Model reconstruction and pose acquisition using extended Lowe's method

Michael Ming-Yuen Chang and Kin-Hong Wong

***Abstract*—Finding the pose and structure of an unknown object from an image sequence has many applications in graphics, virtual reality and multimedia processing. In this paper we address this problem by using a two-stage iterative method. Starting from an initial guess of the structure, the first stage estimates the pose of the object. The second stage uses the estimated pose information to refine the structure. This process is repeated until the difference between the observed data and data re-projected from the estimated model is minimized. This method is a variation of the classical bundle adjustment method, but is faster in execution and is simpler to implement. Synthetic and real data have been tested with good results.**

***Index Terms*—3D structure acquisition, structure from motion, pose estimation, Lowe's method, bundle adjustment.**

I. Introduction

This work investigates the problem of finding both the pose and structure of an unknown object from an image sequence. This line of research is known as structure from motion (SFM) in the literature. There are many applications related to this research. For example, in 3D object reconstruction, in creating a real-life scene useful in virtual reality, and in mixing real scenes with artificial objects in augmented reality.

Many SFM approaches are based on the method of factorization proposed by Tomasi and Kanade [37]. The main idea is that the motion of the 2D features depends on the object's structure and the motion parameters (rotation, translation) involved. Factorization provides a way to recover the structure and motion parameters involved in generating the motion. The method is originally designed for orthographic cameras but later versions are able to handle other projection models such as weak perspective, para-perspective, affine and full perspective [22][32].

Another common approach is based on epipolar geometry. A pair of frames in the image sequence is used to calculate the fundamental matrix, which contains information about the camera motion. If the camera intrinsic parameters of the camera are known, the camera motion (or extrinsic parameters) can be obtained up to a scale factor.

M.Y.Y. Chang is with the Information Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong . Email: mchang@ie.cuhk.edu.hk

K.H. Wong is with the Computer Science and Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong . Email: khwong@cse.cuhk.edu.hk

This work was supported by a grant from the Research Grant Council of Hong Kong Special Administrative Region. (Project Number. CUHK4389/99E)

First submitted on 3 Jan. 2003.

If the parameters are not known, we can still obtain the camera motion and 3D structure but only up to a particular projective camera model. These techniques and algorithms are explained in [40]. If we have a sequence of more than two images of an object, we can utilize the additional information to improve the result. The Kalman filter based approach is a popular choice [12] for combining information from an image sequence.

Some researchers work on a class of techniques based on space carving and silhouette ([14], [15]). The concept is to use the information in the input images to remove those 3D parts that are inconsistent with the projections to obtain the final 3D model. However, since each volume element (VOXEL) is required to be processed independently, it is in general a very slow process and requires a lot of working memory.

Another novel approach is based on the Markov-Chain Monte Carlo (MCMC) method, which is able to recover the model with little prior knowledge [6]. The drawback is that the computational cost is considerable.

Models produced by the algorithms described above typically require further refinement. Bundle adjustment is the most accurate and common technique in use. It is a global optimization technique that aims to reduce the errors between the 2D feature points and the predicted feature points from the model ([41], [39] and [2]). A system that combines the use of epipolar geometry, Kalman filtering and bundle adjustment has been reported by Pollefeys [28]. The model obtained by making use of epipolar geometry and Kalman filtering is treated as the initial guess for a bundle adjustment process. The main drawback of bundle adjustment is its slow speed. This problem is particularly acute if the number of parameters involved is large. Ways of improving the speed of the process are suggested in [28].

In this paper, we propose to solve the SFM problem by using a two-stage bundle adjustment method. It is similar to the classical bundle adjustment method, but can run at a faster speed. In our method, each iterative step has two stages. The first stage uses an approximate model to estimate the pose of the object. The second stage uses the pose information to refine the model structure. The two stages are executed repeatedly until the difference between the observed data and data re-projected from the estimated model is minimized.

A number of pose estimation algorithms have been proposed in the literature [17], [18], [24], [25] and [26]. The pose estimation algorithm we used is based on the work by Lowe [19]. Lowe's method is a model-based algorithm, which can estimate the pose of the model provided that the structure of the model is known. Our method can be considered as an extension of Lowe's method. In applications that require the tracking of the pose of an unknown object, e.g. the pose of a person's head [16], our algorithm could be used to recover the pose even if a model of the person's head is not available.

The main contribution of this paper is to propose an efficient and practical SFM algorithm that can be used by the multimedia community for model generation. We also provide explicit analytic formulation in each step of the iterative algorithm. Our system can be used to construct models from images using simple web cameras. Both synthetic and real images have been tested with good results. In our experiment, we constructed a turntable for capturing the rotational motion of the objects. Our system was able to reconstruct 3D models of the objects, which can be viewed at different angles interactively by a VRML 3D browser.

Organization of the paper is as follows: in section II, we will describe the theory used in our approach. In III, we will describe the experimental results of this work. In IV we will discuss the results, and V is the conclusion.

II. THEORY

A. Problem setting

A camera at the world center O_W has a focal length of f. It takes an image sequence of a model M that has N feature points represented by $P=\{P_1,\ldots,P_i,\ldots,P_N\}$, where $P_i=[X_i,Y_i,Z_i]$ is the 3D position of the i^{th} point. The position of the model at time $t=1$ serves as the reference position of the model. The model is moved to new positions by a set of rotation R_t and translation T_t transformations at time $t=1,\ldots,\Gamma$. The set of 3D points P after these operations are projected to the image plane of the camera by the function $g()$. A set of corresponding 2D image points $q_t=\{q_{1,t},\ldots,q_{N,t}\}$ at time *t* is formed.

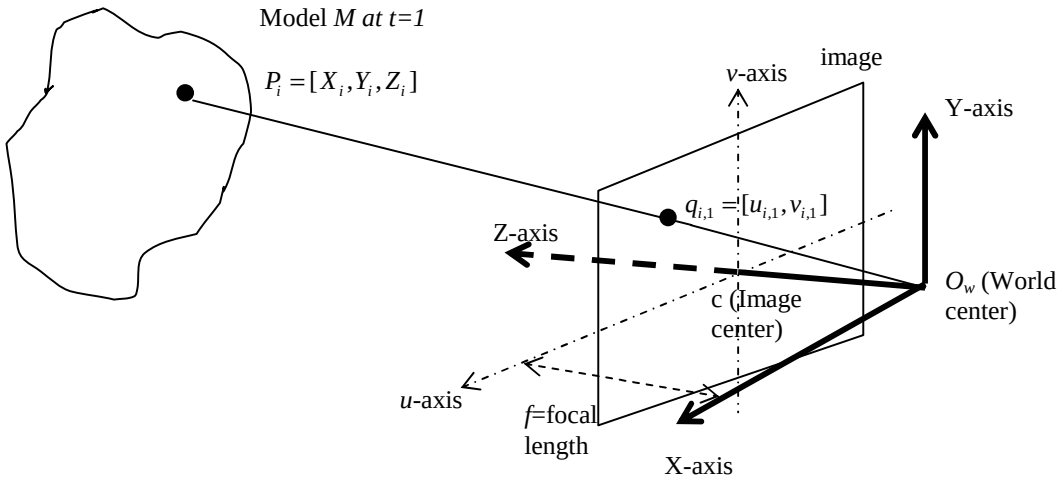


Figure 1 : Perspective projection of an object onto an image

Specifically, the 2D projection point $q_{i,t}$ can be expressed as $q_{i,t}=(u_{i,t},v_{i,t})=g(R_tP_i+T_t)$, where $T_t=([T_1,T_2,T_3]^T)_t$ is the translation vector, $R_t=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is the rotational matrix, and

$$u=f\frac{r_{11}X+r_{12}Y+r_{13}Z+T_1}{r_{31}X+r_{32}Y+r_{33}Z+T_3}$$
$$v=f\frac{r_{21}X+r_{22}Y+r_{23}Z+T_2}{r_{31}X+r_{32}Y+r_{33}Z+T_3} \quad (1)$$

represent the position of a perspectively projected point in the image plane. The image formation process can alternatively be expressed by $q=g(\theta,P)$, where θ represents the pose information (T,R).

B. Pose estimation by Lowe's method

The pose estimation problem can be summarized as follows. Given a known model M and a set of corresponding 2D image points q, find the pose parameter θ of the model so that $q_i = g(RP_i + T)$ for all i.

