COMPUTING POSE SEQUENCES DIRECTLY FROM VIDEOS

Ying Kin Yu¹, Kin Hong Wong¹, Michael Ming Yuen Chang² and Siu Hang Or¹

¹ Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.
² Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong.
Email: {ykyu, khwong, shor}@cse.cuhk.edu.hk, mchang@ie.cuhk.edu.hk

ABSTRACT

An innovative extended Kalman filter (EKF) algorithm for pose tracking has been proposed in this paper. It has the advantages of both structure and motion-based (SAM-based) and traditional model-based pose estimation algorithms. With no prior information about the scene, the pose sequence can be computed directly from images while the updating of the 3-D structure is not necessary. To achieve the goal, a constant velocity motion model is used as the dynamic system and the trifocal tensor point transfer function is applied to the measurement model of the filter. The resulting algorithm is stable, accurate and efficient. An empirical comparison with existing EKFs which deal with the same problem has been made and shows that our approach outperformed the others. The proposed method has been tested with various video sequences to demonstrate its performance in real situations.

Index Terms— Image motion analysis, machine vision, Kalman filtering.

1. 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 3-D structure, since a high precision depth map can be constructed with an optimal pose sequence.

Traditional pose acquisition methods require known 3-D structure of the scene [15] [16]. More general approaches, which require no prior information on the 3-D model, are based on the techniques in structure and motion (SAM) such as multiple view geometry [10] [11], factorization [1] and bundle adjustment [3] [13]. To deal with the problem recursively, Kalman filtering can be applied [4] [5] [6] [7] [8] [9] [17] [18]. The work in [6], which uses an iterated extended Kalman filter to achieve the task, is the seminal work. Azarbayejani and Pentland [5] proposed an improvement of [6] by making an extension in recovering the camera focal length and the representation of the 3-D model. In [8], Yu et. al. decoupled the full covariance extended Kalman filter (EKF) such that the computation efficiency is increased as a tradeoff in accuracy. An extension of their work can be found in [9], in which the Interacting Multiple Model (IMM) was added to the original formulation. Soatto et. al. [12] applied the essential constraint to pose estimation but this constraint is susceptible to degeneracy in some real situations [10].

This paper describes an innovative EKF-based algorithm that tackles the pose tracking problem. With no prior information about the 3-D structure of the scene, the pose information can be recovered from a monocular image sequence directly and recursively with the trifocal tensor. The major contribution of our approach is the incorporation of the trifocal tensor into the Kalman filtering formulation. In the algorithm, the trifocal tensor point transfer function is used in the measurement model of the EKF. That is apart from the dynamic system constraint on the motion of the camera, the trifocal constraint has also been employed in the filtering cycle, resulting in an enhancement on the accuracy of the algorithm.

In our formulation, the recovery of pose sequence is independent of the 3-D structure. Traditional recursive SAM-based algorithms require the update of both the 3-D model points and pose parameters either simultaneously [5] or in an interleaved manner [8]. As no computation of the 3-D model coordinates are involved in our approach, the total number of parameters required to be estimated is reduced from N+6, where N is number of available point features, to 12. In this way, both the stability and computation efficiency of our filter can be improved.

An empirical comparison with existing recursive SAM-based pose tracking approaches [5] [8] has been made. Experimental results show that our algorithm had a better overall performance than the others. In addition, the proposed approach has been tested using real video sequences. Its accuracy has been verified by re-projecting the corner features in the 1st image to the succeeding frames with the recovered pose.

2. PROBLEM MODELING

The relationship between a point \( X_m \) of the 3-D structure in the world coordinate frame and its 2-D projection \( p_{n+1} \) on the image plane can be related as:
3. SUMMARY OF THE ALGORITHM

Fig. 1 shows the overview of the proposed pose tracking algorithm. The Kanade-Lucas-Tomasi (KLT) tracker described in [2] 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.

\[
X_{n,t}^c = R_{n,t}X_{n,t}^o + T_{t},
\]

(1)

\[
P_{n,t} = \begin{bmatrix} u_{n,t} \\ v_{n,t} \end{bmatrix} = f^T \begin{bmatrix} X_{n,t}^c \\ z_{n,t} \end{bmatrix}
\]

(2)

where \(X_{n,t}^c = [x_{n,t}^c, y_{n,t}^c, z_{n,t}^c]^T\) denotes the coordinates of \(X_{n,t}^o\) in the camera coordinate frame and the subscript \(t\) denotes the time when the measurement is being made. \(f^T\) represents the focal length of the camera. \(T_{t} = [x, y, z]^T\) is a 3 \(\times\) 1 translation vector and \(R_{t}\) is a 3 \(\times\) 3 rotation matrix parameterized by the Yaw \((\alpha)\), Pitch \((\beta)\) and Roll \((\gamma)\) angle. The objective of the proposed algorithm is to recover the pose information \(R_{t}\) and \(T_{t}\) at each time-step recursively given only the image measurements \(P_{n,t}\).

