# Recursive Camera Motion Estimation with Trifocal Tensor

Ying Kin Yu*, Kin Hong Wong, Michael Ming Yuen Chang and Siu Hang Or

*Abstract*— **We propose an innovative extended Kalman filter (EKF) algorithm for pose tracking using trifocal tensor. In the EKF, a constant velocity motion model is used as the dynamic system and the trifocal tensor constraint is incorporated into the measurement model. The proposed method has the advantages of those structure and motion (SAM) based approaches in that the pose sequence can be computed with no prior information on the scene structure. It also has the strengths of those model based algorithms in which no updating of 3D structure is necessary in the computation. This results in a stable, accurate and an efficient algorithm with low time and space complexity. Experimental results show that our approach outperformed other existing EKFs that tackle the same problem. An extension to the pose tracking algorithm has been made to demonstrate the application of the trifocal constraint to fast recursive 3D structure recovery.**

*Index Terms:* **Pose Tracking, Augmented Reality, Kalman Filtering, Trifocal Tensor**

## I. INTRODUCTION

A fast and robust pose acquisition algorithm is crucial to interactive applications such as augmented reality and robot navigation. An accurate pose estimation method is also important for the recovery of the 3D structure, since a high precision depth map can be constructed with an optimal pose sequence. This paper describes an innovative Kalman filter based approach to tackle the classic vision-based pose tracking problem. With no prior information about the 3D structure of the scene, the camera motion can be recovered from a monocular image sequence directly with the trifocal constraint. In addition to pose tracking, an extension of our algorithm has been made to solve the structure and motion (SAM) problem. The performance of the algorithm is demonstrated by inserting an artificial object into a real image sequence.

Ying Kin Yu is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: ykyu@cse.cuhk.edu.hk Phone:+852-26098438 Fax:+852-26035024

Kin Hong Wong is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: khwong@cse.cuhk.edu.hk Phone:+852-26098397 Fax:+852-26035024

Michael Ming Yuen Chang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: mchang@ie.cuhk.edu.hk Phone:+852-26098347 Fax:+852-26035032

Siu Hang Or is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: shor@cse.cuhk.edu.hk Phone:+852-31634261 Fax:+852-26035024

### A. *Previous work*

Broadly speaking, there are four major ways to solve the problem of pose tracking. Below is a brief illustration of different approaches.

#### *1) Marker based approaches*

To estimate the camera motion from an image sequence, a popular method is based on the visual marker detection and decoding techniques. In that, the scene for processing must contain a special pattern for pose estimation and camera calibration [31]. This technique has been widely used in augmented reality systems such as direct marketing in electronic commerce [34], tele-conferencing [33], entertainment [35] and manufacturing of door-locks [32]. The major weakness of the marker based approaches is that special preparation in the scene is required and thus not feasible for computing the camera motion from old video footages or ad-hoc sequences.

#### *2) Model based approaches*

An early approach to acquire the pose sequence from a video without markers is the three-point algorithms [23]. To increase the accuracy of the solutions, more points are needed and geometric information on the scene is required. This is known as the model based method. Horaud et al [24] estimated the pose of an object using four non-coplanar points. Another four-point algorithm is by Liu and Wong [25]. Lowe proposed an iterative method [26] to solve the pose estimation problem using $N$ model points. The work [29] [30] uses the extended Kalman filter (EKF) to find the pose of the object based on a known CAD model from stereo images. The position and orientation are recovered in real-time and the results are applied to visual servoing of robot manipulators. Alternative model based approaches adopt genetic algorithms. Hati and Sengupta [27] used the genetic algorithm framework to estimate the extrinsic parameters of a camera. Yu et al [41] improved their approach by incorporating a mismatch filtering strategy into the genetic algorithm using composite chromosomes and the results are applicable to augmented reality. Toyama et al [28] took the advantages of the phenotypic forking genetic algorithm to find the pose from the edge images. In case that neither markers can be placed into the scene nor the scene structure is known, more general approaches based on the techniques of structure and motion is necessary.

#### *3) Structure and motion based approaches*

One of the most popular structure and motion (SAM) based approaches is the use of multiple view geometry [14] [15] [17], in particular the epipolar geometry. With the known correspondences between the two views of the same object, a constraint between these two views can be set up. The camera motion can be recovered from the resulting fundamental matrix up to a scale factor if the camera is fully calibrated. Some researchers extended this technique to three views or more [15]. Avidan and Shashua [42] made use

of the trifocal tensor to concatenate the fundamental matrices such that the camera matrices recovered from an image sequence are consistent with each other. Another variation of multiple view geometry is to relate the views taken by cameras of different camera models [18].

Factorization [19] [20] [21] is another general approach. The work in [19] demonstrates the concept under the assumption of orthographic projection. The factorization method has been extended to para-perspective projection [21] and to handle the reconstruction and pose estimation of multiple independently moving objects [20]. Bundle adjustment is also an effective method to recover the motion and structure [16]. The idea is that it minimizes the re-projection error between the estimated model and the image measurements. The minimization procedure can be done in batch either by the well-known Newton's or Levenberg-Marquardt iteration. A branch of it is the interleaved bundle adjustment as described in [16] and [22]. 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. These approaches tackle the problem either in a batch or with a robust statistical estimator, resulting in a certain degree of latency for interactive applications like augmented reality.

*4) Recursive structure and motion based approaches*

To minimize the latency, some researchers developed a partial bundle adjustment scheme [44] in that the structure and motion are computed every three views. However, more elegant recursive solutions to the SAM problem require the use of Kalman filters [11]. In [2] and [1], the iterated extended Kalman filter (IEKF) has been adopted to update the structure in Euclidean and projective framework respectively. However, real-time implementation of [1] is impossible since it involves the use of the RANSAC robust estimator in pose estimation. The series of methods in [3] [4] [5] [6] [7] [8] [9] [10] recover both the structure and motion simultaneously using Kalman filters. The work by Broida et al [5] is the ancestor of this series of researches. They applied a single full-covariance IEKF to recover the structure and pose of an object. Azarbayejani and Pentland described a method in [4] that makes significant improvements over [5]. An extension was made to recover the focal length of the camera in addition to the pose and structure. The pointwise structure is represented by one parameter per point such that the computation is over-determined at every frame when the number of features is larger than 7, resulting in a better convergence and stability of the filter. The most recent Kalman filter based methods are by Yu et al [8] [9] [10]. In [8] and [9], the full covariance extended Kalman filter (EKF) is decoupled such that the computation efficiency is increased as a tradeoff in accuracy. The authors then extended their work by adding the Interacting Multiple Model into the original formulation so as to keep the accuracy at least the same as the full covariance EKFs [10]. Soatto et al [43] applied the essential constraint in epipolar geometry to Kalman filter based motion

estimation so that the pose sequence can be computed directly from images. Similar techniques in SAM have also been applied to simultaneous localization and map-building for robot navigation [6], in which the concern is the demand of computation resources and the issue of repeatable localization.

### *B. Our contributions*

This paper focuses on the recovery of camera motion from image sequences with no prior information about the scene. From the literature we encountered, we believe that we are the first to apply the trifocal tensor to recursive pose tracking and SAM problem. Here we summarize our contributions.

**Integration of the trifocal tensor constraint into Kalman filtering.** The major contribution of the proposed approach is the incorporation of the trifocal tensor into the Kalman filtering formulation. This enables us to apply the strengths of the traditional multiple view geometry to recursive visual tracking. In the algorithm, the trifocal tensor point transfer function is used in the measurement model of the extended Kalman filter. That is apart from the dynamic system constraint on the motion of the camera, the trifocal constraint is also employed in the Kalman filtering cycle. With this additional constraint, the accuracy of the solution can be improved significantly.

**Independence of pose and 3D structure.** A unique characteristic of the proposed algorithm is that the pose sequence is recovered directly in the computation. In other words, pose tracking is dependent of structure recovery in the algorithm. Unlike traditional recursive SAM algorithms, both the 3D structure and pose parameters are required to be updated from frame to frame simultaneously [3] [4] or in an interleaved manner [8]. This implies the search space of our problem is reduced from $N$+6, where $N$ is the total number of available point features, to 12 (excluding the velocity parameters) in our implementation. The convergence rate of our filter can thus be increased. In addition, handling the changeable set of point features is easier than existing Kalman filter based methods as the 3D model features are not involved in the Kalman filtering cycle.

**Reduction of time and space complexity.** The use of trifocal tensor point transfer function also contributes to the increase in speed and reduction in storage requirement while keeping the advantages of the full covariance EKF. The time and space complexity of the proposed approach are respectively $O(N^2)$ and $O(N)$, which is much smaller than that of the traditional EKFs for the SAM problem [4] [5]. The corresponding time and space complexity for the traditional methods are $O(N^3)$ and $O(N^2)$ respectively.

Besides pose tracking, the formulation has been extended to acquire the structure of the scene. An extra set of EKF is adopted to update the 3D model structure with the newly recovered pose. Since the EKF for pose estimation and structure updating are independent of each other, the accuracy is not affected, or on the contrary improved, even the computation of pose and structure is decoupled. Evaluations using both

synthetic data and real images on the proposed algorithm have been made. The results show that our algorithm has a better overall performance over the existing recursive SAM based approaches [4] [8]. Our method suits best the development of the next generation of augmented reality applications.