Lowe's method [19] provides a solution to this problem. Let $\tilde{\theta}$ be an initial estimate of the pose and θ be the true pose, so that $\theta = \tilde{\theta} + \delta\theta$. Expand $q_i = g(RP_i + T)$ into a series, we have

$$q_i = g(\theta, P_i) = g(\tilde{\theta}, P_i) + \frac{dg(\tilde{\theta}, P_i)}{d\tilde{\theta}}\delta\theta + \cdots \tag{2}$$

Rearranging the equation and let $e_i = q_i - g(\tilde{\theta}, P_i)$ be the residual error between the observation and the prediction, we have,

$$e_i = \frac{dg(\tilde{\theta}, P_i)}{d\tilde{\theta}}\delta\theta \tag{3}$$

$\delta\theta$ can now be found because e_i is measurable and the derivative of $g(\tilde{\theta}, P_i)$ can be calculated. Given N model points, equation (3) forms a system of linear equations. $\delta\theta$ can be found using least-squares methods. An improved estimate of the pose is given by $\tilde{\theta}_{i+1} = \tilde{\theta}_i + \delta\theta_i$. This process of pose estimation is repeated iteratively until the residual error is sufficiently small.

The following describes an implementation of Lowe's method by Trucco and Verri [40]. Let the estimated pose $\tilde{\theta} = [T_1, T_2, T_3, \phi_1, \phi_2, \phi_3]$, where ϕ_1, ϕ_2, ϕ_3 are the roll, pitch, and yaw (RPY) angles respectively. For an image point $q_i = (u_i, v_i)$, the change in (u_i, v_i) as a result of the change in θ is given by

$$\begin{aligned} \delta u_i &= \sum_{j=1}^{3}[\frac{\partial u_i}{\partial T_j}\Delta T_j + \frac{\partial u_i}{\partial \phi_j}\Delta\phi_j], \\ \delta v_i &= \sum_{j=1}^{3}[\frac{\partial v_i}{\partial T_j}\Delta T_j + \frac{\partial v_i}{\partial \phi_j}\Delta\phi_j]. \end{aligned} \tag{4}$$

We would like to find ΔT_j and $\Delta\phi_j$ in (4) so as to produce the right amount of shifts $(\delta u_i, \delta v_i)$ that compensate the error e_i in (3). Since e_i can be measured, if the derivatives in (4) are known, then ΔT_j and $\Delta\phi_j$ can be found.

To compute these terms, first let $P_i = (X_i, Y_i, Z_i)$ be a model point in the reference position. Let (X_i^R, Y_i^R, Z_i^R) be the position of the model point after rotation, where

$$\begin{aligned} X_i^R &= r_{11}X_i + r_{12}Y_i + r_{13}Z_i \\ Y_i^R &= r_{21}X_i + r_{22}Y_i + r_{23}Z_i \\ Z_i^R &= r_{31}X_i + r_{32}Y_i + r_{33}Z_i \end{aligned} .$$

And finally, let (X_i', Y_i', Z_i') be the model point after rotation and translation, represented by

$$\begin{aligned} X_i' &= X_i^R + T_1 \\ Y_i' &= Y_i^R + T_2 \\ Z_i' &= Z_i^R + T_3 \end{aligned}$$

Differentiate equation (1) with respect to the various terms, we have

$$\frac{\partial u_i}{\partial T_1} = \frac{f}{Z_i}, \quad \frac{\partial u_i}{\partial T_2} = 0, \quad \frac{\partial u_i}{\partial T_3} = -f\frac{X_i}{Z_i^2},$$

$$\frac{\partial v_i}{\partial T_1} = 0, \quad \frac{\partial v_i}{\partial T_2} = \frac{f}{Z_i}, \quad \frac{\partial v_i}{\partial T_3} = -f\frac{Y_i}{Z_i^2},$$

$$\frac{\partial u_i}{\partial \phi_1} = -f\frac{X_i Y_i^R}{Z_i^2}, \quad \frac{\partial u_i}{\partial \phi_2} = f\frac{X_i X_i^R + Z_i Z_i^R}{Z_i'^2}, \quad \frac{\partial u_i}{\partial \phi_3} = -f\frac{Y_i^R}{Z_i},$$

and $$\frac{\partial v_i}{\partial \phi_1} = -f\frac{Y_i Y_i^R + Z_i Z_i^R}{Z_i^2}, \quad \frac{\partial v_i}{\partial \phi_2} = f\frac{X_i^R Y_i}{Z_i^2}, \quad \frac{\partial v_i}{\partial \phi_3} = f\frac{X_i^R}{Z_i}.$$

If the model has *N* feature points, then from (4) we have 2*N* equations for 6 unknowns $(\Delta T_1, \Delta T_2, \Delta T_3, \Delta\phi_1, \Delta\phi_2, \Delta\phi_3)$. A solution for ΔT and $\Delta\phi$ can be found using standard least squares methods [31]. Improved estimates are given by $\tilde{T} + \Delta T$ and $\tilde{\phi} + \Delta\phi$.

C. Overview of the two-stage algorithm for finding pose and structure from an image sequence

If the structure of a model is known, then one can find the pose of the model by Lowe's method. Alternatively, if the pose of the model is known, then it is possible to recover the model's structure. Based on these ideas, the two-stage algorithm proposed here first uses a rough model to estimate the poses of a sequence of images by Lowe's method. The information obtained is then used to refine the model. This process is repeated iteratively.

D. Steps of our algorithm

Steps of our structure from motion algorithm for an image sequence of *N* features and Γ frames are described below.

1) Camera calibration and Feature extraction

The camera is calibrated by the tools described in [4]. Then we use the KLT tracker described in [38] to extract the feature set $\{q_{i,t}\}$ from a image sequence, where $i = 1, 2, ..., N$, and $t = 1, 2, .., \Gamma$.

2) Model and system initialization

The first stage of our method uses an approximate model to estimate the pose of an image sequence. Such a model can readily be obtained if we could assume the projection is orthographic. Our algorithm can deal with perspective images, but using orthographic projection for the first guess will provide an initial model for us to start the iteration. Based on this assumption, the depth of the object should be less (e.g. 1/3 or less) than the distance between the object and the camera. The unknown model can then be approximated by a planar object located at a distance Z_{init} from the camera. If the value of Z_{init} cannot be known exactly, then we can still recover the model structure, but only up to a scale factor.

The orthographic projection assumption is used only once at the initialization stage so as to provide the initial model. Perspective projection mapping is used throughout subsequent operations.

In our work, we use the first image in the sequence to construct this initial planar model. Features in the first image are back-projected from the image plane to an object plane located at a distance Z_{init} from the camera. Specifically, the i^{th} image point in the first frame $q_{i,1} = [u_{i,1}, v_{i,1}]$ is mapped to a 3D point $[X_i, Y_i, Z_i]$ of the model by

$$X_i = \frac{Z_{init}}{f} u_i, \quad Y_i = \frac{Z_{init}}{f} v_i, \quad \text{and } Z_i = Z_{init} \tag{5}$$

The initial model can also be obtained in a variety of ways. If the object's geometry is known to be concave or convex, then we can use a concave or convex surface to serve as the initial guess. We could also replace the planar model by a random model with a random set of Z_{init}. If the initial guess is too poor, then the iterative method will produce an incorrect model that has a large residual error. In this case, we can use a different random model and repeat the process until a good model is found. This type of RANSAC-like approach could produce a good result, however, is very time-consuming. From our experience, we found that a planar surface was good enough for most experiments.

3) First pass -- Use an estimated model to find the pose sequence

For *t=1 to* Γ

Use $\{P_1, \ldots, P_N\}$ and the corresponding image features $\{q_{1,t}, q_{2,t}, .., q_{N,t}\}$ of the t^{th} image frame to find $\tilde{\theta}_t$ by Lowe's method

End

Result: An estimated pose sequence $\tilde{\theta} = \{\tilde{\theta}_1, \ldots, \tilde{\theta}_\Gamma\}$ is found.

Description: Based on the planar model obtained from the initialization, we can estimate the pose $\tilde{\theta}_t$ of the t^{th} image by Lowe's method as described earlier. It is important to note that the pose for each of the Γ frames can be estimated independently. Hence the altogether $6 \times \Gamma$ unknown pose parameters can be found in Γ steps. Each step involves the estimation of only 6 unknowns. The same number of unknowns reduces the computational cost considerably.

4) Second pass – Based on the estimated poses, re- estimate the model from a sequence of images

For *i=1 to N*

From the estimated poses $\{\tilde{\theta}_1, \ldots, \tilde{\theta}_\Gamma\}$ and the i^{th} image features in all the image frames $\{q_{i,1}, q_{i,2}, .., q_{i,\Gamma}\}$, refine the model point P_i by a least-squares method to minimize residual image error.

End

Result: An improved structure $\{P_1, \ldots, P_N\}$ is found.