The implementation details of the EKF and the construction of trifocal tensor point transfer function. Equation (7) is written in the tensor notation, in which \(\Gamma\) is known as the trifocal tensor [10]. \(\Gamma\) encapsulates the geometric relations among three views and contains the pose parameters \(R_{f}\) and \(T_{f}\), which is up to an unknown scale factor, are then extracted from the fundamental matrix [10]. This is actually an initial guess of the pose of image \(f_2\).

Starting from image \(f_2\), the measurements are processed by an EKF. In each cycle, three images are processed, within which two of them are images \(f_1\) and \(f_2\) in the sequence. They compose of the base frames of the filter. The third one is the image \(f_3\) at the current time-step \(t\). The EKF computes the pose of image \(f_3\) and, at the same time, refines the initial guess of the pose of image \(f_2\). Its state vector, denoted by \(w_t\), is written as:

\[
w_t = [x_t, y_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t, \gamma_t, \dot{\gamma}_t, \alpha_2, \dot{\alpha}_2, \beta_2, \dot{\beta}_2, \gamma_2, \dot{\gamma}_2]^T
\]

(3)

\(x_t, y_t, z_t\) are the translational velocities corresponding to \(x_t, y_t, z_t\), while \(\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t\) are the angular velocities corresponding to \(\alpha_t, \beta_t, \gamma_t\) . The state transition and measurement equation of the filter are formulated as:

\[
w_t = A_{t-1}w_{t-1} + \epsilon_{t},
\]

(4)

\[
\epsilon_{t} = g_t(w_t) + v_t,
\]

(5)

where

\[
g_t(w_t) = \begin{bmatrix} u_{n,t} \\ v_{n,t} \\ \ldots \\ u_{n,t} \\ v_{n,t} \\ \ldots \\ u_{n,t} \\ v_{n,t} \end{bmatrix}
\]

and

\[
A = \text{diag}\left[ \begin{bmatrix} 1 & S & 1 & S & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & S & 1 & S & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & S & 1 & S & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \right]
\]

\(\epsilon_{t}\) and \(v_{t}\) are zero mean Gaussian noise. The dynamic system models the camera as constant velocity motion with the initial guess of the pose \(R_{f}\) and \(T_{f}\) close to the actual values. \(\epsilon_{t}\) is an \(N\times 1\) column vector representing the measurements from images, where \(N\) denotes the number of point features input to the filter. \(g_t(w_t)\) is the \(N\times 1\)-output trifocal tensor point transfer function. After the fusion of pose information directly from 2-D images possible. \(U_{n,t}\) is the normalized homogenous form of \(P_{n,t}\) such that

\[
U_{n,t} = \begin{bmatrix} \pi_{n,t} \\ \pi_{n,t} \\ \pi_{n,t} \end{bmatrix} = \frac{[u_{n,t}, v_{n,t}, f]}{[u_{n,t}, v_{n,t}, f]} \cdot \pi_{n,t}\]

\(l_{n,t}\) is a line passing through the point \(p_{n,t}\) in image \(f_t\). \(S\) represents the sample period of the image measurements. The implementation details of the EKF and the construction of tensor \(\Gamma\) plus line \(l_{n,t}\) can be found in [14] and [10], respectively. The filtering loop ends when all the images are processed.

If the set of available point features is changing in the image sequence, the additional procedure to handle the case is to find the set of features commonly appeared in all the three images being processed in a filtering cycle. If the set of available features, say extracted from \(f_1\) and \(f_2\) and \(f_3\), processed by the filter falls below 7, the algorithm is bootstrapped. The process followed is that images \(f_{o-1}\) and \(f_{o+1}\) are used to re-initialize the algorithm and become the base frames. A new filtering loop is then started from image \(f_{o+2}\). Here \(o\) is the number of image frames to be re-computed in the new filtering loop. For example, \(o\) is set to 5 in our implementation. In this way, the scale of the translation parameters (before and after re-initialization) can be aligned.
4. EXPERIMENTS AND RESULTS

