering targets in RADAR [18] and vision-based systems [19]. The idea of applying the IMM algorithm to the SFM problem was inspired by the fact that accuracy can be increased if prior information about the structure or motion is utilized [9]. In the pose estimation step of our algorithm, three EKFs, each describing a unique motion dynamic, are adopted. They represent those frequently occurring camera motions (i.e. general, pure translation and pure rotation) in real situations. Intuitively, the IMM provides a mechanism to "select" suitable filters automatically in order to set constraints on the camera motion if prior information is available. With these constraints, the total number of parameters to be estimated is reduced and the accuracy can be improved. In addition, the problem of motion discontinuity in real images can be handled properly with the IMM framework.

Our structure and motion algorithm consists of a total of *N*+3 small EKFs, where *N* is the number of point features in the scene. With such an arrangement, the time complexity of the algorithm is lower than those traditional approaches that use a single full covariance EKF. This is necessary since the computation speed is crucial for augmented reality applications. Although each point in the 3-D space is updated by a separate EKF, the rigidity of the scene structure under reconstruction is maintained in our algorithm and is achieved by expressing the coordinates of the 3-D points in terms of their corresponding 2-D coordinates in the images.

C. Organization of the paper

The rest of this paper is organized as follows. Our geometric setup is introduced in Section II. The overview of our robust algorithm is then described in Section III. In Sections IV and V the detailed formulas for the IMM algorithm, the EKFs for pose estimation and structure refinement, are presented. In Section VI, the handling of the changeable set of feature points in our implementation is discussed. In Section VII, a computation comparison among our IMM-based approach and other existing algorithms [6] [14] [20] is made. In Section VIII, experiments with real and synthetic data are performed and the results from the four approaches are analyzed. In addition, the proposed algorithm has been applied to insert a synthetic object into a real image sequence.

II. Geometric Setup of the System

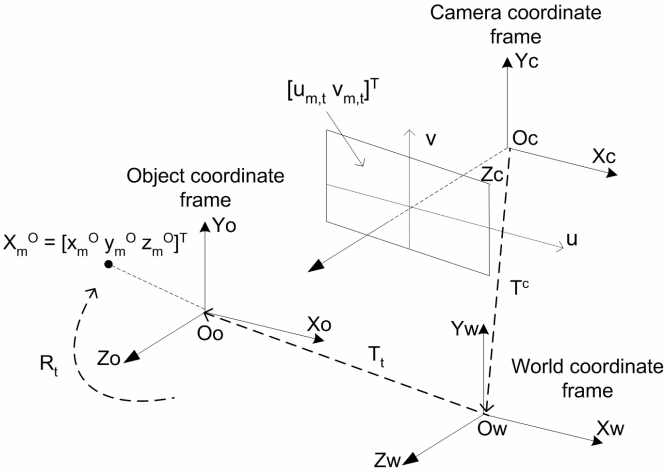


Fig. 1. The geometric setup of our system.

Fig. 1 shows the geometric setup of our system. X_m represents the m^{th} model point in the 3-D space. $X_m^O = [x_m^O, y_m^O, z_m^O]^T$ and $X_{m,t}^C = [x_{m,t}^C, y_{m,t}^C, z_{m,t}^C]^T$ are the coordinates of point X_m with respect to the object and the camera coordinate frame, respectively. t denotes the time-step. $p_{m,t} = [u_{m,t}, v_{m,t}]^T$ is a point on the image plane. The recovered structure is centered at the origin O_o in the object frame. The relationship between the object frame and the camera frame is as follows:

$$X_{m,t}^C = (R_t X_m^O + T_t) + T^C \tag{1}$$

where R_t is a 3x3 rotation matrix. It represents the object-centered rotation. T_t is a 3x1 translation vector. T^C is a 3x1 vector that brings the model structure from the world frame to the camera frame. It is a constant and regarded as a system parameter that can be measured during calibration. Knowledge of T^C is necessary if we recover the structure and motion of an object on a turntable. When $T^C = 0_{3x3}$, the position of the world center is equal to the camera center. Equation (1) becomes $X_{m,t}^C = R_t X_m^O + T_t$, which is the traditional expression of rigid transformation.

Parameters R_t and T_t compose of the pose sequence. The camera is calibrated with fixed focal length *f*. The camera model is full perspective and the projection can be expressed as:

$$\begin{bmatrix} u_{m,t} \\ v_{m,t} \end{bmatrix} = \frac{f}{z_{m,t}^C} \begin{bmatrix} x_{m,t}^C \\ y_{m,t}^C \end{bmatrix} \tag{2}$$

The problem of simultaneous recovery of structure and motion in our system is to compute the coordinates of model point X_m^O in the object coordinate frame and the pose sequence of the scene, i.e. the rotation R_t and translation T_t, with respect to the views at each time-step.