Description: Based on the pose sequence $\{\tilde{\theta}_1,\ldots,\tilde{\theta}_\Gamma\}$ obtained from the first pass, we can now improve our estimation of the 3D model points $\{P_1,\ldots,P_N\}$ using the similar approach as in equation (2). Like step 3, each model point can be estimated independently. The estimation of the complete model can be divided into *N* independent steps. Each step can be computed efficiently. As in (2), by keeping $\tilde{\theta}$ constant, and varying P_i, we have

$$q_{i,t} = g(\tilde{\theta}_t, \tilde{P}_i) = g(\tilde{\theta}_t, \tilde{P}_i) + \frac{dg(\tilde{\theta}, \tilde{P}_i)}{d\tilde{P}_i}\delta\tilde{P}_i + \cdots$$

Rearranging and keeping only the first order terms, we have

$$e_{i,t} = \frac{dg(\tilde{\theta}, \tilde{P}_i)}{d\tilde{P}_{ii}}\delta\tilde{P}_i = \frac{dg(\tilde{\theta}, \tilde{P}_i)}{d\tilde{P}_i}\frac{d\tilde{P}_i}{dX_i}\delta X_i + \frac{dg(\tilde{\theta}, \tilde{P}_i)}{d\tilde{P}_i}\frac{d\tilde{P}_i}{dY_i}\delta Y_i + \frac{dg(\tilde{\theta}, \tilde{P}_i)}{d\tilde{P}_i}\frac{d\tilde{P}_i}{dZ_i}\delta Z_i,$$

where

$$e_{i,t} = q_{i,t} - g(\tilde{\theta}_t, \tilde{P}_i) = [\delta u_{i,t}, \delta v_{i,t}] \tag{6}$$

is the discrepancy between the position of the observed image point and the predicted value. From (1), differentiating u and v with respect to X_i, Y_i, Z_i, we have

$$\begin{aligned} \delta u_{i,t} &= a^i_{11}\delta X_i + a^i_{12}\delta Y_i + a^i_{13}\delta Z_i \\ \delta v_{i,t} &= a^i_{21}\delta X_i + a^i_{22}\delta Y_i + a^i_{23}\delta Z_i \end{aligned} \tag{7}$$

where

$$a^i_{11} = f\left[\frac{r_{11}}{Z_i} - \frac{r_{31}X_i}{Z_i^2}\right], a^i_{12} = f\left[\frac{r_{12}}{Z_i} - \frac{r_{32}X_i}{Z_i^2}\right], a^i_{13} = f\left[\frac{r_{13}}{Z_i} - \frac{r_{33}X_i}{Z_i^2}\right],$$
$$a^i_{21} = f\left[\frac{r_{21}}{Z_i} - \frac{r_{31}Y_i}{Z_i^2}\right], a^i_{22} = f\left[\frac{r_{22}}{Z_i} - \frac{r_{32}Y_i}{Z_i^2}\right], a^i_{23} = f\left[\frac{r_{23}}{Z_i} - \frac{r_{33}Y_i}{Z_i^2}\right].$$

For a sequence of Γ images, we have Γ observations of the same model point. This gives us 2Γ equations with only three unknowns $[\delta X, \delta Y, \delta Z]$,

$$\begin{pmatrix} \delta u_{i,1} \\ \delta v_{i,1} \\ \vdots \\ \delta u_{i,t} \\ \delta v_{i,t} \\ \vdots \\ \delta u_{i,\Gamma} \\ \delta v_{i,\Gamma} \end{pmatrix} = \begin{pmatrix} a^i_{11} & a^i_{12} & a^i_{13} \\ a^i_{21} & a^i_{22} & a^i_{23} \\ \vdots & \vdots & \vdots \\ a^i_{11} & a^i_{12} & a^i_{13} \\ a^i_{21} & a^i_{22} & a^i_{23} \\ \vdots & \vdots & \vdots \\ a^i_{11} & a^i_{12} & a^i_{13} \\ a^i_{21} & a^i_{22} & a^i_{23} \end{pmatrix} \begin{pmatrix} \delta X_i \\ \delta Y_i \\ \delta Z_i \end{pmatrix} \tag{8}$$

$[\delta X_i, \delta Y_i, \delta Z_i]$ can be solved from (8) by least-squares methods. An improved estimate of the i^{th} model point is given by $[X_i, Y_i, Z_i] + [\delta X_i, \delta Y_i, \delta Z_i]$. If a particular model point is not observable in some of the images, probably due to

occlusion, we could discard the corresponding $a_{m,n}^{i}$ parameters in (8).

5) Stopping rule

Error in the model and pose estimation can be measured directly by summing up all the re-projection errors in (6), as

$$error = \sqrt{\frac{1}{N\Gamma}\sum_{i=1}^{N}\sum_{t=1}^{\Gamma} e_{i,t}^{2}} \quad \text{(unit pixel)} \tag{9}$$

This figure measures the average re-projection error per point per frame. The algorithm is terminated if the rate of reduction of the error becomes too small (0.01 pixels). Otherwise the algorithm loops back to step 3.

6) Mismatched Feature filtering step

Features are tracked by the KLT tracker. Nevertheless, errors due to measurement or feature mismatches do arise. Errors due to measurement noise could be contained by using a standard weighted least squares method to replace the ordinary least squares method used here. The magnitude of the re-projection errors in equation (6) can be used to form the weighting matrix in the weighted least squares method. For errors due to feature mismatches, it will be difficult to find a model point that can provide a good fit to the observed data. Consequently, some of the re-projection errors will be significantly larger than the rest of the measurement. In this work, we compute the mean and standard deviation of the magnitude of the re-projection error ($|e_{i,t}|$) in (6). Points that lie far away from the mean (*e.g.* five times the standard deviation) are considered as mismatched features. These points are filtered. Step 1 to 6 can be executed many times until no bad feature is found.

E. Comparison with the classical bundle adjustment method

Both classical bundle adjustment and our two-stage adjustment aim at estimating the pose (θ) and model (P) parameters by minimizing the error $= \sum_{i=1}^{N}\sum_{t=1}^{\Gamma}[q_{i,t} - g(\tilde{\theta}_t, \tilde{P}_i)]^2$. The main difference between the two methods is that the classical method estimates the pose and model simultaneously, while the two-stage method estimates the pose and model separately. Compare with classical bundle adjustment, the main advantages of the two-stage method are speed, simplicity, and self-initialization. From our simulation results (section III), we did not find any significant difference in accuracy between the two methods.

For an image sequence of N model points and Γ views, let x be the state vector representing the $3N + 6\Gamma$ unknowns, $g(x)$ be the perspective transformation function that yields the set of 2D image points, and y be the vector representing the observed points. Our goal is to find an optimal x^* so that $y = g(x^*)$. Starting from an initial value x_0, and assuming that g is locally linear, then a first order approximation of $g(x^*)$ is given

by $g(x^*) = g(x_0) + J\Delta$, where J is the Jacobian and Δ is the refinement needed to yield a better estimate. An improved estimate can be obtained by applying $x_{i+1} = x_i + \Delta_i$ iteratively. The solution of Δ can be found by solving the equation $y = g(x_0) + J\Delta$ using least-squares methods. Let $e = y - g(x_0)$, so that $e = J\Delta$, we have

$$J^{\mathrm{T}} J\Delta = J^{\mathrm{T}} e \tag{10}$$

This equation is known as the normal equation. The size of $J^{\mathrm{T}} J$, Δ, and e are $(3N + 6\Gamma)\times(3N + 6\Gamma)$, $(3N + 6\Gamma)$, and $2N\Gamma$ respectively. If the size of $J^T J$ is large, then (10) is computationally expensive to solve. Faugeras *et al* [7] and Pollefeys [29] have described two ways to reduce the computation cost. First, since $J^{\mathrm{T}} J$ is a sparse matrix, solving (10) using sparse matrix operations can reduce the cost considerably. Second, it is possible to break $J^{\mathrm{T}} J$ into two smaller matrices, each of size $3N \times 3N$ and $6\Gamma \times 6\Gamma$. This also helps to reduce the computational cost. However, if either N or Γ is large, the computational cost is still considerable.

In comparison, the two-stage approach estimates the pose and model parameters separately. In the pose estimation stage, the estimation of the 6Γ unknowns is divided into Γ independent steps. Each step involves solving a normal equation like (10) but with a Jacobian of size 6×6 only. Likewise, in the model estimation stage, the estimation of the $3N$ unknowns is divided into N independent steps. Each step involves solving a normal equation with a Jacobian of size 3×3. The computational cost of the two-stage approach grows only linearly with N and Γ, which is comparatively more efficient than classical bundle adjustment for large N and Γ. Furthermore, our two-stage method only requires the support of the most rudimentary matrix operations, and does not rely on any sparse matrix operations. Our method is therefore simpler to implement than the classical bundle method.

