ICPR 2004 Cambridge

A Fast Recursive 3D Model Reconstruction Algorithm for Multimedia Applications

Ying-Kin Yu[1], Kin-Hong Wong[1] and Michael Ming-Yuen Chang[2]

[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}@cse.cuhk.edu.hk, mchang@ie.cuhk.edu.hk

Abstract

A recursive two-step method to recover structure and motion from image sequences based on Kalman filtering is described in this paper. The algorithm consists of two major steps. The first step is an extended Kalman filter for the estimation of the object's pose. The second step is a set of extended Kalman filters, one for each model point, for refining the positions of the model features in the 3D space. The initial guess is a planar model formed under the assumption of orthographic projection on the first image. These two steps alternate from frames to frames. The planar model converges to the final structure as the image sequence is scanned sequentially. The performance of the algorithm is demonstrated with both synthetic data and real world objects. Comparisons with different approaches have been performed and show that our method is more efficient.

1. Introduction

The research work presented in this paper falls into the category of structure and motion in computer vision. The goal is to reconstruct a 3D structure and its pose from a sequence of 2D images. A major stream of the solutions is to tackle the problem in a batch. Factorization [3] and bundle adjustment [5] are common approaches. The factorization method in [3] recovers the 3D structure under the assumption of orthographic projection. The latter approach [5] has a branch called the interleaved bundle adjustment. It breaks the minimization problem into two steps so as to reduce the size of the Jacobian involved, resulting in speeding up the algorithm.

Besides the batch methods, there are solutions that recover 3D models in a sequential way. Most of them are based on Kalman filtering. The work in [2] uses extended Kalman filter for pose estimation. Some researchers adopt iterated extended Kalman filter for structure updating [1] [10]. The series of methods in [6] [7] [8] recover both the structure and motion in a recursive manner. The work in [8] is the ancestor of this series of researches. The authors apply a single iterated extended Kalman filter to recover the structure and pose of the object. Azarbayejani and Pentland describe a method in [7] that improves [8] by making an extension in recovering the camera focal length and the representation of the 3D structure. The most recent work of recursive structure recovery is by Chiuso et al [6]. They have discussed the handling of occlusion and disocclusion in their implementation. Similar Kalman filtering techniques in structure from motion have also been applied to simultaneous localization and map-building for robot navigations [9].

The two-step Kalman filter based algorithm presented in this paper is inspired by the interleaved bundle adjustment method [5]. In our algorithm, the pose and the structure of the object are computed sequentially in an interleaved sense. The main advantage of our two-step approach over the recursive algorithms in [6] and [7] is that a single extended kalman filter is broken down into smaller ones. This results in a linear time and space complexity in terms of the number of point features. The decoupling of the filters is valid in our 3D model reconstruction problem since visual features can be treated as transient entities that are matched over a certain period of time and then discarded. This strategy saves a lot of computations when the number of features needed to be handled is large. It is quite common for the reconstruction of objects with full details. Our approach also has a higher speed and better scalability over the interleaved bundle adjustment method [5]. It can handle an extra view of the object naturally by calculating the prediction and update equations for both the pose and structure only for that new measurement. However, the interleaved bundle adjustment method needs to re-compute from the first frame to the latest frame for a several iterations. In addition, the implementation of our algorithm can tackle the structure from motion problem with a changeable set of feature points. The complete 360^o view of an object can be reconstructed. We have also applied the pose sequence of a real scene acquired in the model reconstruction process to produce an augmented reality video.