### C. *Organization of the paper*

The rest of this paper is organized as follows. The geometry of our system is first introduced in section II. The overview of the proposed pose tracking algorithm is then described in section III. In section IV, the formulation of the extended Kalman filter is presented. In section V and VI, an extension of our algorithm to the structure and motion problem is illustrated. In section VII, the advantages of our recursive approach over existing methods are discussed. In section VIII, an empirical comparison among our approach, the EKF by Azarbayejani and Pentland [4] and the 2-step EKF by Yu et al [8] [9] is made using real and synthetic data. The results from these three approaches are analyzed.

## II. GEOMETRY OF THE SYSTEM

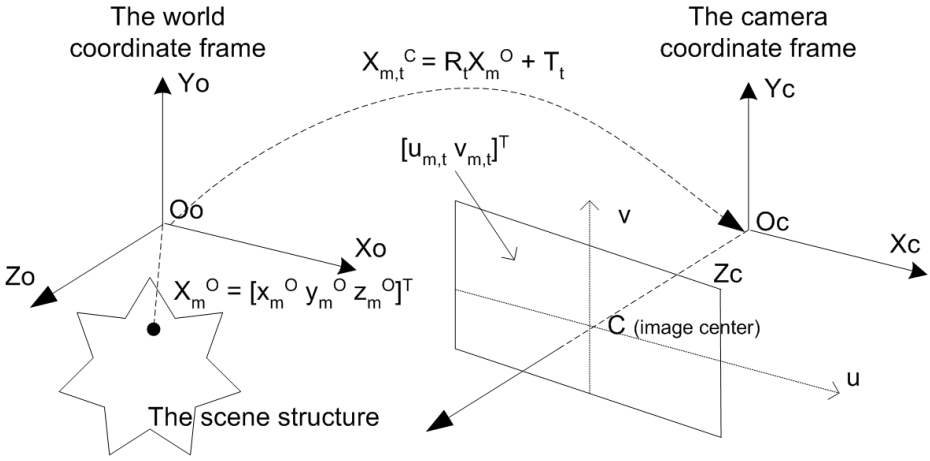


Figure 1. The geometry of our system.

The relationship between the 3D coordinates of a point in the scene structure and its 2D projection on the image plane can be related linearly as:

$$\begin{bmatrix} \tilde{u}_{m,t} \\ \tilde{v}_{m,t} \\ \tilde{w}_{m,t} \end{bmatrix} = M_t \begin{bmatrix} x_m^O \\ y_m^O \\ z_m^O \\ 1 \end{bmatrix} \tag{1}$$

$M_t$ is the 3x4 camera projection matrix at time $t$. $X_m^O = [x_m^O, y_m^O, z_m^O]^T$ denotes the coordinates of the model point $X_m$ with respect to the world coordinate frame. The actual image coordinates $p_{m,t} = [u_{m,t}, v_{m,t}]$ are

given by:

$$\begin{bmatrix} u_{m,t} \\ v_{m,t} \end{bmatrix} = \begin{bmatrix} \tilde{u}_{m,t} / \tilde{w}_{m,t} \\ \tilde{v}_{m,t} / \tilde{w}_{m,t} \end{bmatrix} \tag{2}$$

Assuming that the camera is calibrated with a fixed focal length $f$ and the motion of the camera relative to the scene structure is rigid, the projective relation in (1) can be rewritten as:

$$X^{C}_{m,t} = R_t X^{O}_{m} + T_t \tag{3}$$

$$\begin{bmatrix} u_{m,t} \\ v_{m,t} \end{bmatrix} = \frac{f}{z^{C}_{m,t}} \begin{bmatrix} x^{C}_{m,t} \\ y^{C}_{m,t} \end{bmatrix} \tag{4}$$

$R_t$ is a 3x3 rotation matrix and $T_t$ is a 3x1 translation vector. $X^{C}_{m,t} = [x^{C}_{m,t}, y^{C}_{m,t}, z^{C}_{m,t}]^T$ represents the model point $X_m$ with reference to the camera coordinate frame. Since the world and the camera center are the same initially in our case, $X^{O}_{m}$ is equivalent to $X^{C}_{m,1}$.

The parameters $R_t$ and $T_t$ compose of the pose sequence. The objective of the proposed pose tracking algorithm is to recover the camera motion, i.e. $R_t$ and $T_t$, at each time-step recursively given only the image measurements $p_{m,t}$

## III. ALGORITHM OVERVIEW

Figure 2 shows the overview of the proposed pose tracking algorithm. The KLT tracker described in [13] is used to extract feature points and track them in the images. It is assumed that the point features extracted by the tracker are contaminated only by Gaussian noise and are reliable enough for pose estimation.

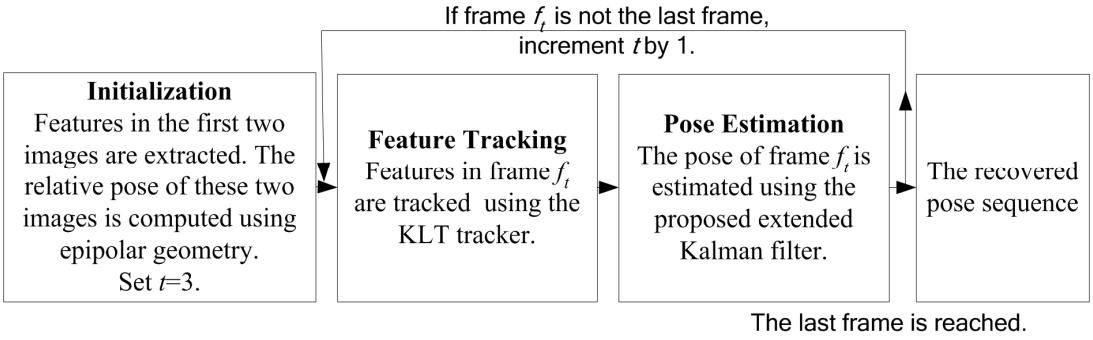


Figure 2. The flowchart of the proposed recursive algorithm.

To make the presentation clear, we denote the time-step of the algorithm by $s_t$ and frame index of the image sequence by $f_t$, where $t$ is an integer. The algorithm is initialized by estimating the relative pose of the

first two image frames ($f_1$, $f_2$) using epipolar geometry. Specifically, the fundamental matrix is first computed using the 8-point algorithm [15] plus a RANSAC robust estimator [39]. The pose parameters, $R_2$ and $T_2$, are then extracted from the fundamental matrix. This is actually an initial guess of the pose of image frame $f_2$ and will be refined later in each Kalman filtering cycle. The translation vector $T_2$ is recovered up to an unknown scale factor. As the RANSAC estimation procedure is used once in the initialization, it does not affect the speed of the algorithm too much.

Starting from frame $f_3$, the image measurements are processed by an extended Kalman filter (EKF). In each cycle, three images are input. Out of these input images, two of them are the images $f_1$ and $f_2$ in the sequence. They compose the base frames of the filter. The third input image is the image $f_t$ at the current time-step $t$. The EKF computes the pose of frame $f_t$ and, at the same time, refines the initial guess of the pose of frame $f_2$. The loop continues until all the images are utilized.

If the set of available point features is changing from frame to frame, the only additional procedure to handle the case is to find the set of features commonly appeared in all the three input images. These features are passed to the EKF as described previously. If the set of available features, say extracted from $f_1$ , $f_2$ and $f_t$, falls below a predefined minimum size, the algorithm restarts by resetting the time-step to $s_1$. The process followed is that frames $f_{t-j}$ and $f_{t-j+1}$ are used to re-initialize the algorithm and become the base frames. A new filtering loop is then started from frame $f_{t-j+2}$. Here $j$ is the number of overlapping frames between two segments of the sequence. $j$ is set to 1 in our implementation. With overlapping, the scale of the translation parameters of the previous segment and the next segment can be aligned. In practical applications like augmented reality, the difference in scale factor is not a concern as the output is the projection of 3D models. The algorithm stops when the end of the image sequence is reached.

## IV. The Extended Kalman Filter Implementation

At each Kalman filtering cycle, the EKF estimates the pose of the scene with respect to the current frame $f_t$ and, at the same time, refines the scene's pose with respect to frame $f_2$. A detailed review on EKF and its derivations can be found in [11]. For the clarity of the presentation, it is assumed that the point features are observable in the whole image sequence.

The state vector, denoted by $w_t$, consists of two groups of parameters and is written as:

$$w_t = [\ x_t\ \ \dot{x}_t\ \ y_t\ \ \dot{y}_t\ \ z_t\ \ \dot{z}_t\ \ \alpha_t\ \ \dot{\alpha}_t\ \ \ldots$$
$$\beta_t\ \ \dot{\beta}_t\ \ \gamma_t\ \ \dot{\gamma}_t\ \ x_2\ \ y_2\ \ z_2\ \ \alpha_2\ \ \beta_2\ \ \gamma_2\ \ ]^T$$

$x_t$, $y_t$ and $z_t$ are respectively the translation parameters of the scene along the $x$, $y$ and $z$ axis. $\dot{x}_t, \dot{y}_t, \dot{z}_t$ are

their corresponding velocities. $\alpha_t, \beta_t, \gamma_t$ are respectively the *Yaw*, *Pitch* and *Roll* angle with $\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t$ as their corresponding angular velocities. These 12 parameters represent the position, orientation, together with the motion of the camera at the current time-step. $x_2, y_2, z_2, \alpha_2, \beta_2, \gamma_2$ are the pose parameters of the scene with respect to frame $f_2$.

The importance of refining the pose at frame $f_2$ is that the initial guess computed from the fundamental matrix may not be exact even with the RANSAC robust estimation procedure. As we are dealing with the pose tracking problem, the camera motion is relatively small in two consecutive image frames. This results in insufficient depth information to make an accurate estimate only with two images.