Self-initialization is another advantage of the two-stage approach. Both of the classical bundle and two-stage bundle adjustments depend on Newton's method. It is well known that the convergence of Newton's method depends critically on the initial value used. For the two-stage method, this initial vector is readily available if the initial model can be approximated by a planar object. We can use this initial model to estimate the pose of the next frame, and then use the result to refine the model structure. This procedure is repeated until all views are covered, which completes the initialization process. The same process is then used to refine the model structure and the pose estimation until the error is sufficiently small. The transition from initialization to refinement is seamless. The implementation of the system is relatively simple. For classical bundle adjustment, it is difficult to obtain an initial vector that carries an approximate value for both the model and poses information. Typically the vector can only be found by using other methods [28]. This is why classical bundle adjustment is usually used only as part of a structure-from-motion system, typically at the last stage, where an approximate model is already available and the main purpose of bundle adjustment is to provide an accurate final structure.

III. Experimental results

A. Simulation results and comparison with classical bundle method

In this section, we compare the accuracy and the convergence rate between the classical bundle adjustment method

and the two-stage method. Fifty independent runs were carried out in our simulation. In each run, a model of 300 random points was generated. All points lied inside a cube of 0.13 m^3. The center of the model was 0.33 m away from a camera of focal length 6mm. The model was rotated and translated at a rate of [0.2, -0.2, 0.2] degrees and [0.01, 0.01, 0.01] meters per frame. To simulate a real-life scenario, additional perturbation ranging from -0.5 to 0.5 degrees for rotation and from 0 to 0.04 meters for translations were added to each frame. To simulate the measurement errors, 2D Gaussian noise (mean=0, $\sigma = \sqrt{2}$ pixels) was added to the image points. Each run used a sequence of 30 image frames. All the sequences were generated by different models but with the same pose information. The machine used was a 2.4 GHz PC running MATLAB 6.5. To provide a fair basis for comparison, the same initial model and pose information were fed to both bundle adjustment methods. Because the classical bundle method was ill-suited to provide an initial model, the two-stage method was used to build the initial model and pose information.

Fig. 2 and 3 show the tracking result. It can be seen that both algorithms were able to track the pose of the model equally well. Fig. 4 compares the convergent rate of the residual errors (an average of 50 runs) between the methods. The residual error is defined in (9). In theory, if the model and pose information can be recovered perfectly, then the residual error should be equal to the standard deviation of the Gaussian noise added, *i.e.*, $\sqrt{2}$ pixels. The result showed that both algorithms did manage to converge to this theoretical limit in all 50 runs. In fact, the residual error was even slightly less than the theoretical minimum, which we believe was due to the over-fitting of the model and pose. The result clearly showed that the two-stage method had a significantly faster rate of convergence.

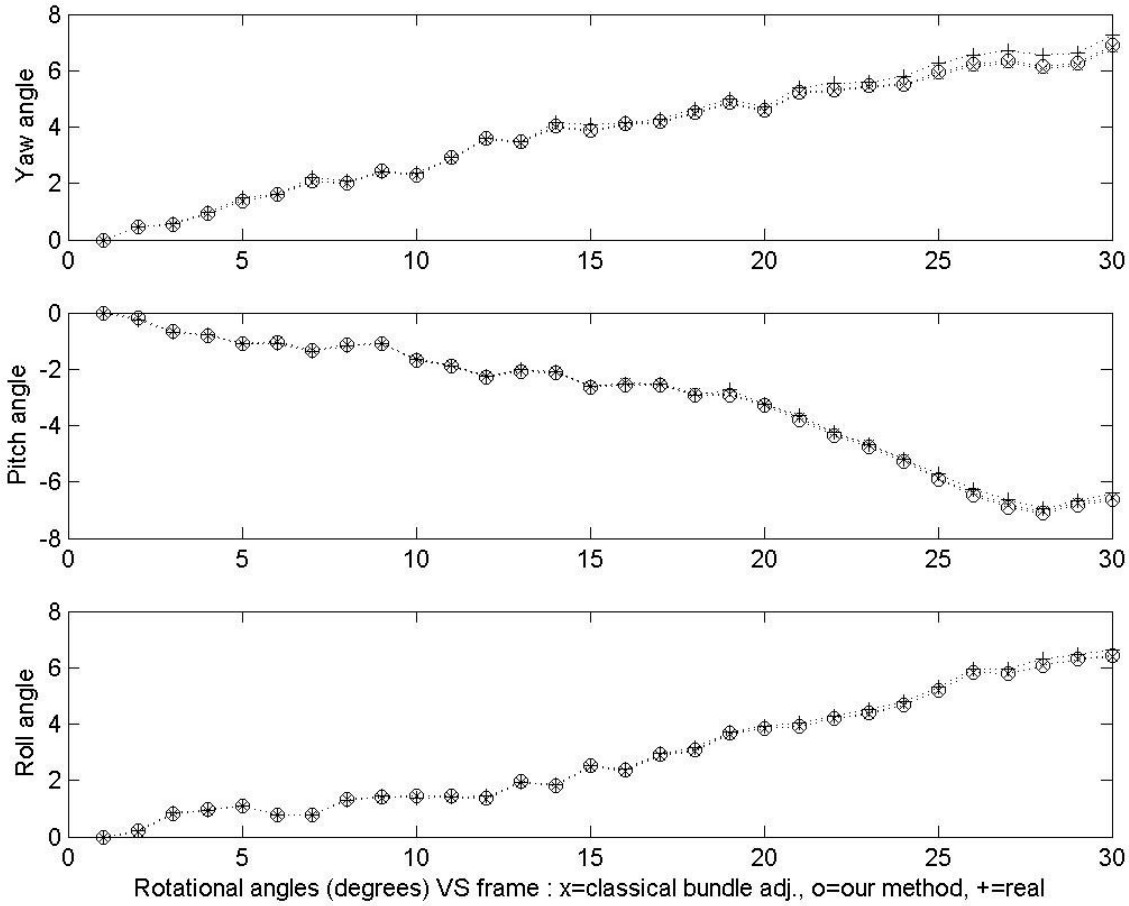


Figure 2: The pose tracking result of a test. The upper 3 diagrams show the tracked and real yaw, pitch roll angles in degrees.

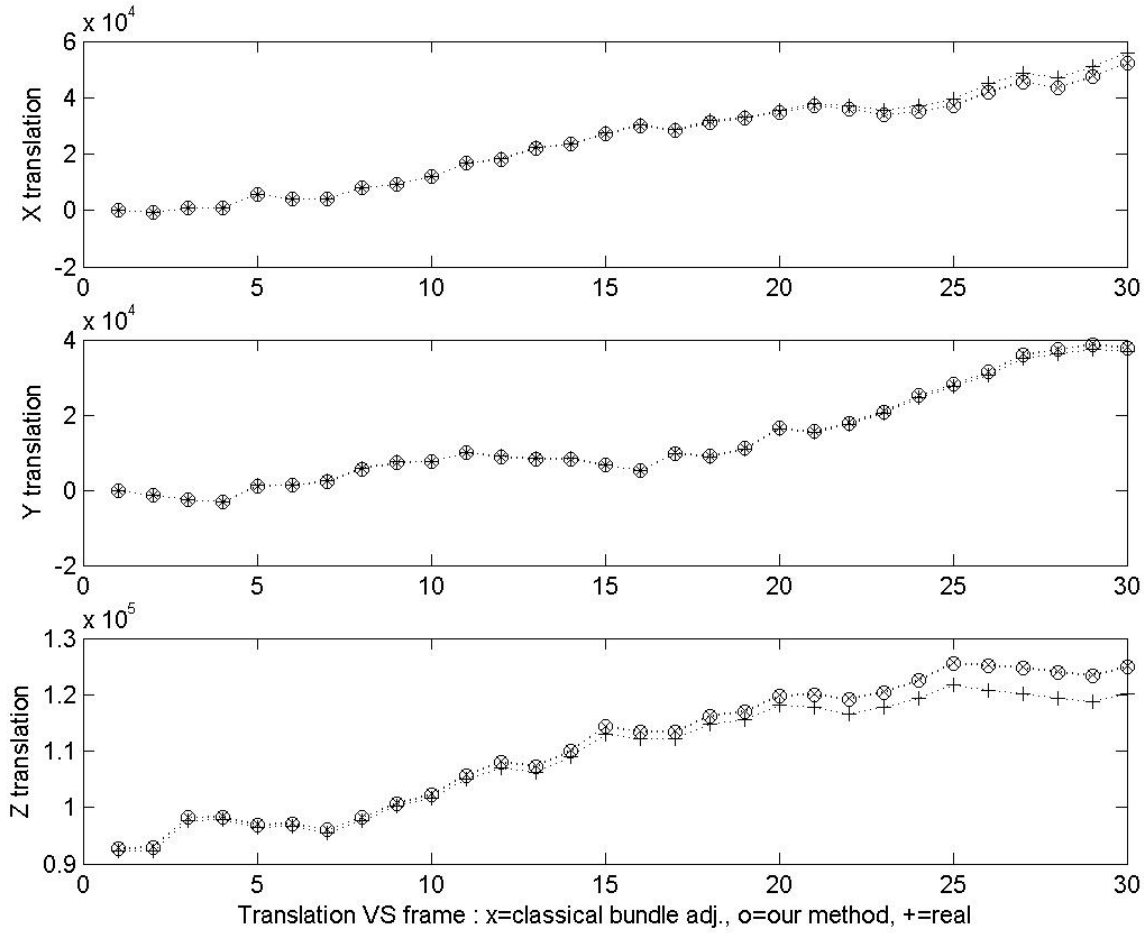


Figure 3: The pose tracking result of a test. The above 3 diagrams show the tracked and real translations of X, Y, Z in pixels (the pixel width is 5.42μm in both image *u*, *v* dimension, the focal length is 6mm).