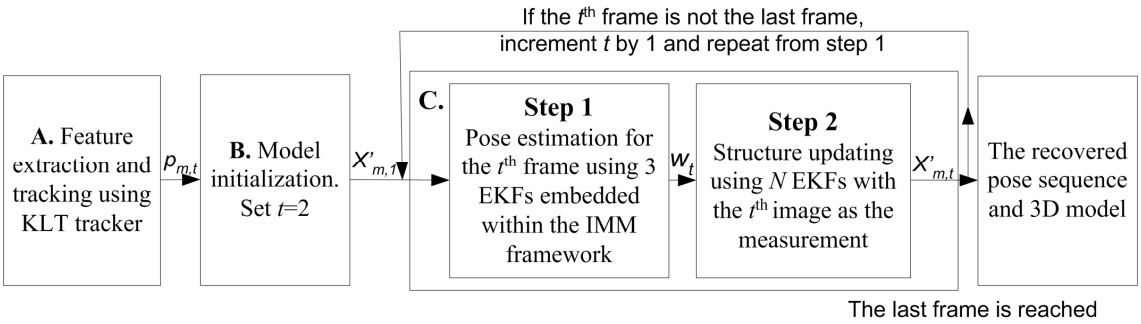


Fig. 2. The flowchart of the proposed IMM-based algorithm.

III. Overview of the Algorithm

The system can be divided into three parts: feature extraction and tracking, model initialization, pose estimation and structure updating. An illustration of the flow of the proposed algorithm is shown in Fig. 2.

A. Feature extraction and tracking

The Kanade-Lucas-Tomasi (KLT) tracker described in [3] is used to extract feature points from the scene and track them in the image sequence. In our work, it is assumed that the problem of feature tracking has been solved and point matches from the tracker are reliable enough for pose estimation and structure recovery.

B. Model initialization

The model is assumed to be a static rigid body. The initialization is achieved by letting the projection of the first image in the sequence be weak perspective. Perspective projection is assumed for the remaining frames so that our algorithm can deal with perspective images. The weak perspective projection is expressed mathematically as:

$$\begin{bmatrix} u_{m,t} \\ v_{m,t} \end{bmatrix} = \frac{f}{z_{init}} \begin{bmatrix} x^C_{m,t} \\ y^C_{m,t} \end{bmatrix} \tag{3}$$

where z_{init} is the distance between the model structure and camera center. It is a parameter given by the user of the system and can be approximated easily. To obtain the initial model, features in the first image are back-projected from the image plane to the camera coordinate frame according to (3). The resulting initial structure, represented by parameter $X'_{m,1}$, is a planar model located at a distance z_{init} from the camera. Such an initialization is fast but has a limitation. The depths of the object features should not vary too much to ensure proper convergence of the filters. Actually, other initialization procedures, such as the use of epipolar geometry plus the RANSAC robust estimator, can be employed despite of the computation speed.

C. Structure and pose updating

The initial model and the second image are fed to the first step of the main loop for pose estimation. Three extended Kalman filters (EKFs), each representing a unique motion dynamic, are adopted. These three filters interact with one another using the Interacting Multiple Model (IMM) [10]. The recovered pose w_t is a mixture of the outputs of the three filters and is then passed to the second step for structure updating.

The second step consists of a set of N EKFs, where N is the number of point features. Each EKF corresponds to one feature point in the recovered 3-D structure. With the observations and the pose recovered for the current image frame, the parameter $X'_{m,t}$ representing the coordinates of each feature point is updated. The algorithm alternates between step 1 and 2 until all images in the sequence are used.

IV. Step 1: Pose Estimation

The pose estimation step consists of three EKFs embedded within the IMM framework. Each of the three EKFs describes frequently occurring motion dynamics in real situations. The IMM algorithm provides a probability framework for filter switching.

A. Design of the individual EKFs

The three EKFs describing three different motion dynamics are defined as follows:

1) The General Motion Filter (GMF): GMF is designed to handle arbitrary object motion with unrestricted rotation and translation. Constant velocity is assumed for the GMF.

2) The Pure Translation Motion Filter (TMF): TMF is designed for tracking the objects with zero rotation motion.

3) The Pure Rotation Motion Filter (RMF): RMF is dedicated to tackle the objects with pure rotation around the y-axis (i.e. non-zero Pitch angle).

The state vector $w_t(i)$ of the i^{th} motion filter is:

$$w_t(i) = \begin{bmatrix} t_x & \dot{t}_x & t_y & \dot{t}_y & t_z & \dot{t}_z & \alpha & \dot{\alpha} & \beta & \dot{\beta} & \gamma & \dot{\gamma} \end{bmatrix}^T$$

where t_x, t_y, and t_z are the translation parameters of the object along the x, y and z axis, respectively. $\dot{t}_x, \dot{t}_y, \dot{t}_z$ are their corresponding velocities. α, β, γ are respectively the Yaw, Pitch and Roll angle with $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ as their corresponding angular velocities. As there are three filters in the IMM algorithm, i ranges from 1 to 3. The state transition and measurement equation for the filters are:

$$w_t(i) = A(i) w_{t-1}(i) + \eta'_t(i) \tag{4}$$

$$\varepsilon'_t = g_t(w_t(i)) + v'_t(i) \tag{5}$$

$$g_t(w_t(i)) = f\left[\frac{x_{1,t}^C}{z_{1,t}^C} \quad \frac{y_{1,t}^C}{z_{1,t}^C} \cdots \frac{x_{m,t}^C}{z_{m,t}^C} \quad \frac{y_{m,t}^C}{z_{m,t}^C} \cdots \frac{x_{N,t}^C}{z_{N,t}^C} \quad \frac{y_{N,t}^C}{z_{N,t}^C}\right]^T \quad (6)$$

where $\eta'_t(i)$ and $v'_t(i)$ are zero-mean Gaussian noise. ε'_t is a 2*N*x1 column vector representing the measurements from the images. $g_t(w_t(i))$ is the 2*N*x1-output image projection function. $X_{m,t}^C$ is computed by (1) and the rotation matrix R_t and translation vector T_t are evaluated with the parameters encoded in the column vector $w_t(i)$. $A(i)$ is a 12x12 block diagonal state transition matrix. $A(i)$ is different for the 3 EKFs and is defined as follows:

For GMF:

$$A(1) = A_{GMF} = diag\left\{\begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}\right\}$$

For TMF:

$$A(2) = A_{TMF} = diag\left\{\begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\right\}$$

For RMF:

$$A(3) = A_{RMF} = diag\left\{\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, \begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\right\}$$

where T_s is the sampling period. From the above dynamic system and measurement model, the four core Kalman filtering equations for pose estimation can be derived. The prediction equations for calculating the optimal estimates are:

$$\hat{w}_{t,t-1}(i) = A(i)\hat{w}_{t-1,t-1}(i) \quad (7)$$

$$P_{t,t-1}(i) = A(i)P_{t-1,t-1}(i)A(i)^T + Q_t'(i) \quad (8)$$

The update equations for the corrections of estimates are:

$$\hat{w}_{t,t}(i) = \hat{w}_{t,t-1}(i) + K(i)(\varepsilon'_t - g_t(\hat{w}_{t,t-1}(i))) \quad (9)$$

$$P_{t,t}(i) = P_{t,t-1}(i) - K(i)\nabla g_{w(i)}P_{t,t-1}(i) \quad (10)$$

$$K(i) = P_{t,t-1}(i)\nabla g_{w(i)}{}^T(\nabla g_{w(i)}P_{t,t-1}(i)\nabla g_{w(i)}^T + C'_t(i))^{-1}$$

$\hat{w}_{t,t-1}(i)$ and $\hat{w}_{t,t}(i)$ are the estimates of state $w_t(i)$ after the prediction and update, respectively. $P_{t,t-1}(i)$ and $P_{t,t}(i)$ are 12x12 matrices and are the covariances of $\hat{w}_{t,t-1}(i)$ and $\hat{w}_{t,t}(i)$, respectively. $K(i)$ is the 12x2*N* Kalman gain matrix for the filter. $\nabla g_{w(i)}$ is the Jacobian of the non-linear observation equation $g_t(w_t(i))$ evaluated at $\hat{w}_{t,t-1}(i)$. $Q_t'(i)$ and $C'_t(i)$ are the covariances of the noise terms $\eta'_t(i)$ and $v'_t(i)$, respectively.

The error covariances in EKFs are design parameters and can be tuned according to specific applications to obtain optimal performance. In practice, it is assumed that the noise of each parameter is uncorrelated. So the covariance matrices $Q_t'(i)$ and $C'_t(i)$ are chosen to be block diagonal. The error covariance of the measurement model $C'_t(i)$ can first be set in accordance with the accuracy of the imaging device and feature tracker used. Then the error covarinace of the object motion $Q_t'(i)$ is tuned such that the innovation process of the EKF becomes white. The value of $Q_t'(i)$ controls the degree of smoothness allowed for the object motion. These covariances are required to be set only once and are fixed in the Kalman filtering cycle.

EKF can be applied safely to our problem. Given that the frame rate of a video sequence is sufficiently high, the motion of the scene between successive images becomes small. In this case, the KLT tracker is reliable and we can assume that the tracker, together with the image sensor, only introduce Gaussian noise to the positions of the point features. The EKF is stable and effective under this condition.

B. The Interacting Multiple Model Algorithm

The IMM algorithm is a sub-optimal filter originally proposed by Blom [17]. It is regarded as one of most cost-effective hybrid state estimation schemes. It achieves an excellent compromise between performance and complexity. The algorithm adopted in the system is the baseline IMM in [10].