The state transition and measurement equation of the filter are formulated as:

$$w_t = A w_{t-1} + \gamma'_t \tag{5}$$

$$\varepsilon'_t = g_t(w_t) + v'_t \tag{6}$$

$\gamma'_t$ and $v'_t$ are zero mean Gaussian noise. $\varepsilon'_t$ is an $n$x1 column vector representing the selected real measurements from the images. $g_t(w)$ is the $n$x1-output trifocal tensor point transfer function. Using the image measurements in the first two images and the pose parameters encoded in $w_t$, the estimated coordinates of the feature points at frame $f_t$ can be computed as.

$$\begin{aligned} g_t(w_t) &= \begin{bmatrix} u_{1,t} & v_{1,t} & \dots & u_{m,t} & v_{m,t} & \dots & u_{n,t} & v_{n,t} \end{bmatrix}^T \\ U''^k(m) &= U^i(m) l'_j(m) T_i^{jk} \end{aligned} \tag{7}$$

The above formulae are written in the tensor notation. $T_i^{jk}$ is known as the trifocal tensor, which encapsulates the geometric relations among three views. For details, please refer to [15]. $U(m)$ and $U''(m)$ are respectively the normalized homogenous 2D coordinates of the $m^{\text{th}}$ feature in frame $f_1$ and $f_t$ such that $U(m) = \begin{bmatrix} \bar{u}_{m,1} & \bar{v}_{m,1} & 1 \end{bmatrix}^T = \begin{bmatrix} u_{m,1}/f & v_{m,1}/f & 1 \end{bmatrix}^T$ and $U''(m) = \begin{bmatrix} \bar{u}_{m,t} & \bar{v}_{m,t} & 1 \end{bmatrix}^T = \begin{bmatrix} u_{m,t}/f & v_{m,t}/f & 1 \end{bmatrix}^T$. Normalization is taken since we have made the assumption that $M_t = I$ in the trifocal tensor $T_i^{jk}$. With that, $T_i^{jk}$ can be expressed more simply in tensor notation as:

$$T_i^{jk} = a'^j_i \, a''^k_4 - a'^j_4 \, a''^k_i \tag{8}$$

$a'^j_i$ and $a''^k_i$ are the elements of the 3x4 camera matrices $M_2$ and $M_t$ that project the 3D structure from the world coordinate frame onto images $f_2$ and $f_t$ respectively. The camera matrix can be transformed into the rotation matrix and the translation vector. The rotation matrix can be further converted to *Yaw*, *Pitch*, *Roll* angles. The required values can be obtained by decoding the state vector $w_t$.

$l'(m)$ is a line passing through the corresponding point in image $f_2$. This line must not be the epipolar

line and can be constructed as:

$$\begin{aligned} l'(m) &= \begin{bmatrix} l_{e2} & -l_{e1} & -\bar{u}_{m,2} l_{e2} + \bar{v}_{m,2} l_{e1} \end{bmatrix}^T \\ l'_e(m) &= \begin{bmatrix} l_{e1} & l_{e2} & l_{e3} \end{bmatrix}^T = e_{12} \times U'(m) \end{aligned} \tag{9}$$

$e_{12}$ is a rough estimate of the epipole in image $f_2$, which is calculated in the initialization step, and $U'(m) = \begin{bmatrix} \bar{u}_{m,2} & \bar{v}_{m,2} & 1 \end{bmatrix}^T = \begin{bmatrix} u_{m,2}/f & v_{m,2}/f & 1 \end{bmatrix}^T$. Actually, $l'(m)$ is perpendicular to the line joining $e_{12}$ and $U'(m)$, which is not necessarily the exact epipolar line through $U'(m)$. With this approximation, $e_{12}$ can be constant in the Kalman filtering cycle.

In equation (5), $A$ is a 18x18 block diagonal state transition matrix, which is defined as:

$$A = diag\left\{ \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}$$

The physical meaning of the above dynamic system is that the camera undergoes a constant velocity motion within a sampling period $T_s$ and the initial guess of the scene's pose with respect to frame $f_2$ is close to the actual values.

From the above dynamic system and measurement model, the four core Kalman filtering equations can be derived. The prediction equations for calculating the optimal estimates are:

$$\begin{aligned} \hat{w}_{t,t-1} &= A\hat{w}_{t-1,t-1} \\ P_{t,t-1} &= AP_{t-1,t-1}A^T + Q_t' \end{aligned} \tag{10}$$

The update equations for the corrections of estimates are:

$$\begin{aligned} \hat{w}_{t,t} &= \hat{w}_{t,t-1} + K(\varepsilon'_t - g_t(\hat{w}_{t,t-1})) \\ P_{t,t} &= P_{t,t-1} - K\nabla g_w P_{t,t-1} \end{aligned} \tag{11}$$

$$K = P_{t,t-1}\nabla g_w{}^T (\nabla g_w P_{t,t-1} \nabla g_w^T + R'_t)^{-1}$$

$\hat{w}_{t,t-1}$ and $\hat{w}_{t,t}$ are the estimates of state $w_t$ after prediction and update respectively. $P_{t,t-1}$ and $P_{t,t}$ are 18x18 matrices. They are respectively the covariances of $\hat{w}_{t,t-1}$ and $\hat{w}_{t,t}$. $R'_t$ is the covariance of the image noise $\nu'_t$. It is a tuning parameter and is set according to the quality of the images. It can also be acquired during the process of camera calibration. $Q_t'$ is the covariance of the noise terms $\gamma'_t$. $K$ is the 18x2$n$ Kalman gain matrix for the filter. $\nabla g_w$ is the Jacobian of the non-linear observation equation $g_t(w)$

evaluated at $\hat{w}_{t,t-1}$.

This is the complete EKF formulation of the pose tracking algorithm. For $\Gamma$ images in the sequence, the camera motion of the whole sequence can be recovered after $\Gamma - 2$ Kalman filtering cycles.

## V. AN EXTENSION TO STRUCTURE AND MOTION

With an optimal pose sequence recovered from the images, the scene structure can be computed accurately. Here we are going to extend our pose tracking algorithm to tackle the structure and motion problem recursively with the trifocal tensor. The flow of the proposed extension is outlined in figure 3.

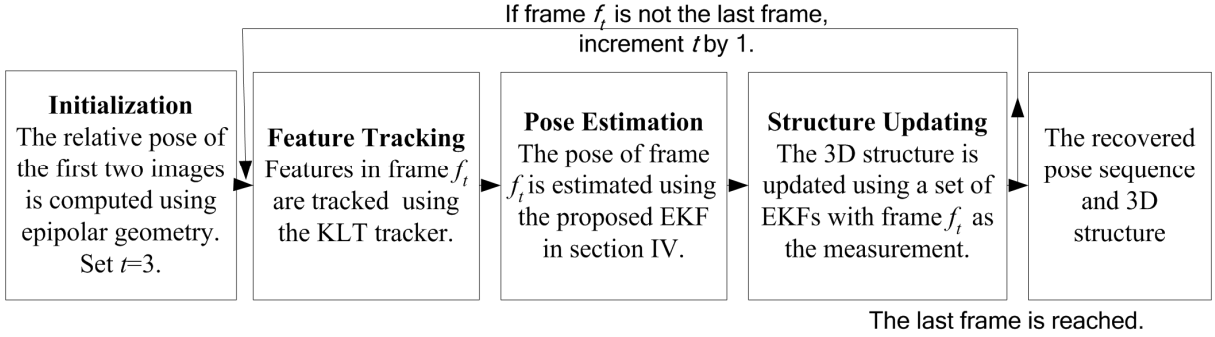


Figure 3. The flow of the recursive algorithm with the structure recovery extension.

The structure updating step is inserted into the main loop of the algorithm. Pose estimation and structure computation are interleaved. The pose tracking algorithm presented in section IV is employed to estimate the camera motion. A set of *N* extended Kalman filters (EKFs), each corresponds to a point in the scene structure, is used find out the 3D model.

Initially, the structure of the scene is a planar model located at a distance $z_{init}$ from the camera. $z_{init}$ is the average depth of the model calculated by triangulating the point features in the first two images with the pose recovered in the initialization step. In this way, both the translation parameters and the 3D model are in the same scale factor. The 3D coordinates are computed by back-projecting the corresponding features from the first image to the camera coordinate frame according to the orthographic projection equation:

$$\begin{bmatrix} u_{m,t} \\ v_{m,t} \end{bmatrix} = \frac{f}{z_{init}} \begin{bmatrix} x^C_{m,t} \\ y^C_{m,t} \end{bmatrix} \tag{12}$$

With the pose computed using frame $f_t$, the structure is updated by making use of frame $f_t$ as the measurements. The computation of structure is dependent on the pose acquired. However, the pose estimation step is still standalone from the calculation of the scene structure.