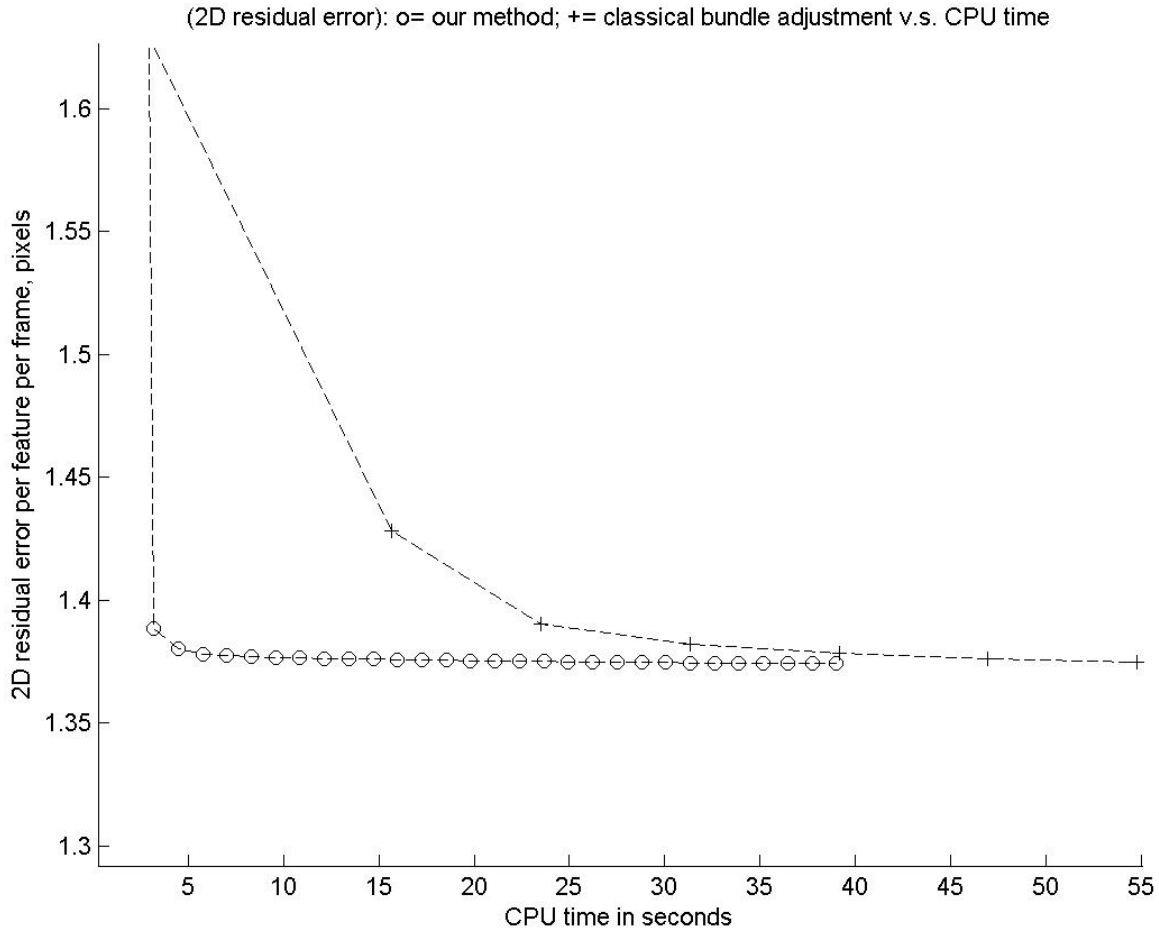


Figure 4. Compare the 2D residual errors against CPU times of our method and the classical bundle adjustment method. The lines represent the average of 50 runs.

B. Experiment on the mismatched feature filter

In this experiment, we tested the mismatch feature filter described in step 6 of our algorithm. To simulate the mismatch noise, noise equal to 14 pixels was added randomly to 5% of the data in section III-A. We repeated the simulation of 50 runs and the average result was plotted in Fig. 5. The result showed that the noise added had an impact on the residual error. After applying the filter, the residual error converged again to the expected $\sqrt{2}$ pixel.

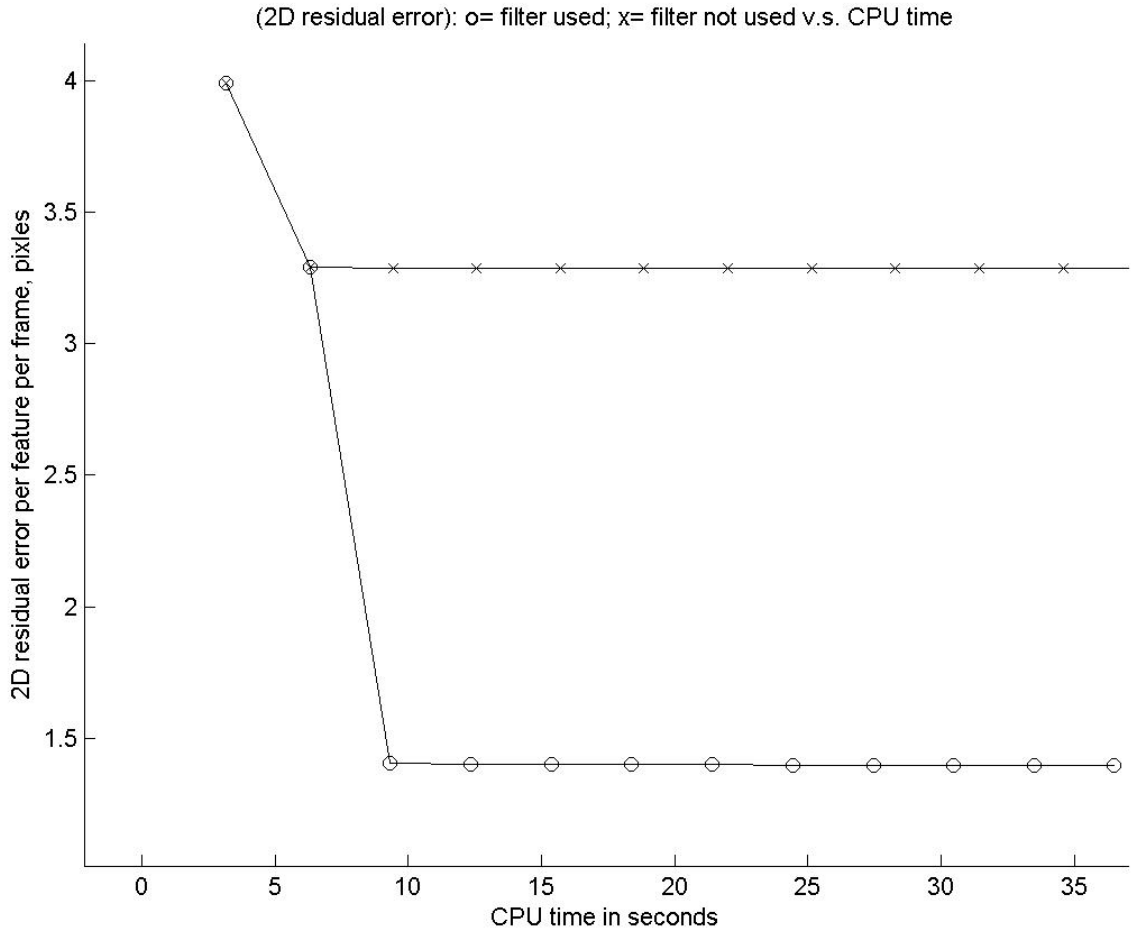


Figure 5. Effects of the mismatch filter. The lines represent the average of 50 runs.

C. Real scene experiments (more results are found at http://www.cse.cuhk.edu.hk/~khwong/demo/ct/)

A number of real objects were tested. Features in the image sequences were extracted by the KLT [38] tracker. Stationary image features obtained from objects on turntables were classified as the background or noise and were filtered. The feature sets were passed to our algorithm for model reconstruction. For each model built, we used the first image of the sequence to provide the texture for the VRML file. The results were viewed by a VRML browser.