2. Problem modeling

Figure 1 shows the geometry of our system. $X_i^O = [x_i^O, y_i^O, z_i^O]$ and $X_i^C = [x_i^C, y_i^C, z_i^C]$ denote the

coordinates of the point *Xi* with respect to the object coordinate frame and the camera coordinate frame respectively. $p_i = [u_i, v_i]$ is a point on the image plane. The reconstructed object is centered at the origin *Oo*. The relationship between the object frame and the camera frame can be described by the following equation:

$$X_i^C = (RX_i^O + T) + T^C \quad (1)$$

R is a 3x3 rotation matrix and *T* is a 3x1 translation matrix. T^C is a 3x1 translation matrix that brings the object in the object frame to the camera frame. The camera used in the system is calibrated with fixed focal length *f*. The camera model is full perspective and the projection can be represented as:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{f}{z_i^C} \begin{bmatrix} x_i^C \\ y_i^C \end{bmatrix} \quad (2)$$

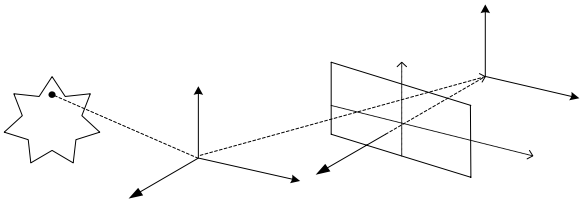

Figure 1. The geometry of our system.

3. Overview of the algorithm

The model reconstruction system can be divided into four parts: feature extraction and tracking, model initialization, pose estimation and structure updating.

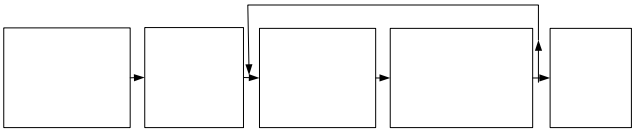

Figure 2. The flowchart of our two-step Kalman filter based algorithm.

The KLT tracker described in [4] is used to extract feature points and track them in the images. The 3D model is initialized by assuming that the projection of the first image in the sequence is orthographic.

The initial model and the second image are then fed into the first step of the main loop for pose estimation. An extended Kalman filter is adopted. The pose of the object with respect to the next image is estimated with a calculation of the prediction and update equations of the extended Kalman filter. The newly recovered pose and the input image are passed to the second step of the algorithm for structure updating.

The second step consists of a set of *N* extended Kalman filters. Each Kalman filter corresponds to each coordinate point in the reconstructed 3D model. *N* Kalman filters are needed for a model of *N* feature points. With the observations and the pose recovered for the current image frame, the coordinates of each feature point are updated accordingly. The algorithm alternates between the step 1 and 2 until all images in the sequence are used.

4. Step 1: pose estimation

Here is the dynamic model that describes the motion of the object in the EKF. *w* is the state of the system defined as:

$$w = \begin{bmatrix} t_x & \dot{t}_x & t_y & \dot{t}_y & t_z & \dot{t}_z & \alpha & \dot{\alpha} & \beta & \dot{\beta} & \gamma & \dot{\gamma} \end{bmatrix}$$

t_x, t_y, and t_z are the translations of the object along the x, y and z axes respectively. $\dot{t}_x, \dot{t}_y, \dot{t}_z$ are their corresponding velocities. α, β, γ are the Yaw, Pitch and Roll angles with $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ as their corresponding angular velocities. T_s denotes the duration over the sample period. The state transition equation for the model is:

$$\hat{w}_t = A\hat{w}_{t-1} + \gamma'_t, \quad A = diag\left\{ \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \right\}$$

γ'_t is an zero mean Gaussian noise. *A* is a 12x12 block diagonal state transition matrix. The measurement equation is:

$$\varepsilon'_t = g_t(w_t) + v'_t$$

v'_t is the zero mean Gaussian noise. ε'_t is a *n*x1 column vector representing the real measurements from the image sequence for *n* selected feature points in the object. $g_t(w)$ is the *n*x1-output projection function similar to equation (2).

In our system, a fixed number of feature points (e.g. 150 in our experiment) extracted by the tracker are passed to the EKF for pose estimation. The model points are chosen based on how much they are updated in the step of structure refinement. Those points that are steady and have a high tendency to remain at the same 3D coordinates are used. The reason is that less update on a point implies that the point is in an accurate position. The EKF implementation followed is straightforward, which can be found in related textbooks.