## VI. STRUCTURE ACQUISITION WITH EKFS

The structure updating step consists of *N* identical EKFs, each corresponds to one model point in the 3D space. The structure is assumed to be static. The dynamic model of a 3D point and its measurement equation are:

$$X'_{m,t} = X'_{m,t-1} + \gamma_t \tag{13}$$

$$\varepsilon_{m,t} = h_t(X'_m) + v_t \tag{14}$$

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

$\gamma_t$ and $v_t$ are the zero mean Gaussian noise. $\varepsilon_{m,t}$ is the real measurement from the image sequence. $h_t(X'_m)$ is the projection function, in which $X^C_{m,t}$ is obtained by substituting suitable values into equation (15) and (3). $X'_m$ is a scalar that represents a model point:

$$X^O_m = \begin{bmatrix} x^O_m \\ y^O_m \\ z^O_m \end{bmatrix} = \begin{bmatrix} u_{m,1} \\ v_{m,1} \\ 0 \end{bmatrix} + \frac{X'_m}{f}\begin{bmatrix} u_{m,1} \\ v_{m,1} \\ f \end{bmatrix} \tag{15}$$

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 [4]. This is valid for most of the modern high-resolution image capturing devices. Intuitively, the 3D coordinates of the points are expressed in terms of the first images that the features appear. This measure reduces the computation time required for the EKFs and at the same time maintains the rigidity of the scene structure. Also, the computation is over-determined at every frame when the number of features is larger than 7, resulting in a better convergence and stability. Detailed discussion about the advantages arising from this structure representation and the method to handle biased measurement can be found in [4].

With the dynamic model, the required equations can be derived. The prediction equations that provide an optimal estimate of the state at the next sample time are:

$$\begin{aligned} \hat{X}'_{m,t,t-1} &= \hat{X}'_{m,t-1,t-1} \\ \Lambda_{m,t,t-1} &= \Lambda_{m,t-1,t-1} + Q_t \end{aligned} \tag{16}$$

The update equations that improve the previous estimate using the measurements acquired are:

$$\begin{aligned} \hat{X}'_{m,t,t} &= \hat{X}'_{m,t,t-1} + W(\varepsilon_{m,t} - h_t(\hat{X}'_{m,t,t-1})) \\ \Lambda_{m,t,t} &= \Lambda_{m,t,t-1} + W\nabla h_X \Lambda_{m,t,t-1} \\ W &= \Lambda_{m,t,t-1}\nabla h_X^T(\nabla h_X \Lambda_{m,t,t-1}\nabla h_X^T + R_t)^{-1} \end{aligned} \tag{17}$$

$\hat{X}'_{m,t,t-1}$ is the estimate of $X'_m$ after prediction. $\Lambda_{m,t,t-1}$ is the variance of $\hat{X}'_{m,t,t-1}$ and $Q_t$ is the variance of the noise term $\gamma_t$, which is a scalar. The reason for incorporating a noise term having a considerable value into a static structure is that our initial structure is a planar model. Adjustment of the depth of the recovered model is necessary. $W$ is known as the 1x2 Kalman gain matrix of the filter. $R_t$ is a 2x2 measurement noise covariance matrix. $\nabla h_X$ is the Jacobian of the non-linear projection function $h_t(X')$ evaluated at $\hat{X}'_{m,t,t-1}$. The above formulation is enough to recover the 3D structure under the assumption that all point features are observable throughout the whole image sequence. The actual 3D structure is obtained by transforming $X'_m$ to the world coordinate frame using equation (15).

In real images, the set of observable point features is changing from frames to frames in the sequence. When a new point feature appears at frame $f_a$, the corresponding point in the 3D space is initialized by assuming its projection on that frame is orthographic. The initial position, expressed in camera coordinate frame, is computed according to equation (12). This is equivalent to setting parameter $X'_m$ equal to $z_{init}$ in our structure representation. The structure parameter for that point is now expressed in terms of its image coordinates in frame $f_a$. The relationship between the structure parameter $X'_m$ and world coordinate frame is:

$$X_m^O = R_a^{-1}\left\{\left(\begin{bmatrix} u_{m,a} \\ v_{m,a} \\ 0 \end{bmatrix} + \frac{X'_m}{f}\begin{bmatrix} u_{m,a} \\ v_{m,a} \\ f \end{bmatrix}\right) - T_a\right\} \tag{18}$$

A new EKF is set up to refine its position in its camera coordinate frame with parameter $X'_m$. The final position can be obtained by calculating its coordinates in the world frame using equation (18).

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

## VII. Advantages of the Proposed Algorithm

### A. *The use of dynamic system and trifocal constraint*

The proposed pose tracking algorithm makes use of two important constraints: 1) the dynamic system constraint in Kalman filtering and 2) the trifocal tensor constraint. The former constraint represents the physical movement of the camera. It has been widely used in visual tracking for robotic applications. The latter one expresses the geometric relations among the point correspondences in three views, which is actually a constraint on the imaging system. Previously, the trifocal tensor, which is analogous to epipolar geometry, has been applied to 3D structure recovery and guided searching of point correspondences. The

computation is usually offline and takes a relatively long period of time to complete. Our new formulation allows the use of trifocal tensor in a recursive manner. The major advantage of trifocal tensor is its independence from the scene structure. By making use of the point transfer function, our EKF is able to compute the pose directly without first estimating the 3D structure. Another advantage is that it is less susceptible to degeneracy than the essential constraint in epipolar geometry. It has little restriction on the camera motion to be estimated. Since the computation of the 3D model is no longer needed, the speed, accuracy and stability can be increased compared to SAM based method for pose tracking. This is the strength of merging trifocal tensor with Kalman filtering.

### *B. Speed of the algorithm*

The theoretical upper bounds on time and storage of the proposed pose tracking algorithm are $O(N^2)$ and $O(N)$ respectively in terms of the number of available point features $N$. The upper bounds are still the same even the tracking algorithm is extended to recover the scene structure. The time and space complexity of traditional SAM based recursive algorithms that have the same functionality and comparable accuracy are $O(N^3)$ and $O(N^2)$ [4] [5] respectively. The improvement on computation and storage requirement is due to the use of the trifocal tensor point transfer function in the measurement model.

Decoupled EKFs [8] [9] can achieve the same time and space complexity as our approach. Linear computation complexity can only be achieved if a good feature selection strategy is applied to choose a fixed number of feature points to the EKF for pose tracking. Indeed, the decoupling is regarded as a tradeoff between speed and accuracy. Our new pose tracking approach can improve the speed performance and at the same time keep the implementation as a full covariance EKF. This is contributed by the fact that pose estimation no longer depends on the 3D model structure. Structure updating while estimating the pose is not necessary and thus the complexity is reduced.

### *C. Accuracy and stability*

The proposed tracking algorithm has a higher accuracy than existing SAM based methods. Unlike our pose estimation method that is independent of the 3D structure, the approaches using full covariance EKFs [4] [5] compute both the pose and structure within a single filter. It means that the solution space of our approach is smaller than such methods by $N$ dimensions, where $N$ is the number of point features in the 3D structure. In other words, the probability of encountering the local optima is minimized. Also, the trifocal constraint on images, in addition to the dynamic system constraint on motion, is added to the EKF. Thus, both the accuracy and convergence of the filter is improved.

The decoupled EKFs [8] [9] that compute the pose and structure in an interleaved manner have the problem of retaining the relations between the camera motion and 3D structure. This type of recursive

algorithms should have a lower accuracy than traditional methods with full covariance EKFs and our new approach that have no dependency on the scene structure. This fact has been verified in the experiment in section VIII.

### *D. Handling of the changeable set of point features*

It is straightforward to handle the changing set of point features in our pose tracking algorithm. Since point features are regarded as measurements in our formulation, it is somewhat arbitrary to add or remove an input to the EKF, provided that the additional feature is not an outlier or the number of point features falls below a minimum after the removal. For SAM based methods, it is important to keep track of the corresponding feature in the 3D structure properly. Incorrect addition or removal of a point feature could be hazardous since the final pose sequence is highly dependent on the correctness of the recovered structure. Some full covariance filters, say the one in [3], require a sub-filter to construct the initial condition of the newly appeared point feature before it is added to the main filter. The proposed algorithm does not have such a problem.

## VIII. EXPERIMENTS AND RESULTS

### *A. Experiments with synthetic data*

The first experiment was conducted with synthetic data. A synthetic structure with 300 random feature points in 3D within a cube of volume of $0.13\text{m}^3$, centered at a place 0.33m away from the camera, was generated. The motion of the object 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 0.2 to 1.2 degrees per frame for the *Yaw*, *Pitch*, *Row* angle and 0.005 to 0.015 meters per frame for $t_x$, $t_y$ and $t_z$. The focal length of the camera was 6mm with a 2D zero mean Gaussian noise of 0.1 standard deviation. The length of each synthetic sequence is 99 frames. A total of 50 independent tests were carried out. The proposed algorithms, the EKF by Azarbayejani and Pentland [4] and the 2-step EKF by Yu et al [8] [9] were implemented in Matlab and run on a Pentium III 1GHz machine to estimate the camera motion. The results were compared and analyzed.

Figure 4 shows the average total rotation and translation errors of the three approaches under the 50 test cases. For the plots in figure 4 to 7, the line with asterisk (*), triangle (/\) and circle (O) markers are for our proposed approach, the EKF by Azarbayejani and Pentland [4] and the 2-step EKF by Yu et al [8] [9] respectively. Here, the total rotation was calculated using the axis-angle representation. The difference between the actual and the recovered angle is the error. The total translation error was computed by

subtracting the recovered translation vector from the actual one and the magnitude was taken. From the plots, the proposed approach has a lower total rotation and translation errors than the other methods under comparison. Figure 5 shows the errors in terms of percentages. Our method achieves an average of 0.85% and 3.80% for rotation and translation respectively. To have a more detailed picture, the errors of each individual pose parameters were plotted. From figure 6, it can be seen that there is an obvious improvement on the accuracy of the *Yaw*, *Pitch* and *Roll* angle. For the each translation parameter, our algorithm has a comparable accuracy with the 2-step EKF but is much better than the EKF by Azarbayejani and Pentland. Table I shows the average errors of each pose parameter per frame of the three algorithms. Our approach has the lowest error for all the parameters except the translation $t_z$, which is slightly (less than 15%) higher than that of the 2-step EKF. It is encouraging that our new formulation has reduced the errors by at least 36% and 90% for the total translation and rotation respectively. The improvement on the accuracy is due to the use of trifocal tensor. A detailed explanation has been described in section VII.