4.1. Experiments with synthetic data

An object with 300 random feature points in 3-D was generated. The motion of the object was composed of three different segments, a pure translation section, a pure rotation section and a mixed motion section. The motion parameters were generated randomly from 0.2 to 1.2 degrees per frame for $\alpha$, $\beta$, $\gamma$, and 0.005 to 0.015 meters per frame for $x$, $y$, and $z$. The length of each synthetic sequence was 99 frames. A total of 50 independent tests were carried out. The proposed algorithm, the EKF by Azarbayejani and Pentland [5] and the 2-step EKF by Yu et al. [8], respectively, here, the total rotation of the camera were calculated using the axis-angle representation, with which the Yaw, Pitch, Roll angle was reduced into a single angle. The difference between the actual and the recovered value is the accumulated error. The accumulated total translation error was computed by subtracting the recovered translation vector from the actual one and the magnitude was taken. From the plots, it is clear that the proposed approach had a lower error than the other methods.

Fig. 2 shows the average accumulated total rotation and translation errors of the three approaches. The line with asterisk (*), triangle (△) and circle (○) markers are for our approach, the EKF by Azarbayejani and Pentland [5] and the 2-step EKF by Yu et al. [8], respectively. Here, the total rotation of the camera were calculated using the axis-angle representation, with which the Yaw, Pitch, Roll angle was reduced into a single angle. The difference between the actual and the recovered value is the accumulated error. The accumulated total translation error was computed by subtracting the recovered translation vector from the actual one and the magnitude was taken. From the plots, it is clear that the proposed approach had a lower error than the other methods.

![Fig. 2. The average accumulated total rotation (top, in degrees) and accumulated total translation error (bottom, in meters) versus frame number of the 3 algorithms.](image)

Table I shows the time needed to recover the pose when new image measurements were sequentially fed to the algorithms. Our algorithm outperformed the full covariance EKF by Azarbayejani and Pentland. However, the 2-step EKF took the shortest time to achieve the task. The reason is that their EKF is decoupled, which is actually a tradeoff between speed and accuracy.

4.2. Experiments with real images

Two real 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 sideways on a rig. The length of the image sequence is 100 frames. In the second sequence, the Grand Canyon in northwestern Arizona was viewed from an airplane. It is 5-second long and contains 50 images. The proposed algorithm was applied to track the changes of the pose.

Figs. 3 and 4 show the results. To verify whether the recovered pose is correct, corner features in the 1st image of the sequence were extracted. A set of trifocal tensors was computed using the pose parameters recovered. It was used to transfer (re-project) the corner features from the 1st image to the succeeding frames. We checked the consistency of the motion of these corner features with respect to the background images. From Figs. 3a and 4a, you can see that the features, which are indicated by cross markers, stick to the same position relative to the background. We can say that the results are accurate and visually acceptable. Figs. 3b and 4b illustrate the values of the acquired pose parameters. The line with triangle (△), circle (○) and square (□) markers on the left plot are respectively for the translation parameters $x$, $y$, and $z$, while the line with triangle (△), circle (○) and square (□) markers on the right plot are respectively for $\alpha$, Yaw), $\beta$, Pitch) and $\gamma$, Roll angle. By inspecting the motion in the original videos, the recovered pose sequences are reasonable. More results can be found at [http://www.cse.cuhk.edu.hk/~khwong/demo/](http://www.cse.cuhk.edu.hk/~khwong/demo/)

5. CONCLUSION

A high-speed recursive pose tracking algorithm has been proposed in this paper. By integrating the trifocal tensor with the extended Kalman filter, a significance improvement on the accuracy and computation efficiency has been achieved. The pose sequence can be recovered directly from images without the explicit reconstruction of 3-D structure. Thus, the procedure to handle the changeable set of point features becomes simple. It is shown in the experiment that the proposed algorithm is accurate in both simulations and real situations. Our novel approach is suitable for a wide range of image processing applications such as augmented reality and visual servoing.

| Table I |
|-----------------|-----------------|-----------------|
| **TIME REQUIRED TO PROCESS AN EXTRA IMAGE** |
| (seconds) |
| Our approach    | Azarbayejani’s EKF | Yu’s 2-step EKF |
| 1.56            | 2.60             | 0.42            |

A table showing the average CPU time for the 3 algorithms to recover the pose when extra frames were added to the image sequence.
6. ACKNOWLEDGEMENT

The work described in this paper was supported by a grant (Project No.: 4204/04E) from the Research Grant Council of Hong Kong Special Administrative Region and a direct grant (Project Code: 2050350) from the Faculty of Engineering of the Chinese University of Hong Kong.

7. REFERENCES