5. Step 2: structure updating

For *N* model points, *N* EKFs are needed for the structure update. The model is assumed to be static. The dynamic model of a 3D point in the structure and its measurement equation are.

$$X_t = X_{t-1} + \gamma_t$$
$$\varepsilon_t = h_t(X) + v_t$$

γ_t and ν_t are the zero mean Gaussian noise. ε_t is the real measurement from the image sequence. $h_t(X)$ is the projection function of the system, which can be found by substituting X into equation (1) and (2). Again, the implementation of EKF in this step is standard and the formulation is not repeated here.

6. Experiments and results

6.1. Experiments with synthetic data

In the synthetic experiment, an object with 300 random feature points in 3D within a cube of volume of $0.13m^3$, centered at a place 0.33m away from the camera, was generated. The camera has a focal length of 6mm. Its sensor has a zero mean Gaussian noise with standard deviation 0.1 pixels. The object was moving with a steady motion at a rate of [0.005 0.01 0.02] degrees and [0.001 0.002 0.0003] meters per frame for *[Yaw Pitch Row]* and *[Tx Ty Tz]* respectively. Random noise of 0.01 degrees and 0.0005 meters were added to each rotation angle and translation parameter respectively. 300 frames were generated for each test and a total of ten independent tests were carried out. Our Kalman filter based algorithm, the interleaved bundle adjustment method [5] and the EKF by Azarbayejani and Pentland [7] were tested and compared.

Figure 3 shows the average model error of the ten tests. Here the model error is defined as the percentage of mean square error of the 3D coordinates in the object coordinate frame. With our algorithm, the average error is 0.7%. The best-case error is well below 0.1%.

The performances in speed of the three algorithms are shown in figure 4 and 5. Figure 4 shows the time for the three algorithms to optimize the image residual error of the back-projected model. By careful analysis, you can see that our algorithm minimizes the residual error to a low level in a shorter time than the other two algorithms. Our approach finishes the processing of the 300-frame sequence in 133 seconds but the EKF by Azarbayejani and Pentland and the interleaved bundle adjustment method complete at 786 and 436 seconds respectively. Figure 5 shows the time needed to reconstruct a model when extra frames were added sequentially to the image sequence. The first step in creating this plot was to reconstruct a model with the first 10 frames. The succeeding 40 frames were sequentially fed to the algorithm as the new measurements of the scene. You can see that our approach outperformed the other two algorithms. Our algorithm takes only 0.5 seconds to update the structure of the scene for every extra frame added to the image sequence. The EKF by Azarbayejani and Pentland takes about 3 seconds while the interleaved bundle adjustment method takes at least 10 seconds to do the same task.

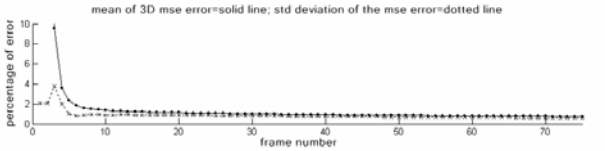

Figure 3. The average 3D model error versus frame number with our algorithm. The solid line is the average error. The dotted line is the standard deviation of the ten test cases.

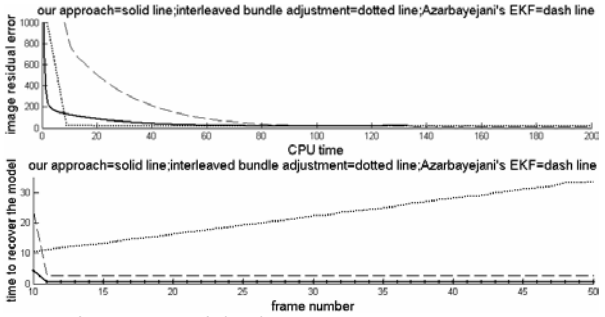

Figure 4, 5. Figure 4 (top) shows the relationship between the CPU time and the image residual error. Figure 5 (bottom) shows the time needed to reconstruct a model when extra frames are added to the image sequence. In these 2 plots, the solid line, dotted line and the dash line are for our two-step approach, the interleaved bundle adjustment method and the EKF by Azarbayejani and Pentland respectively. Note that the 3 algorithms were implemented in Matlab with a Pentium III 1GHz machine and the time measurement is in seconds

