Application of Motion Vector in Live 3D Object Reconstruction

Renshu Gu, Jie Yuan, Sidan Du
School of Electronic Science and Engineering, Nanjing University
Nanjing, China
Email: gu_renshu@yahoo.cn, yuanjie@nju.edu.cn, coff128@nju.edu.cn

Abstract—**This paper applied a novel method to live 3D object reconstruction. Provided that static 3D reconstruction has been completed with basic frames, the proposed linear method calculates free 3D motion of a rigid-body in a video sequence, so as to display the moving object. The succinct method does not involve the fundamental matrix, and the point correspondence procedure in 3D reconstruction is reused. Least Mean Square decomposition is adopted to solve linear equation set and thus obtain motion vector. Furthermore, an iterative process enables the method to eliminate outliers. Sufficient experiments proved the validity and efficiency of the method.**

Keywords-3D motion; volumetric display; point correspondence; linear equation set; QR decomposition.

I. Introduction

Estimating rigid-body 3-D motion parameters from two or more images of an image sequence has been extensively studied for a long time in the field of computer vision. It can be tracked down to the 1970s, when early researches proved that three views are necessary to recover motion from an orthographic camera projection model, and that two views are enough to estimate motion from full perspective projection [1].

In 1989, Weng, Huang and Ahuja [2] gave a detailed discussion on motion estimation from two views of full perspective projection, and proposed the classic 8-point algorithm, namely to utilize the epipolar geometric constraint, estimate the basic matrix from feature points (matrix E in [2]), and then obtain 3D motion parameters from the basic matrix. From then on, to overcome its sensitivity to noise, a large number of improved methods were proposed, including American researcher Hartley's Improved 8-point Algorithm which standardizes 2D data. Except from methods based on point feature, researchers also proposed various methods based on line feature [4] [5], point feature and line feature combined, optic flow [6] [7] and methods using multiple frames and iterative algorithm [8] [9] [10].

Nevertheless, the methods mentioned above are integrated methods of 3-D motion estimation from two (or more) images without any prior knowledge of the object. In general, how to apply the epipolar geometric constraint is a key problem. As for methods based on point feature, recent works focus on error analysis and control (e.g., [1] [11] [12]), or on better estimation of the fundamental matrix [13] [14] [15], all of which inevitably engage the fundamental matrix. The most recent and relevant work in [16] described 2-frame recovery of structure and motion using uncalibrated cameras. This is somewhat similar to Han-Kanade's method in [17].

To perform live 3D object reconstruction, we intend to display moving 3D object in accordance with real pictures from live videos. However, adopting Han-Kanade reconstruction to video sequences over time would be too time-consuming. In our scheme, static 3D reconstruction is performed only at intervals, while motion vector is calculated and added to the static model at any time during the intervals. This paper proposed a linear method to calculate motion alone instead of both motion and structure, added the motion to the original 3D object, and then displayed it from whatever view the spectator prefer.

Once static 3D reconstruction has been completed, we know the 3D coordinates corresponding to 2D feature points in image t0, and we also know new 2D coordinates of those points in image t1 through image matching. It is convenient for us to match images, since it is a step in 3D reconstruction. With these data we know, 3D motion parameters can be calculated already, and there is no need to use the integrated methods in the above articles. In other words, we have obtained enough information of 2D mapping 3D object so there is no need for the epipolar constraint. In theory, with 3D reconstruction completed, 3 point pairs from 2 views can recover 3D motion when motion is small. This paper gives Approximate Algorithm applicable to small motion, and also Universal Algorithm applicable to any motion. Frame rate of a video is greater than 25 fps, and motion between two close frames are quite small; therefore, if the two images under estimation t0, t1 are two successive (or close) frames in the video sequence, the Approximate Algorithm is adequate enough for motion estimation. Part Ⅲ of this paper demonstrates experiments on both algorithms, and compares them with work in [17] (similar to [16]).

II. Theory and Algorithm

A. *Camera model: affine camera*

2D coordinates and 3D coordinates obey:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = P \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \tag{1}$$

where $P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is a matrix that determines a way of linear mapping from 3D space to 2D plane.

It should be noted that affine camera is the approximation of real cameras. It is applicable only when the depth changes of the interested object can be neglected compared to its depths.

B. *Motion model of rigid-body*

In 3D space, motion of rigid-body between two positions can be decomposed to rotation and translation. Rotation of rigid-body in 3D space can be described by a 3 x 3 matrix R, and translation can be described by a 3 x 1 matrix T.