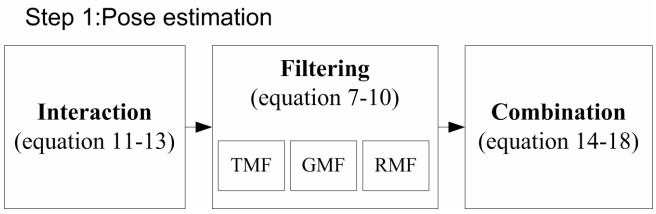


Fig. 3. The flowchart of the baseline IMM algorithm. The terms TMF, GMF and RMF are the short forms of the pure translation motion filter, the general motion filter and the pure rotation motion filter, respectively. The exact definitions can be found in Section IV.A.

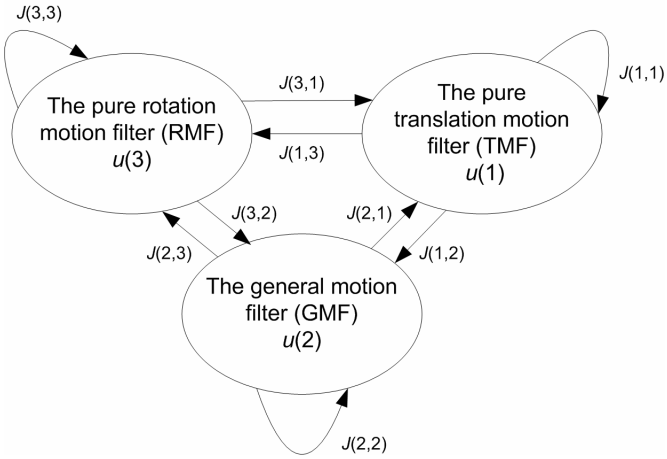


Fig. 4. The switching of motion filters in the IMM algorithm.

The basic IMM algorithm consists of several steps, which can be visualized in Fig. 3. Firstly, the likelihood of each filter $u_{t-1}(i)$ is updated according to the 3x3 switching matrix *J(i,j)*:

$$u_t^*(i) = \sum_j J(i,j)u_{t-1}(j) \quad (11)$$

where $u_t^*(i)$ is the likelihood probability of the filter after

interacting with the switching matrix *J(i,j)*. Notation *j*, in addition to *i,* is another index to the 3 EKFs for pose estimation. $J(i,j)$ denotes the probability of switching from filter *i* to filter *j*. In our implementation, the initial likelihood of the filters is equal. The switching matrix is set by assuming that the model under reconstruction continues with a single motion for an extended period of time with an occasional transition to another motion model. The diagonal entries of $J(i,i)$ is slightly less than 1 with off-diagonal entries $J(i,j) = (1 - J(i,i))/2$. Fig. 4 shows the switching process of the three EKFs in our system.

Secondly, the state estimates and their corresponding covariances of the previous time-step are mixed:

$$\hat{w}^*_{t-1,t-1}(i) = \frac{1}{u^*_t(i)} \sum_j J(i,j) u_{t-1}(j) \hat{w}_{t-1,t-1}(j) \tag{12}$$

$$P^*_{t-1,t-1}(i) = \frac{1}{u^*_t(i)} \sum_j J(i,j) u_{t-1}(j) \bullet \left(P_{t-1,t-1}(j) + [\hat{w}_{t-1,t-1}(j) - \hat{w}_{t-1,t-1}(i)][\hat{w}_{t-1,t-1}(j) - \hat{w}_{t-1,t-1}(i)]^T \right) \tag{13}$$

where $\hat{w}^*_{t-1,t-1}(i)$ and $P^*_{t-1,t-1}(i)$ are the state estimates and its covariance of filter *i* after the interaction with the switching matrix *J(i,j)*, respectively. They are then passed to the EKFs for prediction and smoothing with the measurements in the current time-step. The outputs of the i^{th} filter after the prediction phase are $\hat{w}_{t,t-1}(i)$ and $P_{t,t-1}(i)$ while that of the smoothing phase are $\hat{w}_{t,t}(i)$ and $P_{t,t}(i)$. After the Kalman filtering cycle, the likelihood of each filter $u_t(i)$ is updated with regard to the innovation vector $v_t(i)$ and its corresponding residual covariance $S_t(i)$ of the filters:

$$u_t(i) = \kappa \frac{u^*_t(i) \exp\left[-\frac{1}{2} v^T_t(i) S^{-1}_t(i) v_t(i) \right]}{\left\| S_t(i) \right\|^{\frac{1}{2}}} \tag{14}$$

$$v_t(i) = \varepsilon'_t - g_t(\hat{w}_{t,t-1}(i)) \tag{15}$$

$$S_t(i) = \nabla g_w P_{t,t-1}(i) \nabla g^T_w + C'_t \tag{16}$$

where κ is a normalization factor such that $\sum_i u_t(i) = 1$. $u_t(i)$ is computed according to an *n*-dimension zero-mean normal distribution function. In our system, a random sample of model features, having a size of *n*, is chosen to compute the innovation vectors and residual covariances for the calculation of the above normal distribution function. Experimentally, *n* is set to 10 to reduce the computation complexity.