It is expected that the 2-step EKF by Yu et al should have a lower accuracy than the EKF by Azarbayejani and Pentland as the filter has been decoupled. On the contrary, it is found that the former approach performed better from the experimental results. Actually, both algorithms diverged in some test cases. It seems that the EKF by Azarbayejani and Pentland diverged to a greater extent, leading to a higher average error compared to the 2-step EKF.

TABLE I
THE AVERAGE ERROR OF EACH POSE PARAMETER

| | Roll | Pitch | Yaw | $t_x$ | $t_y$ | $t_z$ | Total rotation (% error) | Total translation (% error) |
|---|---|---|---|---|---|---|---|---|
| Our approach | 0.0822 | 0.1769 | 0.5411 | 0.0067 | 0.0199 | 0.0210 | 0.2417 (0.8466%) | 0.0306 (3.8015%) |
| Azarbayejani's EKF | 0.8296 | 2.1704 | 7.9952 | 0.0314 | 0.0673 | 0.0336 | 3.2283 (11.830%) | 0.0871 (15.509%) |
| Yu's EKF | 1.0856 | 2.5604 | 4.8386 | 0.0131 | 0.0406 | 0.0183 | 2.6504 (9.7289%) | 0.0485 (8.1139%) |

A table showing the average errors of each pose parameter per frame of the 3 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 translation errors (i.e. the total translation, $t_x$, $t_y$ and $t_z$ error) are in meters. The values in brackets are the percentage errors.

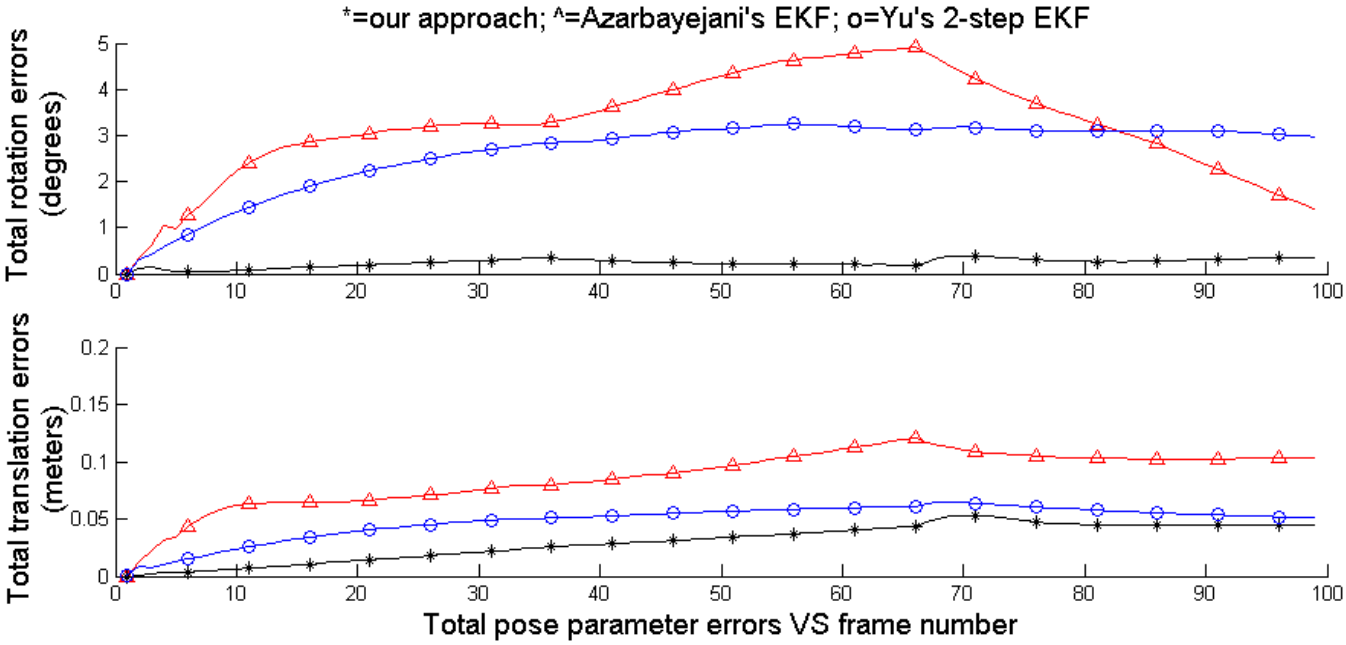


Figure 4. The average total rotation error (top, in degrees) and total translation error (bottom, in meters) versus frame number of the 3 algorithms.

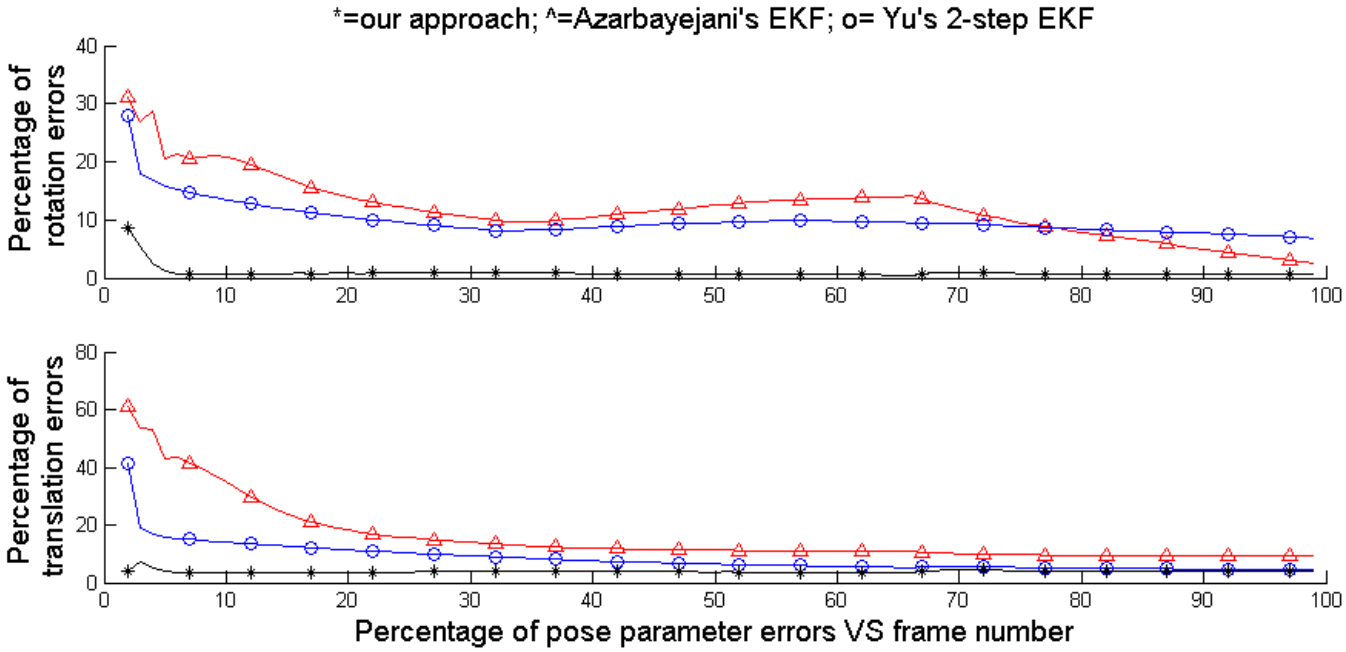


Figure 5. The average percentage of rotation error (top) and translation error (bottom) versus frame number of the 3 algorithms.