1) A flask on a turntable

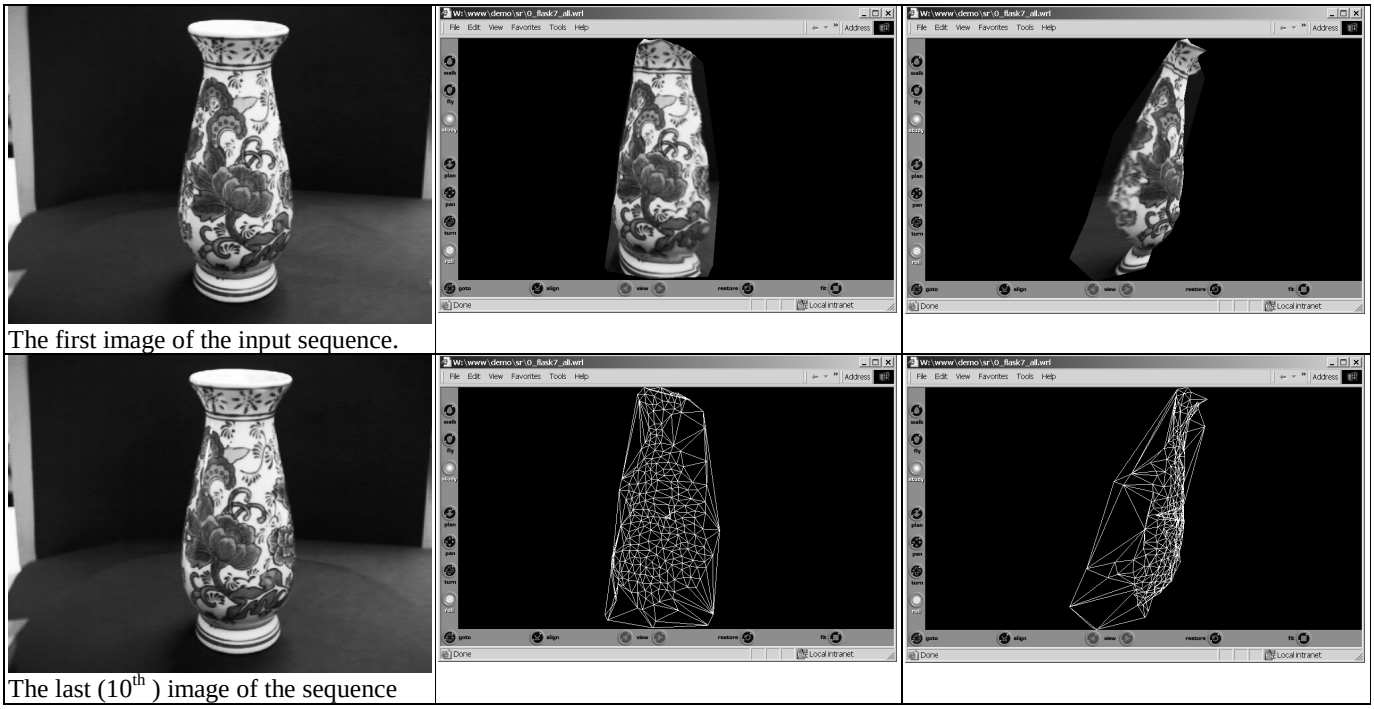


Figure 6: First column: first and last image of the input sequence. Second and third columns: two reconstructed views and their wire-frames of the object. (The initial model for the algorithm is a plane; the motion has little translation and large rotation.)

2) A box on a turntable

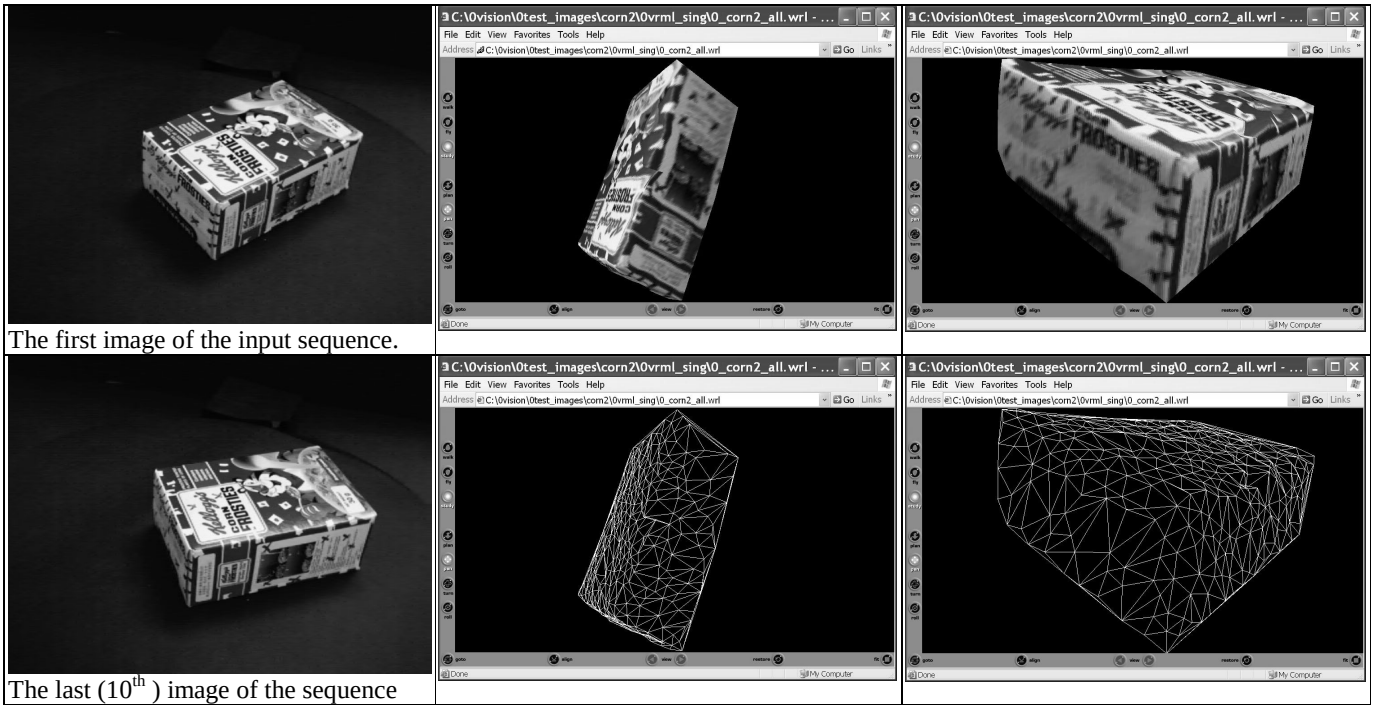
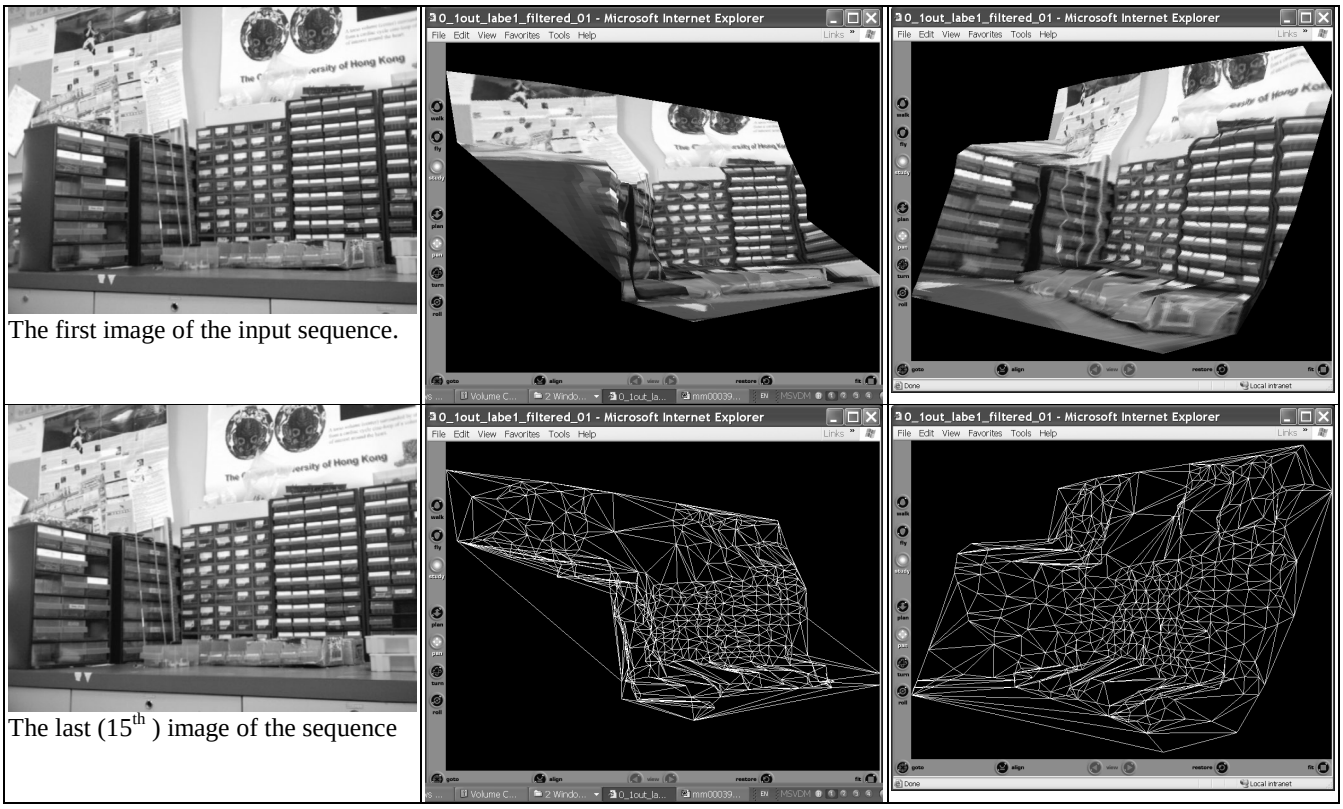