Lastly, the usable output state vector $\hat{w}_{t,t}$ and covariance matrix $P_{t,t}$ at the current time-step *t* are generated with the following equations:

$$\hat{w}_{t,t} = \sum_i u_t(i) \hat{w}_{t,t}(i) \tag{17}$$

$$P_{t,t} = \sum_i u_t(i) \left(P_{t,t}(i) + [\hat{w}_{t,t}(i) - \hat{w}_{t,t}][\hat{w}_{t,t}(i) - \hat{w}_{t,t}]^T \right) \tag{18}$$

The final output of the system, i.e. $w_t = \hat{w}_{t,t}$, is a linear sum of the smoothed state and covariance estimates of each filter weighted by the corresponding updated filter likelihood. With the IMM algorithm and three EKFs, the pose of the model can be estimated.

V. Step 2: Structure Updating

The structure updating step consists of *N* identical extended Kalman filters (EKFs), each corresponding to one model point in the 3-D space. For simplicity, one filter is considered in the discussion. The model is assumed to be static. The dynamic model of a 3-D point and its measurement equations are:

$$X'_{m,t} = X'_{m,t-1} + \eta_t \tag{19}$$

$$\varepsilon_{m,t} = h_t(X'_{m,t}) + v_t \tag{20}$$

$$h_t(X'_{m,t}) = f \begin{bmatrix} \frac{x^C_{m,t}}{z^C_{m,t}} & \frac{y^C_{m,t}}{z^C_{m,t}} \end{bmatrix}^T \tag{21}$$

where η_t and v_t are the zero-mean Gaussian noise. $\varepsilon_{m,t}$ is the real measurement from the image sequence. $h_t(X'_{m,t})$ is the projection function, in which $X^C_{m,t}$ is obtained by substituting suitable values into (22) and (1). $X'_{m,t}$ is a scalar that represents a model point:

$$X^C_{m,1} = X^O_m + T^C = \frac{X'_{m,t}}{f} \begin{bmatrix} u_{m,1} \\ v_{m,1} \\ f \end{bmatrix} \tag{22}$$

Each model point is represented by a single parameter. Such a representation is made under the assumption that the measurements acquired by the camera are non-biased. Intuitively, the 3-D coordinates of the points are expressed in terms of the first images that the features appear. Detailed discussions on the advantages arising from this structure representation and the method to handle biased measurement can be found in [6].

With the dynamic system and measurement model, the required equations for the EKF, which are similar to those in (7) – (10), can be derived. Due to limited space, readers please refer to [15] for further details.

VI. Handling the Changeable Set of Feature Points

The set of active feature points is changing due to occlusion. Extra treatments are needed in the structure acquisition process.

New model points in the structure are initialized when new point features appear in the image sequence. This is obtained by assuming the projection of that point on its first appeared image, say on the frame at time-step *t*, is weak perspective. The initial position, expressed in the camera coordinate frame, is computed according to (3). The structure parameter for that point is now expressed in terms of its image coordinates at time-step *t*. The relationship between the structure parameter $X'_{m,t}$ and object coordinate frame is:

$$X_m^O = R_t^{-1}\left\{\left(\frac{X'_{m,t}}{f}\begin{bmatrix} u_{m,t} \\ v_{m,t} \\ f \end{bmatrix}\right) - T_t\right\} \quad (23)$$

A new EKF, as described in Section V, is set up to refine its position in its camera coordinate frame with parameter $X'_{m,t}$. The final position can be obtained by calculating its coordinates in the object frame using (23).

When a point feature vanishes from the image sequence, the filter that corresponds to the point is removed and the 3-D position of that feature will no longer be updated.

VII. Experiments and Results

A. Experiments with synthetic data

The first set of experiments was conducted with synthetic data. A synthetic structure with 300 random feature points in 3-D within a cube of volume of 0.13m^3, centered at a place 0.33m away from the viewing camera, was generated. The camera motion was composed of three different segments, a pure translation section, a pure rotation section and a general motion section. The motion parameters were generated randomly from 0.05 to 0.15 degrees per frame for Yaw, Pitch, Row angle and 0.0005 to 0.0015 meters per frame for t_x, t_y and t_z. The length of each synthetic sequence was 99 frames. A total of 20 independent tests were carried out. Our IMM-based algorithm was implemented in Matlab and tested with the data. The interleaved bundle adjustment method [14], the EKF by Azarbayejani and Pentland [6] and the 2-step EKF by Yu et al [20] were also implemented in the same platform and tested with the same set of data for comparison.

Table I shows the average pose parameter errors per frame for the four algorithms under the 20 test cases. The total rotations of the actual and recovered structure were calculated using the axis-angle representation to reduce the Yaw, Pitch, Roll angle into a single angle. The difference between these two values is the error. The total translation error was computed by subtracting the recovered translation vector from the actual one and the magnitude was taken. The errors per frame are equal to dividing the summation of errors by the number of frames. You can see that our approach achieved the lowest total rotation and translation error per frame.