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Let 3D coordinates of any feature point in image t0 be (X_1, Y_1, Z_1), and coordinates of this point in image t1 be (X_1', Y_1', Z_1'). The motion between t0 and t1 can be represented by:

$$\begin{bmatrix} X_1' \\ Y_1' \\ Z_1' \\ 1 \end{bmatrix} = M \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \qquad (2)$$

where the motion matrix is

$$M = \begin{bmatrix} R & T \\ O & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

Note that an important property of rigid-body is, under a certain coordinate system, any point of the body has the same motion, i.e., the motion matrix. That is, every point pair obeys equation (2).

Although rotation matrix has 9 elements, it is an orthogonal matrix and subjects to 6 independent constraints:

$$\sum_{j=1}^{3} r_{ij}^2 = 1 (i = 1,2,3) \quad \sum_{k=1}^{3} r_{ik} r_{jk} = 0 (i, j = 1,2,3) \qquad (4)$$

Therefore, there are only 3 independent parameters. Among various ways of expressing R by 3 independent parameters, the following 2 ways are commonly used:

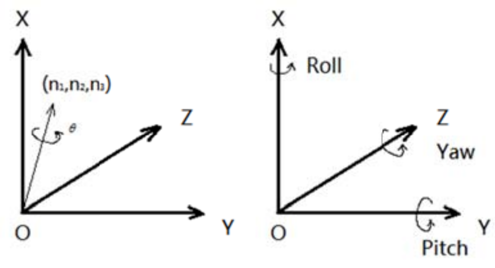
(i) Axis-angle representation

$$\begin{cases} r_{11} = \cos\theta + (1-\cos\theta) n_1^2 \\ r_{12} = (1-\cos\theta) n_1 n_2 - (\sin\theta) n_3 \\ r_{13} = (1-\cos\theta) n_1 n_3 + (\sin\theta) n_2 \\ r_{21} = (1-\cos\theta) n_1 n_2 + (\sin\theta) n_3 \\ r_{22} = \cos\theta + (1-\cos\theta) n_2^2 \\ r_{23} = (1-\cos\theta) n_2 n_3 - (\sin\theta) n_1 \\ r_{31} = (1-\cos\theta) n_3 n_1 - (\sin\theta) n_2 \\ r_{32} = (1-\cos\theta) n_3 n_2 + (\sin\theta) n_1 \\ r_{33} = \cos\theta + (1-\cos\theta) n_3^2 \end{cases} \qquad (5)$$

The 3D vector from the origin $(0,0,0)$ to (n_1, n_2, n_3) represents the axis the rigid-body rotates around. The angle it rotates is θ.

(ii) Roll, Pitch and Yaw representation

$$R = \begin{bmatrix} \cos\theta_y \cos\theta_z & -\cos\theta_y \sin\theta_z & \sin\theta_y \\ \sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z & -\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & -\sin\theta_x \cos\theta_y \\ -\cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z + \sin\theta_x \cos\theta_z & \cos\theta_x \cos\theta_y \end{bmatrix} \qquad (6)$$



(a) left: Angle-axis way of expressing 3D motion
(b) right: Roll-Pitch-Yaw way of expressing 3D motion
Figure 1. Two common way of expressing 3D motion

C. *Solve the motion problem*

Motion estimation can be completed in 2 steps.

First step: convert the problem of estimating motion to solving equation set.

From $\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = P \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$, $\begin{bmatrix} x_1' \\ y_1' \\ 1 \end{bmatrix} = P \begin{bmatrix} X_1' \\ Y_1' \\ Z_1' \\ 1 \end{bmatrix}$ and $\begin{bmatrix} X_1' \\ Y_1' \\ Z_1' \\ 1 \end{bmatrix} = M \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$

we have:

$$\begin{bmatrix} x_1' \\ y_1' \\ 1 \end{bmatrix} = PM \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \qquad (7)$$

Every point pair gives 2 equations:

$$\begin{cases} x_1' = p_{14} + p_{11}t_1 + p_{12}t_2 + p_{13}t_3 + X_1(p_{11}r_{11} + p_{12}r_{21} + p_{13}r_{31}) \\ \quad + Y_1(p_{11}r_{12} + p_{12}r_{22} + p_{13}r_{32}) + Z_1(p_{11}r_{13} + p_{12}r_{23} + p_{13}r_{33}) \\ y_1' = p_{24} + p_{21}t_1 + p_{22}t_2 + p_{23}t_3 + X_1(p_{21}r_{11} + p_{22}r_{21} + p_{23}r_{31}) \\ \quad + Y_1(p_{21}r_{12} + p_{22}r_{22} + p_{23}r_{32}) + Z_1(p_{21}r_{13} + p_{22}r_{23} + p_{23}r_{33}) \end{cases} \qquad (8)$$