Figure 7: First column: first and last image of the input sequence. Second and third columns: two reconstructed views and their wire-frames of the object. (The initial model for the algorithm is a plane; the motion has little translation and large rotation.)

3) A laboratory scene captured by a camera moving horizontally with small amount of rotation



The first image of the input sequence.

The last (15^{th}) image of the sequence

Figure 8 : First column: first and last image of the input sequence taken by a camera translating horizontally. Second and third columns: two reconstructed views and their wire-frames of the object. (The initial model for the algorithm is a plane; the motion has little rotation and large translation).

4) Improvement made by applying the mismatch feature filter (step 6 of our algorithm)

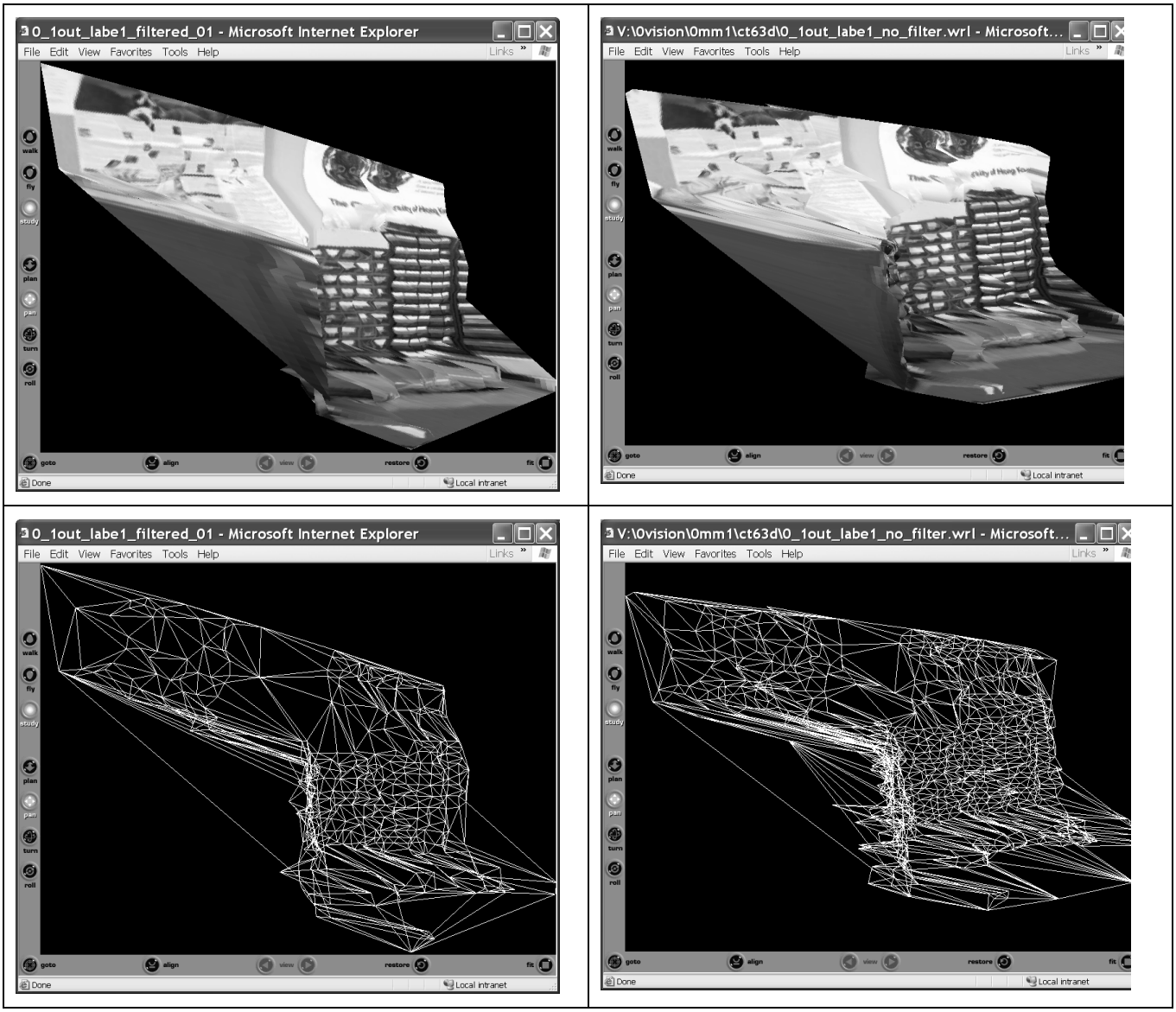


Figure 9 : The first column: Texture and wire-frame of the result after the mismatched feature filter (step 6 of the algorithm) was applied. The second column: Texture and wire-frame of the result when the filter was not applied.

It was found that the mismatched feature filter helped to remove some spikes in the reconstructed model.

IV. Discussion

A. Effect of initial guess

A good initial guess is critical to the convergence of any Newton-based methods. Prior knowledge of the structure could provide a better initial guess for the optimization. For the kind of objects that we tested, we found that good results could be obtained by simply using a planar object as the initial model. A reasonable guess of the distance of the object from the camera was also required, though it was not a critical factor.

B. Error ambiguity

It is known that the translation and rotation errors have certain correlation [35]. For example, translation error in the horizontal (*X-axis*) direction can be mixed up with rotation around the *Y*-axis (pitch), since both can generate a similar motion of the features. Future work can concentrate on how to reduce this type of error. One possible solution is to make use of some prior knowledge of the motion. For example, in the turntable case, we can assume the motion is mainly rotational; or in a robot navigation case, the motion is mainly translational.

C. Occlusion and omni-direction surface model reconstruction

Our implementation required all features to appear in all frames; hence it was not able to handle occluded points or a very long sequence (200 pictures). At present, the system can only find a partial model at a time. Nevertheless, several partial models can be combined in order to produce an omni-directional (360 degrees) view of the object. For example, a camera can capture a 360 degrees rotation of an object in a long sequence of 200 frames. The first 10 frames are used to produce the first partial model. Frame 6 to 15 can be used to produce the second partial model. Common features in the overlapping frames are responsible to find the relative pose between the two partial models so that they can be combined. Repeat this process until all the partial models are merged into an omni-direction surface model.

D. Sampling rate of the input pictures

Since the system was based on the KLT feature tracker, it was only able to track features that moved in a relatively small neighboring region in successive frames. For a casual user that captures the scene by a still-picture digital camera with a small sampling rate, unintentional panning caused by the user's hand movement may produce a large translational motion between successive pictures. As a result, the tracker may fail to track the features. We propose to solve this problem by aligning the pictures using a prominent feature point within the scene in order to reduce the effect of the spurious translation.

V. Conclusion

In this paper, we developed a method of recovering the structure and pose sequence of an object from a sequence of images. The proposed method is a variation of the classical bundle adjustment method. Within each iteration, our method estimates the structure and poses separately. This separation helps to reduce the size of the Jacobian in the computation considerably, making the method particularly efficient for scenes that have larger number of feature points and longer lengths. In our experiment, we found that both the classical bundle adjustment and our two-stage

methods converged virtually to the same minimum. This showed that the two-stage method was as accurate as the classical bundle adjustment method. The separated estimation of the structure and pose also helps to simplify the initialization process. To start the two-stage algorithm, we only need to provide an initial model. The pose sequence can be estimated by using the model provided. If the model is located not too close to the camera, then we found that the initial model can be approximated by a planar object. Prior knowledge of the object may also help to provide a better initial model. A simple outlier filtering scheme is proposed to reduce tracking errors in features extraction. Finally the model is translated into the VRML format for viewing and interactive manipulation.