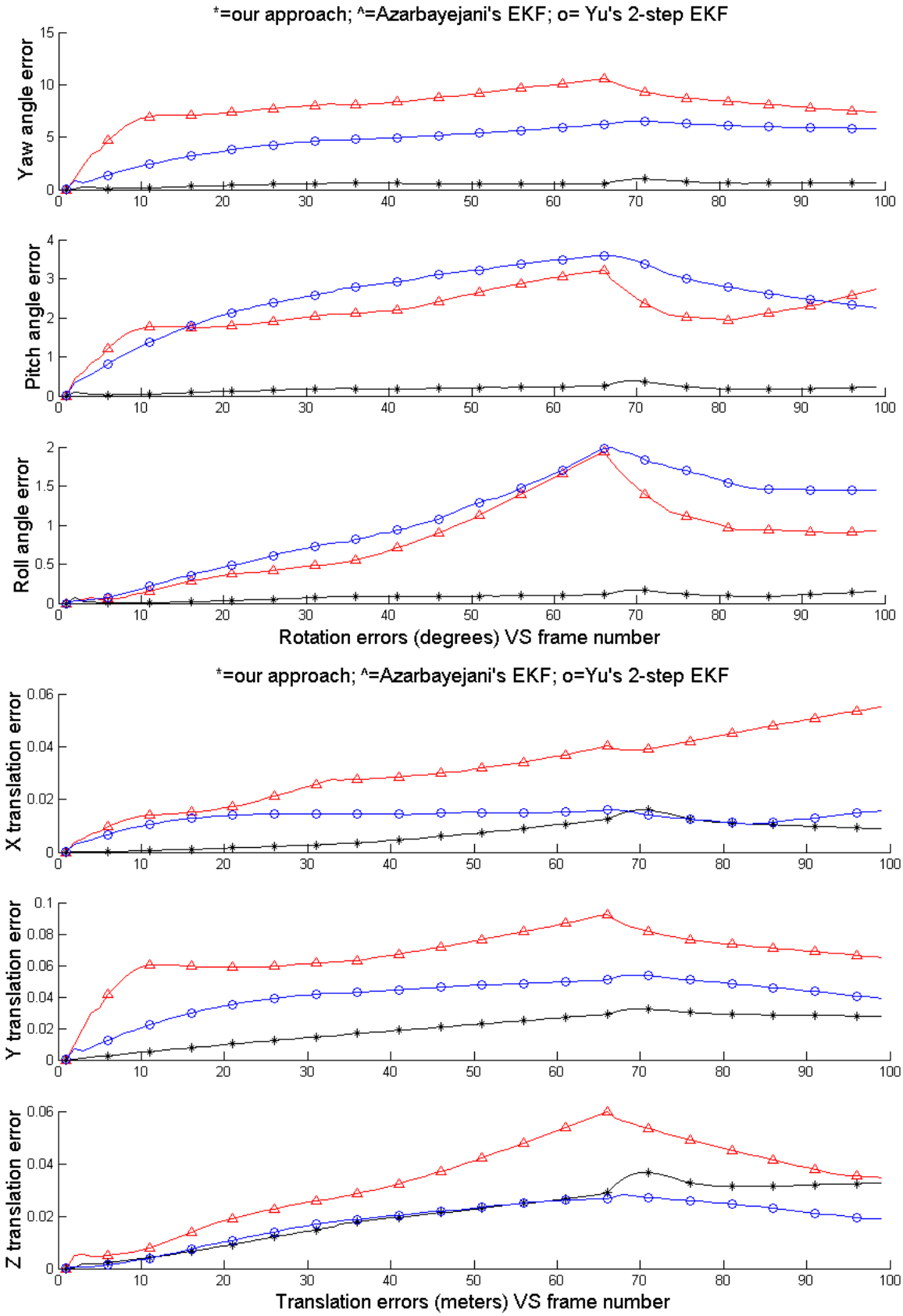


Figure 6. The average error of each recovered pose parameter versus frame number of the 3 algorithms.

The extension of our pose tracking approach has been applied to recover the synthetic structures. Figure 7 shows the image residual errors against CPU time, which was calculated by re-projecting the recovered 3D structure onto the actual image with the computed pose sequence. The proposed approach has an average final error about 11 pixels per feature over 99 frames. Since a zero mean pixel noise of 0.1 standard deviation has been added to the images, it is expected that the root mean square error of a feature in the re-projected structure is 13.86 pixels over 99 synthetic images. This means our approach just overfits a little compared to the 2-step EKF, which has an error of 4 pixels. The residual error of the EKF by Azarbayejani and Pentland is 25 pixels. Such an improvement is contributed by the fact that the estimation of pose in our algorithm is independent of the scene structure. The search space is reduced and the chance of locking into a local optimum is minimized. In addition, the results reveal that the problem of structure and pose ambiguities, which has been reported in [12], is alleviated with the incorporation of the trifocal constraint into the recursive algorithm.

Table II shows the time needed to recover the camera motion when new image measurements were sequentially fed to the algorithms. The first step in creating this plot was to initialize the algorithm using the first 10 frames so that it can converge to a steady state. Then the succeeding 89 frames were sequentially fed to the algorithm and the required computation time is measured. The computation time of the algorithms remains roughly at a constant level over time. On average, our algorithm takes 1.56 seconds to compute the pose of the scene for each image. Its extension to structure recovery takes an addition of 0.11 seconds, i.e. a total of 1.67 seconds, to recover both the pose and scene structure for 300 point features. Both of them outperformed the full covariance EKF by Azarbayejani and Pentland, which needs 2.60 seconds to achieve the same task. However, the 2-step EKF takes only 0.42 seconds to process an extra image frame. The reason is that their EKF is decoupled, which is actually a tradeoff between speed and accuracy. Between the two full covariance EKFs under comparison, it can be concluded that the proposed approach performed better in terms of speed and accuracy.

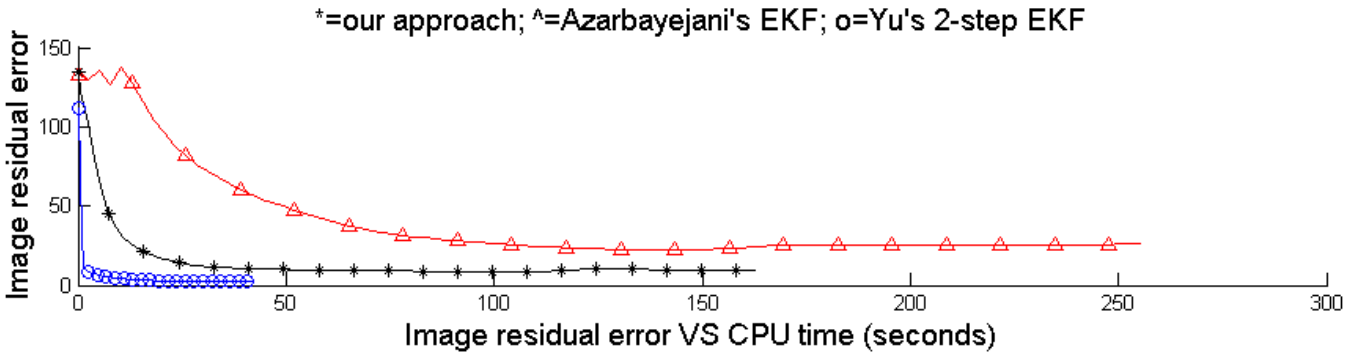


Figure 7. The relationship between the CPU time and the image residual error. Note that the algorithms were implemented in Matlab with a Pentium III 1GHz machine and the time measurement is in seconds.

TABLE II
TIME REQUIRED TO PROCESS AN EXTRA IMAGE MEASUREMENT

| | Our approach | Our extended approach | Azarbayejani's EKF | Yu's 2-step EKF |
|---|---|---|---|---|
| Time required (in seconds) | 1.56 | 1.67 | 2.60 | 0.42 |

A table showing the average CPU time for the 4 algorithms to recover the pose and 3D structure when extra frames were added to the image sequence.

### B. *Experiments with real images*

An experiment using real images was also performed. Three image sequences were used to test the proposed approach. The first 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 is 100 frames. The second sequence was recorded from a live TV programme. The stadium in the images is located in Athens and was holding the closing ceremony of the Olympic games 2004. The resolution of the images is 352 X 255. It lasts for 9 seconds and consists of 90 frames. The third one was captured from a DVD. The Grand Canyon in the northwestern Arizona was viewed from a helicopter. It is 5-second long and contains 50 images. Please refer to the attachment or visit the URL http://www.cse.cuhk.edu.hk/~vision/demo/tensorkalman/ to see the original sequences. The proposed algorithm was applied to track the camera motion. The recovered pose sequences were used to produce augmented reality videos

Figure 8 and 9 show the results from the laboratory scene sequence. An augmented reality video was made successfully. A synthetic car, which is drawn by wire-frames, was placed in front of the yellow box. The motion of the car is consistent with the background scene. The plots in figure 9 illustrate the pose parameters acquired from the image sequence. It is reasonable that the recovered *Yaw*, *Pitch* and *Roll* angle are smaller than 1 degree, since the camera was translating horizontally with the viewing angle fixed.

Figure 8. Results of inserting an artificial object into the laboratory scene sequence using the proposed approach. The left column: The 1st, 50th and 100th image of the sequence. The right column: A synthetic car, which is drawn by wire-frames, was augmented into the scene.

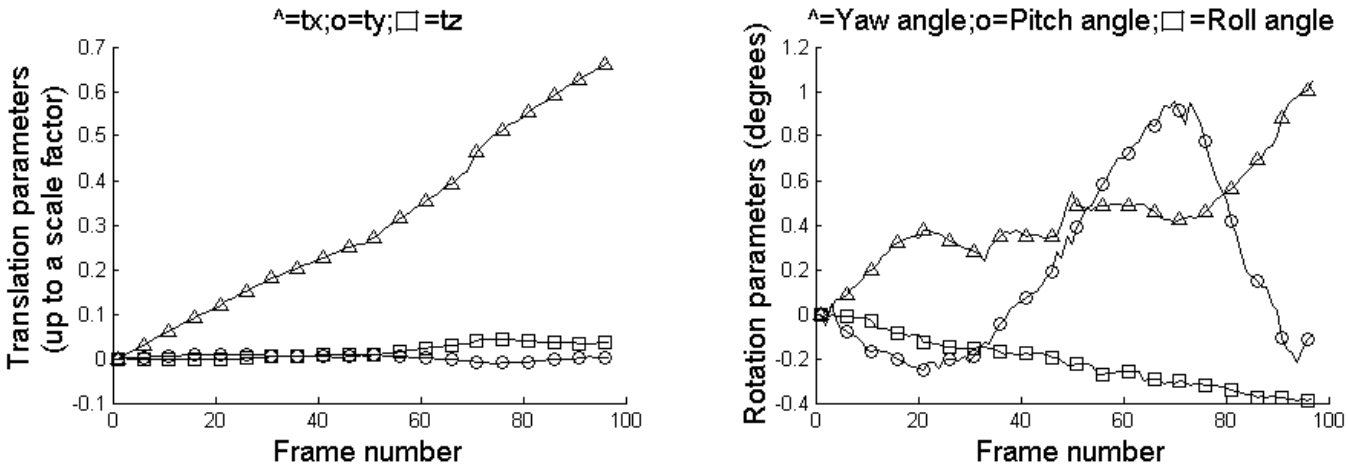


Figure 9. The pose sequence recovered from the laboratory sequence. The line with triangle (/\), circle (O) and square (□) markers on the left plot are for the translation parameters $t_x$, $t_y$ and $t_z$ respectively while the line with triangle (/\), circle (O) and square (□) markers on the right plot are respectively for the *Yaw*, *Pitch* and *Roll* angle.