Fig. 4 shows the time for the four algorithms to optimize the image residual error of the back-projected model. For the plots in Figs. 4 and 5, the lines with an asterisk (*), triangle (△), circle (○), square (□) are for our IMM-based approach, the interleaved bundle adjustment method, the EKF by Azarbayejani and Pentland and the 2-step EKF by Yu et al, respectively. From Fig. 4, the error of our algorithm falls to a low value at the earliest time among the four methods. The final residual error is the second lowest in the comparison. It is reasonable that the interleaved bundle adjustment method (a batch processing method) had a lower error than ours, since our recursive algorithm could not optimize the structure and pose error for all the images in the sequence simultaneously. However, the interleaved bundle adjustment method is ranked the third in pose accuracy (see Table I). The solutions found by their algorithm overfit the data.

Fig. 5 shows the time needed to reconstruct a model when extra frames were added sequentially to the image sequence. Our algorithm took 0.79 seconds to update the structure and pose for every extra frame. It outperformed the EKF by Azarbayejani and Pentland and the interleaved adjustment method, which needed 2.60 and at least 4.55 seconds, respectively. The 2-step EKF by Yu et al took 0.42 seconds to process an extra frame. However, experimental results show that the 2-step EKF resulted in a total rotation and translation error that were ten and four times larger than our approach, respectively. It shows that our IMM-based approach was a better tradeoff between time and accuracy since a little addition of computation cost could cause a significant improvement on the resulting errors.

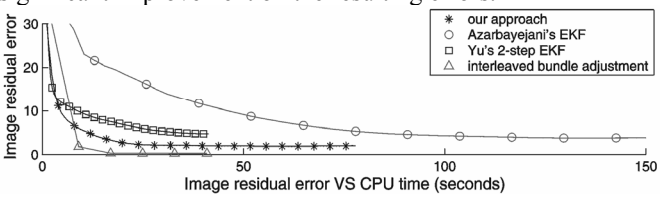


Fig. 4. The relationship between the CPU time and the image residual error. Note that the algorithms were implemented in Matlab with a Pentium III 1GHz machine.

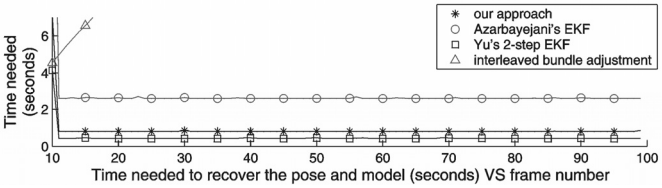


Fig. 5. A graph showing the time needed for the 4 algorithms to reconstruct the model and pose when extra frames were added to the image sequence.

B. Experiments with real images

An experiment using real scene images was also performed. The sequence was taken in the laboratory. The images were captured while the camera was translating sideway on a rig. The length of the image sequence was 100 frames. To ensure the algorithm would achieve high accuracy and stability, the camera motion between successive image frames could not be too large. Otherwise, drifting of pose may occur due to the limitation of the KLT feature tracker. The KLT tracker assumes that the 2-D motion of a feature point is not large. In real situations, the motion can be virtually slowed down given that the frame rate of the image sequence is sufficiently

TABLE I
THE AVERAGE ERROR OF EACH POSE PARAMETER

Symbol	Roll	Pitch	Yaw	t_x	t_y	t_z	Total rotation	Total translation
Our approach	0.4055	0.2411	0.0257	0.0750	0.2196	0.1867	0.3356	0.32055
Azarbayejani's EKF	1.0467	0.5496	0.0205	0.1102	0.2732	0.2172	0.6669	0.38894
Yu's EKF	3.7069	3.3599	0.3487	0.2	0.4	1.2	3.7747	1.4
The interleaved bundle adjustment method	2.6789	1.3259	0.1141	0.1	0.2	1.4	1.3405	1.5

A table showing the average errors of each pose parameter per frame of the 4 algorithms in the experiment. Note that the angular errors (i.e. the total rotation, the Roll, Pitch and Yaw angle error) are in degrees and the translational errors (i.e. the total translation, t_x, t_y and t_z error) are in meters.

high.

Fig. 6. Results of inserting an artificial object into the test scene using our IMM-based approach. First row: The first and the last image of the test sequence. Second row: A synthetic car, which was drawn by wire-frames, was put into the real scene. More results can be found at http://www.cse.cuhk.edu.hk/~khwong/demo/