6.2. Experiments with real scene

Experiment using real scene images was also performed. The test image sequence presented in this paper was captured by translating the camera sideway on a rig. The length of the image sequence is 100 frames. Our two-step Kalman filter based algorithm was applied to acquire the 3D models. After that, the wire-frame of the object was built and texture from the appropriate images was mapped to the recovered structure. The resultant object was output in the form of VRML file. The pose sequence acquired in the model reconstruction process was also applied to produce an augmented reality video, in which a synthetic car was put onto the yellow box in the real scene.

Figure 6 shows the results of the experiment. We have successfully reconstructed the laboratory scene model. The total number of point features present in the model is about 500. The quality is good in general. For the augmented reality video, you can see that the orientation of the synthetic car is consistent with the real scene. More results, including the reconstruction of the complete 360^o view of a paper box, can be found at www.cse.cuhk.edu.hk/~khwong/demo/

Figure 6. Reconstruction results of the laboratory scene. The first row: The first and the last image of the laboratory sequence. The second and the third row: The reconstructed 3D model viewed in Cortona. Here are the two views with texture mapping (on the left) and their wire-frames (one the right). The forth row: A synthetic car, which is drawn in blue, was augmented into the real scene.

7. Conclusion

Our proposed two-step algorithm achieves linear time and space complexity in terms of the number of available point features, which is lower than the extended Kalman filter based approach by Azarbayejani and Pentland [7]. The interleaving of pose and structure recovery reduces the number of parameters to be estimated in computation of the filters. Our approach also has better computation efficiency than the interleaved bundle adjustment method. We have used our algorithm to recover the models from real scenes and applied the recovered pose sequence to augmented reality applications.

8. 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. CUHK4389/99E)

9. References

[1] P.A. Beardsley, A.Zisserman and D.W.Murray, “Sequential updating of projective and affine structure from motion”, IJCV 23, pp235-259, 1997.

[2] V.Lippiello, B.Siciliano and L.Villani, “Position and orientation estimation based on Kalman filtering of stereo images”, Proc. of the IEEE Intl. Conf. on Control Applications Mexico City, pp702-707, 2001.

[3] C.Tomasi and T.Kanade, “Shape and motion from image streams under orthography: A factorization method”, IJCV 9(2), 137-154, 1992.

[4] C.Tomasi and T.Kanade, “Detection and Tracking of Point Features”, Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[5] 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.

[6] A.Chiuso, P.Favaro, H.Jin and S.Soatto, “Structure from motion casually integrated over time”, IEEE Trans. on PAMI, vol24, No. 4, 2002.

[7] A.Azarbayejani and A.P.Pentland, “Recursive estimation of motion, structure, and focal length”, IEEE Trans. on PAMI, vol 17, no 6, June 1995.

[8] T.J.Broida, S.Chandrasiiekhar and R.Chellappa, “Recursive 3-D motion estimation from monocular image sequence”, IEEE Trans. on Aerospace and Electronic Systems, vol 26, no. 4, July 1990.

[9] A.J.Davison and D.W.Murray, “Simultaneous localization and map-building using active vision”, IEEE Trans. on PAMI, Vol.24, No.7, July 2002.

[10] C.G.Harris and J.M.Pike, “3D positional integration from image sequence”, Image and Vision Computing, vol. 6, No. 2, 1988.