The results from the Olympic stadium sequence are in figure 10 and 11. This time the synthetic car was put at the center of the stadium. The quality of the augmented reality sequence is quite good even the image resolution is not high. As the video was taken on a helicopter hovering on the stadium, the roof of the stadium and the top of the car can be seen. For the same reason, the motion of the scene mainly consists of rotation on the z-axis. It has been reflected in the recovered rotation parameters, which is shown in figure 11.

Figure 12 and 13 are the results from the Grand Canyon sequence. The camera motion is more arbitrary and the depth of the scene is larger than the previous sequences. As the camera is getting around the canyon, the major motion is translation along the x-axis and rotation on the pitch angle, which can be verified from figure 13. By inspecting the resulting augmented reality sequence, the synthetic object, which was placed on the slope, is synchronized with the movement of the background. The results are accurate and visually acceptable.

The last two examples best demonstrate the advantages of SAM-based pose tracking algorithms over those marker-based approaches. Only the former methods are applicable to these sequences since the they were taken by the third parties and neither the scene structure is known nor markers can be placed.

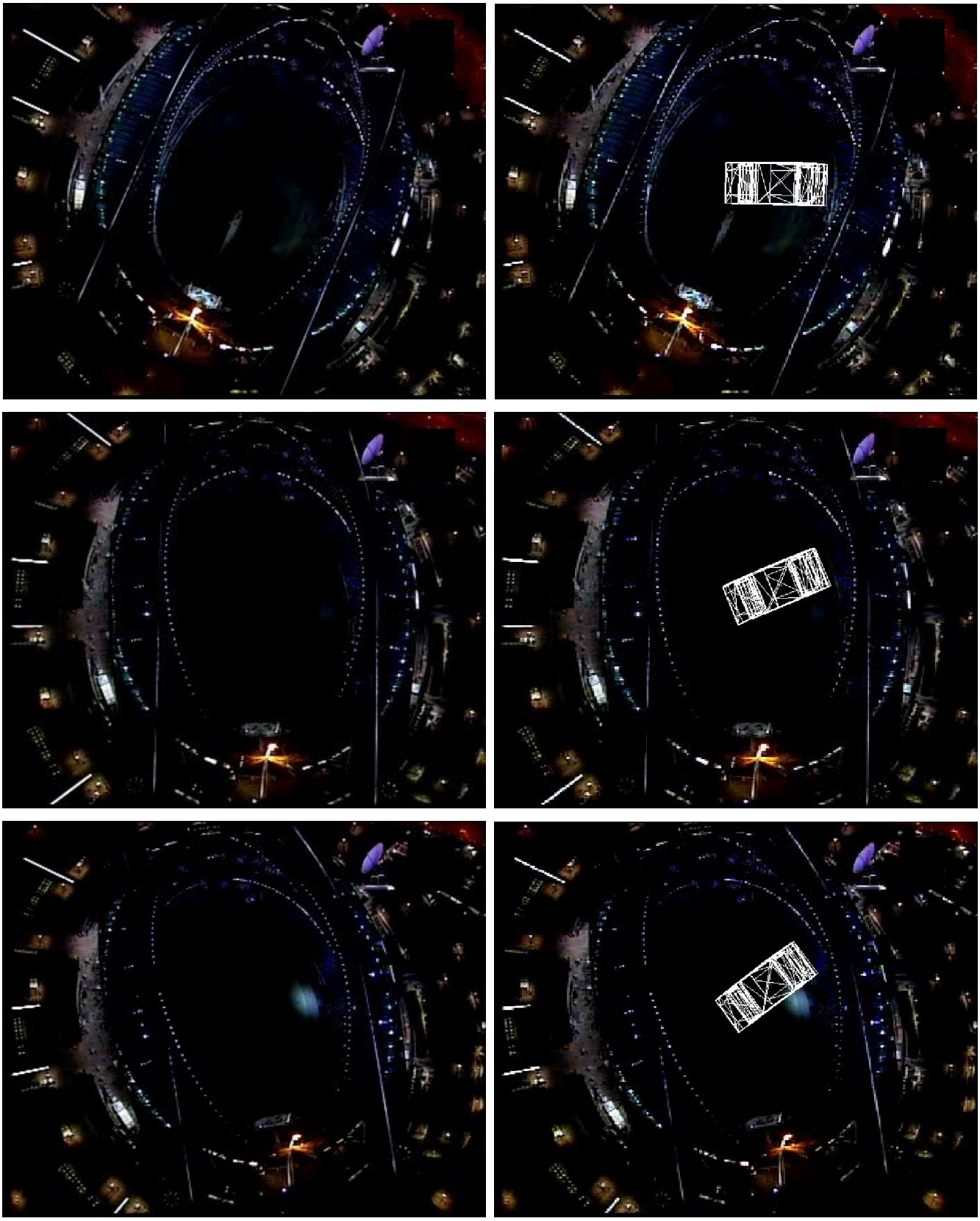

Figure 10. Results of inserting an artificial object into the Olympic stadium sequence. The sequence was recorded from a live broadcast of the closing ceremony of the Athens Olympic games. The left column: The 1st, 45th and 90th image of the sequence. The right column: A synthetic car, which is drawn by wire-frames, was placed at the center of the stadium.

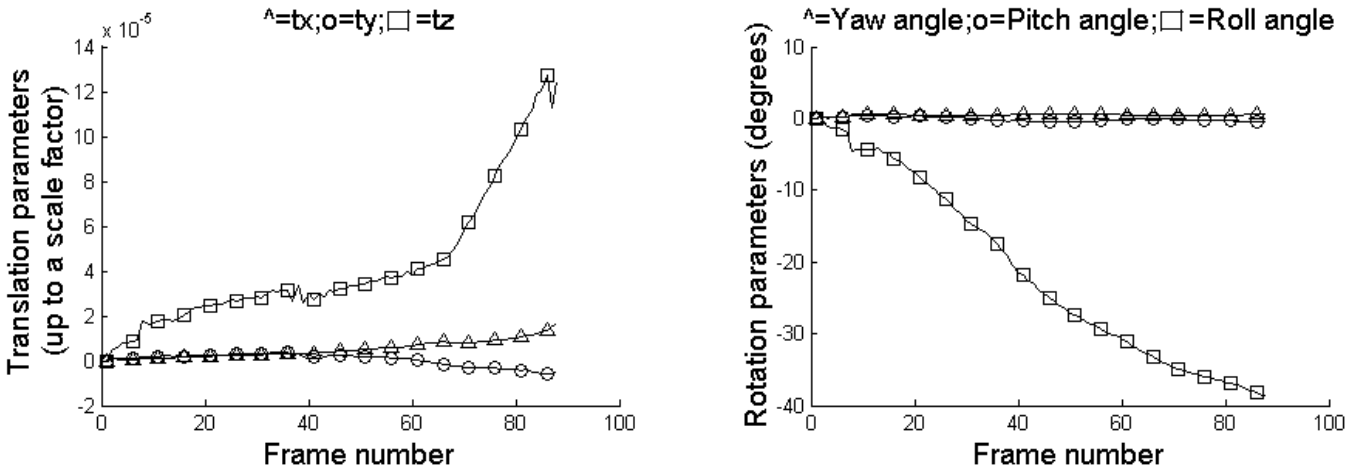


Figure 11. The pose sequence recovered from the Olympic stadium sequence. The line with triangle (/\), circle (O) and square (□ ) markers on the left plot are for the translation parameters $t_x$, $t_y$ and $t_z$ respectively while the line with triangle (/\), circle (O) and square (□) markers on the right plot are respectively for the *Yaw*, *Pitch* and *Roll* angle.

Figure12. Results of inserting an artificial object into the Grand Canyon sequence. The sequence was captured from a DVD. The left column: The 1st, 25th and 50th image of the sequence. The right column: A synthetic car, which is drawn by wire-frames, was placed on the slope.