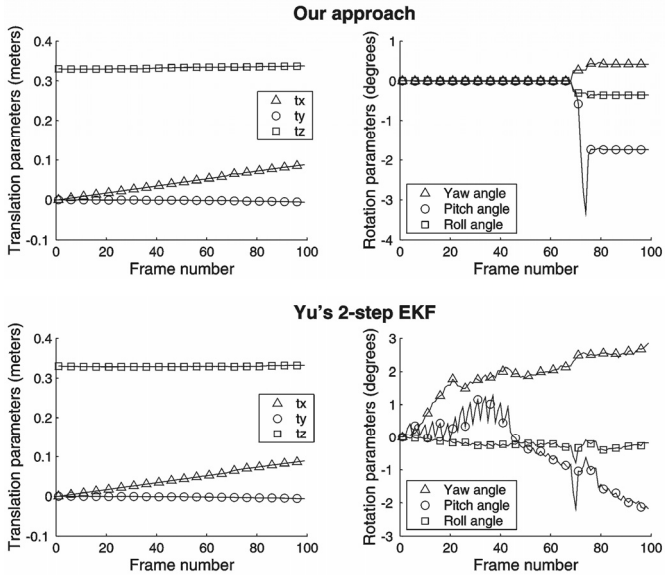


Fig. 7. The pose sequences recovered from the test sequence with our IMM-based approach (the top two plots) and with the 2-step EKF by Yu et al (the bottom two plots).

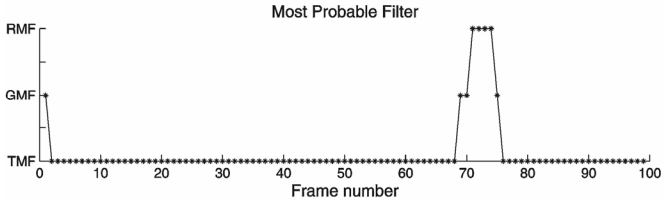


Fig. 8. A plot showing the most probable EKF for pose estimation against frame number resulting from the test image sequence.

Our IMM-based algorithm was applied to track the pose sequence from the video while reconstructing the scene structure. The recovered pose was used to produce an augmented reality video, in which a synthetic car was inserted into the scene.

Fig. 6 represents the result. The orientation of the synthetic car is consistent with the real scene in the whole video sequence. To show the significance of using the IMM algorithm, a comparison between our IMM-based approach and the 2-step EKF by Yu et al [20] in this real image experiment was made. The latter algorithm was chosen as a control since it is similar to our approach in a way that the recovery of pose and structure in the Kalman filtering computation is decoupled. The major differences between these two approaches are the structure representation and the use of the IMM algorithm in pose estimation.

Fig. 7 shows the resulting pose sequences acquired with the two algorithms. The pose recovered with our IMM-based algorithm was smooth with no ambiguities among the Yaw and Pitch angle. The small jump in the rotation angles at the 76th frame was due to the small vibration of the camera. On the other hand, the rotation parameters recovered using the 2-step EKF fluctuated even if the camera motion was almost pure translation.

Fig. 8 shows the switching of the three EKFs for pose estimation in our IMM-based approach. The filter was regarded as "switched to" (in-use) if it had the highest likelihood among the three filters. You can see that the IMM algorithm detected the camera motion correctly. Our system used the pure translation motion filter (TMF) most of the time in this test case, which reflected the actual motion of the camera. It switched to the other two filters, i.e. the general motion filter (GMF) and the pure rotation motion filter (RMF), from the 69th to the 75th frame. This was due to the fact the camera was vibrating. After that, our system switched back to TMF.

VIII. Conclusion

A recursive algorithm that targets to track the camera motion and recover the 3-D structure simultaneously has been proposed and tested in this paper. The Interacting Model Multiple Model (IMM) has been applied to solve the problem in association with the extended Kalman filters (EKFs). With the IMM, the ambiguities among the pose parameters and recovered structure have been resolved successfully, resulting in a higher accuracy on the recovered pose sequence. The required computation time is kept to a minimum by breaking up pose estimation and structure acquisition into two steps and each corresponding point in the 3-D model is decoupled in the EKF implementation. At the same instance, a special structure representation has been adopted to maintain the rigidity of the 3-D model, thus minimizing the effects of decoupling the EKF. The proposed algorithm attains an optimal tradeoff between speed and accuracy. Also, a scheme on handling the changeable set of point features has been devised. These advantages make our approach best suit for the augmented reality applications of marker-less video sequences.

Theoretically, the number of EKFs embedded in the IMM algorithm is not limited. If an additional EKF is able to

describe the system dynamics of a particular application, it can be incorporated into the algorithm to improve the accuracy. The addition of the filter does not affect the speed significantly if the algorithm is implemented on a parallel processing system as the filters can run concurrently. To proceed further, we can apply or formulate some sophisticated dynamic systems as described in [22] to make the algorithm more robust under a wider range of conditions with the minimum number of EKFs.

IX. Acknowledgment

The work described in this paper was supported by a grant from the Research Grant Council of Hong Kong Special Administrative Region. (Project No. CUHK4204/04E)

References

[1] P.A. Beardsley, A.Zisserman and D.W.Murray, "Sequential updating of projective and affine structure from motion", *International Journal of Computer Vision*, vol. 23, no. 3, pp. 235-259, 1997.

[2] C.Tomasi and T.Kanade, "Shape and motion from image streams under orthography: A factorization method", *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.