We believe our algorithm is suitable for home users to develop models for Internet applications such as putting 3D objects on web pages and games. To demonstrate the feasibility, we have developed a small turntable system for scanning the 3D objects. The results were found to be satisfactory. In augmented reality applications, one can use the method to estimate the pose sequence of a scene, so that synthetic objects can be inserted into the video stream in a realistic way. Another application is head model generation and head tracking, which is an active research topic in computer vision.

We shall focus our future research in a number of directions. The first possible direction is to investigate how to improve the feature tracking module so that correct features can be obtained more reliably. Probabilistic or statically approaches, such as Condensation (Conditional Density Propagation [11], [42]), may help to improve tracking performances. The second is to explore different constraints for pose tracking. For example, the trajectories of features of an object on a rotating turntable are known to be confined to a series of parallel concentric circles. The third direction is to combine local partial models so as to construct a full omni-view surface model. The forth direction is to investigate how to reduce errors caused by rotation and translation ambiguity. We expect the high efficiency of our two-stage method will provide a firm basis to support the various improvement schemes mentioned.

VI. ACKNOWLEDGMENT

The work described in this work was supported by a grant from the Research Grant Council of Hong Kong Special Administrative Region. (Project Number. CUHK4389/99E)

VII. REFERENCES

[1] H. Araujo, R. Carceroni and C. Brown, "A Fully Projective Formulation for Lowe's Tracking Algorithm", Computer Vision and Image Understanding, Vol. 70, No. 2, May, pp. 227-238, 1998.

[2] P. Beardsley and A. Zisserman and D. Murray, "Sequential updating of projective and affine structure from motion", International Journal of Computer Vision, (23), No.3, Jun-July 1997, P235-259.

[3] D. Birch, "KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker", (http://robotics.stanford.edu/~birch/klt/).

[4] Jean-Yves Bouguet , *Camera Calibration Toolbox for Matlab* (http://www.vision.caltech.edu/bouguetj/calib_doc/)

[5] A. Chiuso, P. Favaro, J. Hailin and S. Soatto, "Structure from motion causally integrated over time", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 24 Issue: 4 , April 2002 Page(s): 523 –535.

[6] F. Dellaert, S.M. Seitz, E.C. Thorpe and S. Thrun. "Structure from motion without correspondence". In Proc. CVPR, pages 557--564, June 2000.

[7] O. Faugeras, Q.T. Luong and T. Papadopoulo, *The Geometry of Multiple Images: The Laws that Govern the Formation of Multiple Images of a Scene and some of Their Applications*, MIT Press.

[8] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. IEEE Transactions on Systems, Man, and Cybernetics, 6(19):1426--1446, November/December 1989.

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

[10] T.S. Huang and A.N. Netravali, "Motion and Structure from Feature Correspondences: A Review", Proceedings of the IEEE, Vol. 82, No.2, Feb 1994.

[11] M. Isard and A. Blake, "CONDENSATION -- conditional density propagation for visual tracking", Proceedings of IEEE International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, 26-27, September 1999, Corfu, Greece.

[12] T. Jebara, A, Azarbayejani and Pentland, A. "3D structure from 2D motion", IEEE Signal Processing Magazine , Volume: 16 Issue: 3 , May 1999 Page(s): 66 –84.

[13] F. Kahl, A. Heyden and L. Long, "Minimal projective reconstruction including missing data Pattern Analysis and Machine Intelligence", IEEE Transactions on , Volume: 23 Issue: 4 , April 2001 Page(s): 418 –424.

[14] K.N. Kutulakos, "A Theory of Shape by Space Carving", International Journal of Computer Vision 38(3): 199-218; Jul 2000.

[15] A. Laurentini, "The visual hull concept for silhouette-based image understanding Laurentini, A.", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 16 Issue: 2 Feb 1994 Page(s): 150 -162.

[16] K.S. Lee, K.H. Wong, S.H. Or and Y.F. Fung, "3D Face Modelling From Perspective-Views and Contour-Based Generic-Model", Real-Time Imaging Journal April, 2001.

[17] M.L. Liu and K.H. Wong, "A Novel Algorithm for Recovering the 3D Motions of Multiple Moving Rigid Objects",14th International Conference on Pattern recognition (ICPR'98), Brisbane, Australia, 17-20 Aug. 1998.

[18] M.L. Liu and K.H. Wong, "Pose Estimation Using Four Corresponding Points", Pattern Recognition Letters, Volume 20, Number 1 January 1999, pp. 69-74.

[19] D.G. Lowe, "Fitting Parameterized Three-Dimensional Models to Images", IEEE Pattern Analysis and Machine Intelligence, Volume: 13 Issue: 5 , May 1991 Page(s): 441 -450.

[20] C.P. Lu, G.D. Hager, E. Mjolsness, "Fast and globally convergent pose estimation from video images Pattern Analysis and Machine Intelligence", IEEE Transactions on , Volume: 22 Issue: 6 , June 2000 Page(s): 610 –622.

[21] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(8):858--867, 1997.

[22] T. Morita, T. Kanade, "A sequential factorization method for recovering shape and motion from image streams Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 19 Issue: 8 , Aug. 1997 Page(s): 858 –867.

[23] J. Oliensis, Y. Genc, "Fast and accurate algorithms for projective multi-image structure from motion", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 23 Issue: 6 , June 2001 Page(s): 546 –559.

[24] S.H. Or, K.H.Wong and T.T.Wong. "On Using Longuet Higgins Equation in Pose Estimation Framework by Lowe." Proceedings of the International Conference on Imaging Science, Systems, and Technology, pp.596-599, on 6 June 1999, Las Vegas USA.

[25] S.H. Or, K.H.Wong, T.K. Lao and T.T.Wong. "An image based pose approach to pose estimation", Invited conference paper, 1999 International Symposium on Signal Processing and Intelligent Symposium (ISSPIS'99), Nov. 26-28, 1999 Guangzhou, China.(R).

[26] S.H. Or, W.S. Luk, K.H. Wong, I.King, "An efficient iterative pose estimation algorithm", Image and Vision Computing Journal, Volume 16, Issue 5, pp.355-364, May 1998.

[27] C.J. Poelman, Kanade, T, "A paraperspective factorization method for shape and motion recovery," in *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI-19, No. 3, pp.206-218, March 1997.

[28] M. Pollefeys, Self-calibration and metric 3D reconstruction from uncalibrated image sequences, PhD. thesis, K.U.Leuven, 1999.

[29] M. Pollefeys, Tutorial on 3D Modeling from Images, June 2000.

[30] L. Quan and Z. Lan, "Linear N-Point Camera Pose Determination," in *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI-21, No. 8, pp.774-780, August 1999.

[31] G. Strang, *Introduction to Linear Algebra*, 2nd ed., Wellesley-Cambridge Press, 1998.

[32] P. Sturm B. Triggs, "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion", EECV 96.

[33] Z. Sun, M. Tekalp, A. Navab, and N. Ramesh, V., "Interactive optimization of 3D shape and 2D correspondence using multiple geometric constraints via POCS", Pattern Analysis and Machine Intelligence, IEEE Transactions on ,Volume: 24 Issue: 4 , April 2002 Page(s): 562 -569.

[34] R. Szeliski and S. B. Kang, "Recovering 3D, shape and motion from image streams using non-linear least squares", Journal of Visual Communication and Image Representation", vol. 5, No. 1, PP. 10--28", 1994.

[35] 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.

[36] R. Szeliski and SB Kang, "Recovering 3D, shape and motion from image streams using non-linear least squares", Journal of Visual Communication and Image Representation", vol. 5, No. 1, PP. 10--28", 1994.

[37] 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, 1992, pp. 137-154.

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

[39] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbonm, "Bundle Adjustment - A Modern Synthesis", In Proceedings of the International Workshop on Visual Algorithm: Thoery and Practice, P.P. 298-372, Corfu, Greece, Sept. 1999.

[40] E. Trucco and A. Verri, *Introductory techniues for 3-D compouter vision,* Prentice Hall , 1998.

[41] J. Weng, N. Ahuja and T.S. Huang, "Optimal motion and structure estimation Pattern Analysis and machine Intelligence", IEEE Transactions on ,Volume: 15 Issue: 9 , Sept. 1993 Page(s): 864 –884.

[42] K.H. Wong, S.H. Or, and M.M.Y Chang, "Pose tracking for virtual walk-through environment construction" Conference Proceeding on the International Conference on Inverse Problems & Numerics, City University of Hong Kong, January 9-12, 2002.

[43] Z. Zhang and G. Xu, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach,* Academic Publishers, 1996.

[44] Z. Zhang and Y. Shan. Incremental Motion Estimation through Local Bundle Adjustment. Technical Report MSR-TR-01-54, Microsoft Research, May 2001. (http://research.microsoft.com/~zhang/publications.htm#sec-techreports)

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

Prof. Wong Kin-hong received the 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