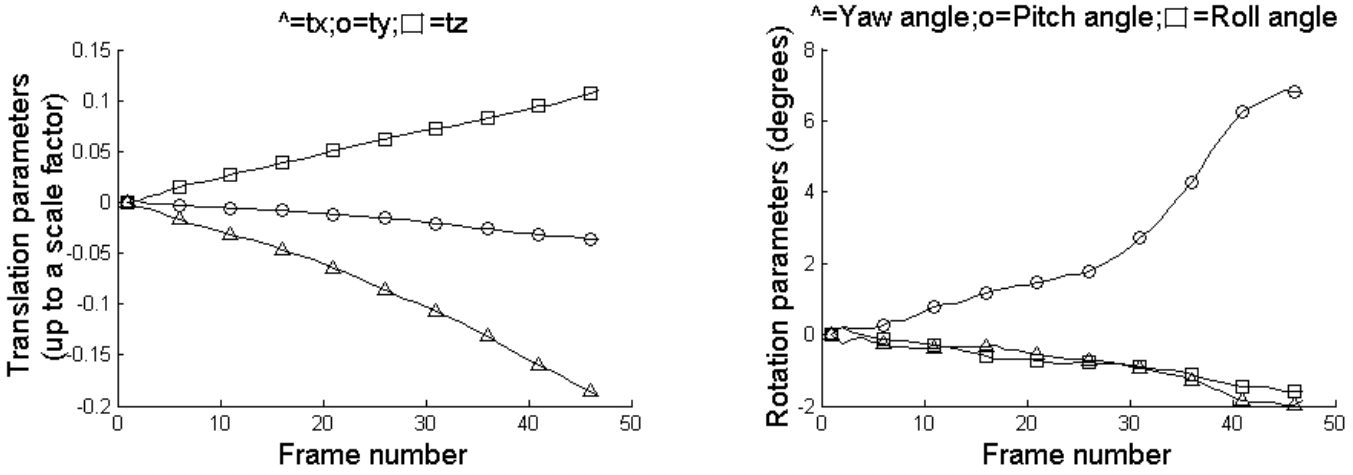


Figure 13. The pose sequence recovered from the Grand Canyon sequence. The line with triangle (Λ), circle (O) and square (□) markers on the left plot are for the translation parameters $t_x$, $t_y$ and $t_z$ respectively while the line with triangle (Λ), circle (O) and square (□) markers on the right plot are respectively for the *Yaw*, *Pitch* and *Roll* angle.

## IX. CONCLUSION

A high-speed recursive pose tracking algorithm for augmented reality has been proposed in this paper. By merging the power of Kalman filtering and trifocal tensor, a significance improvement on the accuracy and computation efficiency of the algorithm has been achieved. The estimation of pose sequence is now independent of the scene structure with the trifocal tensor point transfer function in the measurement model. This also reduces the search space of the algorithm, resulting in a more accurate solution of the pose, an increase in speed and a simpler procedure to handle the changeable set of point features. Our new pose tracking approach has been extended to simultaneous recovery of structure and motion. With the proposed algorithm, the traditional problem of structure and pose ambiguities has been alleviated. Experimental results show that our approach outperformed other Kalman filter based methods with little overfitting problem. It is found that the average rotation error is less than one-tenth of the existing algorithms under comparison. The advantages of our algorithm have been demonstrated by applying it to produce an augmented reality video sequence.

To pursue further, an efficient feature selection strategy could be added to find out a set of reliable point features for the proposed pose tracking algorithm in each time-step. It has been shown that it is effective to incorporate such a scheme into a model based pose tracking algorithm in visual servoring applications [29]. With feature selection, the algorithm can rely on a smaller number of point features in the images. As the time complexity can be independent of the number of available point features, the algorithm can be speeded up, provided that the computation overhead of the selection procedure is small. Also, the accuracy can be further enhanced as the outliers in the images are removed. It is believed that our formulation can be

realized in real-time at 30Hz with suitable implementation.

From a theoretical aspect, a more sophisticated dynamic model that involves the use of twist representation and Lie groups [40] can be applied to estimate the camera motion. With that, the singularities due to the use of the *Yaw*, *Pitch*, *Roll* angles to represent rotations can be avoided. The observation model can be further linearized as the trigonometric functions are removed with the application of twist. In addition, the acceleration component can also be incorporated into the dynamic system elegantly so that the motion of the camera can be modeled more realistically. The Lie groups and their algebras are useful for transforming motion parameters from one coordinate frame into another. They can help upgrading our system from single camera to stereo. We are now working towards this direction.

## X. ACKNOWLEDGMENT

The work described in this paper was supported by a grant from the Research Grant Council of Hong Kong Special Administrative Region. (Project Number: 4204/04E)

## 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.G.Harris and J.M.Pike. “3D positional integration from image sequence”, Image and Vision Computing, vol. 6, no. 2, 1988.

[3] A.Chiuso, P.Favaro, H.Jin and S.Soatto, “Structure from motion casually integrated over time”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, 2002.

[4] 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, June 1995.

[5] 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, July 1990.

[6] 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, July 2002.

[7] J.Weng, N.Ahuja and T.S.Huang, “Optimal motion and structure estimation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 9, September 1993.

[8] Y.K.Yu, K.H.Wong and M.M.Y.Chang, “Recursive 3D model reconstruction based on Kalman filtering”, IEEE Transactions on Systems, Man and Cybernetics- Part B. (to appear)

[9] 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 2004, Cambridge, August 2004.

[10] Y.K.Yu, K.H.Wong and M.Y.Y.Chang, "A fast and robust simultaneous pose tracking and structure recovery algorithm for augmented reality applications", in proc. of the IEEE International Conference on Image Processing 2004, Singapore, October 2004.

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

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

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

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

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

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

[17] M.Pollefeys, "Tutorial on 3D modeling from images", in conjunction with ECCV, June 2000.

[18] P.Sturm, "Mixing catdioptric and perspective cameras", in proc. of Workshop on Omni-directional Vision, Copenhagen, Denmark, 2002.

[19] 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.

[20] J.Costeira and T.Kanade, "A multibody factorization method for independently moving objects", International Journal of Computer Vision, vol. 29, no. 3, pp. 159-179, 1998.

[21] 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.

[22] M.M.Y.Chang and K.H.Wong, "Model and pose acquisition using extended Lowe's method", IEEE Transactions on Multimedia. (to appear)

[23] X.S.Gao, X.R.Hou, J.Tang and H.F.Cheng, "Complete solution classification for the perspective-three-point problem", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, August 2003.

[24] R.Horaud, B.Conio and O. Leboulleux, "An analytic solution for the perspective 4-point problem", Computer Vision, Graphics and Image Processing, vol. 47, pp. 33-44, 1989.

[25] M.L.Liu and K.H.Wong, "Pose estimation using four corresponding points", Pattern Recognition Letters, vol. 20, no. 1, pp. 69-74, January 1999.

[26] D.G.Lowe, "Fitting parameterized three-dimensional models to images", IEEE Pattern Analysis and Machine Intelligence, vol. 13 no. 5, pp. 441 -450, May 1991.

[27] S.Hati, S.Sengupta, "Robust camera parameter estimation using genetic algorithm", Pattern Recognition Letters, vol. 22, pp. 289-298, 2001.

[28] F.Toyama, K.Shoji and J.Miyamochi, "Model-based pose estimation using genetic algorithm", in proc. of the International Conference on Pattern Recognition, pp. 198-201, 1998.

[29] V.Lippiello, B.Siciliano and L.Villani, "Objects motion estimation via BSP tree modeling and Kalman filtering of stereo images", in proc. of the IEEE International Conference on Robotics and Automation, pp. 2968-2973, Washington DC, 2002.

[30] 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.

[31] X.Zhang, S.Fronz and N.Navab, "Visual marker detection and decoding in AR systems: A comparative study", in proc. of the IEEE International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, September 2002.

[32] D.Reiners, D.Stricker, G.Klinker, and S.Muller, "Augmented reality for construction tasks: Doorlock assembly", in proc. of the IEEE International Workshop on Augmented Reality, 1998.

[33] H.Kato and M.Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system", in proc. of the IEEE International Workshop on Augmented Reality, pp. 125-133, 1999.

[34] X.Zhang, N.Navab, and S.Liou, "E-commerce direct marketing using augmented reality", in proc. of the IEEE International Conference on Multimedia & Expo., 2000.

[35] S.Princel, A.D.Cheok, F.Farbiz and T.Williamson, "3D live: Real time captured content for mixed reality", in proc. of the IEEE International Symposium on Mixed and Augmented Reality, pp. 7-13, Darmstadt, Germany, 2002.

[36] 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 the IEEE the International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, September 2002.

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

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

[39] M.A.Fischler and R.C.Bolles, "Random sample concensus: A paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, vol. 24, no. 6, pp. 882-887, June 1981.

[40] T.Drummond, R.Cipolla, "Real-time visual tracking of complex structures", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.

[41] Y.K.Yu, K.H.Wong and M.M.Y.Chang, "Pose estimation for augmented reality applications using genetic algorithms", IEEE Transactions on Systems, Man and Cybernetics- Part B. (accepted for publication, January 2005)

[42] S.Avidan and A.Shashua, "Threading fundamental matrices", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 1, January 2001.

[43] S.Soatto, R.Frezza and P.Perona, "Motion estimation on the essential manifold", in proc. of the European Conference on Computer Vision 1998, Freiburg, Germany, 1998.

[44] Z.Zhang and Y.Shan, "Incremental motion estimation through modified bundle adjustment", in proc. of the IEEE International Conference on Image Processing 2003, Barcelona, September 2003.

Mr. 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. He is now a PhD student in the Department of Computer Science and Engineering in the same university. He has been 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. His contact address is: The Computer Science and Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong. Email: ykyu@cse.cuhk.edu.hk

Prof. 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 3D computer vision, virtual reality image processing, pattern recognition, microcomputer applications and computer music. His contact address is: The Computer Science and Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong. Email: khwong@cse.cuhk.edu.hk

Prof. 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. His contact address is: The Information Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong. Email: mchang@ie.cuhk.edu.hk

Dr. Siu Hang Or joined the Chinese University of Hong Kong in 1991 as an instructor and he is now the project leader of the Computer Game Technology Center in the Computer Science and Engineering Department of the University. He received the B.Sc. in electronic engineering of the Chinese University of Hong Kong in 1988. He obtained his M.Phil. and Ph.D. in Computer Science from the same university in 1991 and 1998 respectively. Besides teaching, he also works actively to incorporate advanced technology in computer game development. His research interests include structure/motion estimation in computer vision and rendering techniques in computer graphics. His contact address is: The Computer Science and Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong. Email: shor@cse.cuhk.edu.hk