[3] C.Tomasi and T.Kanade, "Detection and tracking of point features", Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[4] B.Triggs, P.McLauchlan, R.Hartley and A.Fitzgibbon, "Bundle adjustment – A modern synthesis" In proc. of the Intl. Workshop on Visual Algorithm: Theory and Practice, pp. 298-372, Corfu Greece, 1999.

[5] A.Chiuso, P.Favaro, H.Jin and S.Soatto, "Structure from motion causally integrated over time", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523-535, 2002.

[6] A.Azarbayejani and A.P.Pentland, "Recursive estimation of motion, structure, and focal length", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562-575, June 1995.

[7] T.J.Broida, S.Chandrasiiekhar and R.Chellappa, "Recursive 3-D motion estimation from monocular image sequence", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639-656, July 1990.

[8] A.J.Davison and D.W.Murray, "Simultaneous localization and map-building using active vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865-880, July 2002.

[9] R.Szeliski and S.B.Kang, "Shape ambiguities in structure from motion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 506-512, May 1997.

[10] E.Mazor, A.Averbuch, Y.Bar-Shalom and J.Dayan, "Interacting multiple model methods in target tracking: A survey", *IEEE Transactions on Aerospace and Electrons Systems*, vol. 34, no. 1, pp. 103-123, January 1998.

[11] E.Trucco, A.Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.

[12] R.Hartley and A.Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[13] C.J.Poelman, T.Kanade, "A paraperspective factorization method for shape and motion recovery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206-218, March 1997.

[14] M.M.Y.Chang and K.H.Wong, "Model reconstruction and pose acquisition using extended Lowe's method", *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 253-260, April 2005.

[15] M.S.Grewal, A.P.Andrews, *Kalman Filtering Theory and Practice*, Prentice Hall, 1993.

[16] V.Lippiello, B.Siciliano and L.Villani, "Position and orientation estimation based on Kalman filtering of stereo images", in proc. of the IEEE International Conference on Control Applications, pp. 702-707, Mexico City, 2001.

[17] H.A.P.Blom, "An efficient filter for abruptly changing systems", in proc. of the 23rd IEEE Conf. on Decision and Control, pp. 656-658, Las Vegas, NV, December 1984.

[18] E.Daeipour and Y.Bar-Shalom, "An interacting multiple model approach for target tracking with glint noise", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 2, pp. 706-715, April 1995.

[19] K.J.Bradshaw, I.D.Reid and D.W.Murray, "The active recovery of 3D motion trajectories and their use in prediction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 219-234, March 1997.

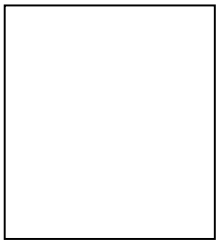
[20] Y.K.Yu, K.H.Wong and M.M.Y.Chang, "Recursive three-dimensional model reconstruction based on Kalman filtering", *IEEE Transactions on Systems, Man and Cybernetics- Part B*, vol.35, no. 3, pp. 587-592, June 2005.

[21] S.Gibson, J.Cook, T.L.J.Howard, R.J.Hubbold, and D.Oram. "Accurate camera calibration for off-line, video-based augmented reality", in proc. of IEEE the International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, September 2002.

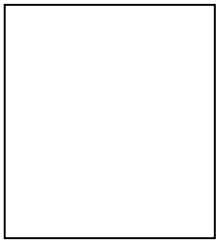
[22] X.R.Li and V.P.Jilkov, "Survey of maneuvering target tracking part I-Dynamic model", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333-1364, October 2004.

[23] Y.K.Yu, K.H.Wong and M.Y.Y.Chang, "A fast recursive 3D model reconstruction algorithm for multimedia applications", in proc. of the International Conference on Pattern Recognition, Cambridge, August 2004.

Ying Kin Yu received a B.Eng (First Class Honours) and an M.Phil. degree from the Chinese University of Hong Kong in 2002 and 2004, respectively. He is now a Ph.D. student in the Department of Computer Science and Engineering in the same university. He was awarded the Sir Edward Youde Memorial Fellowship twice for his academic achievements. His research interests are computer vision, augmented reality, Kalman filtering and genetic algorithms.



Kin Hong Wong received a B.Sc. in Electronics and Computer Engineering from the University of Birmingham in 1982, and a Ph.D. from the Engineering Dept. of the University of Cambridge, U.K. in 1986. He was a Croucher research fellow at the University of Cambridge from 1985 to 1986. Prof. Wong joined the Computer Science Dept. of CUHK in 1988 and is now an Associate Professor. His research interests are 3-D computer vision, virtual reality image processing, pattern recognition, microcomputer applications and computer music.



Michael Ming Yuen Chang received the B.Sc. in electrical engineering from Imperial College, London University and the PhD degree in electrical engineering from University of Cambridge in 1988. He then joined the Department of Information Engineering, The Chinese University of Hong Kong and is now an Associate Professor. His current research interest is in character recognition, scientific visualization and intelligent instrumental control.