If motion is small enough for small-angle approximation, and if the Roll-Pitch-Yaw representation is adopted, the rotation matrix is simplified to:

$$R=\begin{bmatrix}1 & -n_3\theta & n_2\theta\\ n_3\theta & 1 & -n_1\theta\\ -n_2\theta & n_1\theta & 1\end{bmatrix}=\begin{bmatrix}1 & -\phi_3 & \phi_2\\ \phi_3 & 1 & -\phi_1\\ -\phi_2 & \phi_1 & 1\end{bmatrix} \quad (9)$$

where $(n_1\theta)^2+(n_2\theta)^2+(n_3\theta)^2=\theta^2$ .i.e. $\phi_1{}^2+\phi_2{}^2+\phi_3{}^2=\theta^2$ (10)

Rewrite the equation set (8):

$$\begin{bmatrix}x_1'-p_{14}-X_1p_{11}-Y_1p_{12}-Z_1p_{13}\\ y_1'-p_{24}-X_1p_{21}-Y_1p_{22}-Z_1p_{23}\end{bmatrix}=\begin{bmatrix}Y_1p_{13}-Z_1p_{12} & Z_1p_{11}-X_1p_{13} & X_1p_{12}-Y_1p_{11} & p_{11} & p_{12} & p_{13}\\ Y_1p_{23}-Z_1p_{22} & Z_1p_{21}-X_1p_{23} & X_1p_{22}-Y_1p_{21} & p_{21} & p_{22} & p_{23}\end{bmatrix}\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ t_1\\ t_2\\ t_3\end{bmatrix}$$

Multiple point pairs can give equation set:

$$\begin{bmatrix}x_1'-p_{14}-X_1p_{11}-Y_1p_{12}-Z_1p_{13}\\ y_1'-p_{24}-X_1p_{21}-Y_1p_{22}-Z_1p_{23}\\ .\\ .\\ x_n'-p_{14}-X_np_{11}-Y_np_{12}-Z_np_{13}\\ y_n'-p_{24}-X_np_{21}-Y_np_{22}-Z_np_{23}\end{bmatrix}=\begin{bmatrix}Y_1p_{13}-Z_1p_{12} & Z_1p_{11}-X_1p_{13} & X_1p_{12}-Y_1p_{11} & p_{11} & p_{12} & p_{13}\\ Y_1p_{23}-Z_1p_{22} & Z_1p_{21}-X_1p_{23} & X_1p_{22}-Y_1p_{21} & p_{21} & p_{22} & p_{23}\\ & . & & & & \\ & . & & & & \\ Y_np_{13}-Z_np_{12} & Z_np_{11}-X_np_{13} & X_np_{12}-Y_np_{11} & p_{11} & p_{12} & p_{13}\\ Y_np_{23}-Z_np_{22} & Z_np_{21}-X_np_{23} & X_np_{22}-Y_np_{21} & p_{21} & p_{22} & p_{23}\end{bmatrix}\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ t_1\\ t_2\\ t_3\end{bmatrix} \quad (11)$$

The first matrix on the right side of the above equation is written down as A, which has a rank of no greater than 5. This is due to the limitation of single camera, because the camera cannot give information of depth along its optical axis. Below is the *proof*:

Observe matrix A. We can see the odd rows of the right three columns are identical to each other, and so are the even rows. Elementary row transformation operations are applied to A. Eliminate the right three columns of Row 2i-1 with the first row, and those of Row 2i with the second row（i=2,3⋯n）:

$$\begin{bmatrix}Y_1p_{13}-Z_1p_{12} & Z_1p_{11}-X_1p_{13} & X_1p_{12}-Y_1p_{11} & p_{11} & p_{12} & p_{13}\\ Y_1p_{23}-Z_1p_{22} & Z_1p_{21}-X_1p_{23} & X_1p_{22}-Y_1p_{21} & p_{21} & p_{22} & p_{23}\\ & \vdots & & 0 & 0 & 0\\ & \vdots & & & \vdots & \\ & \vdots & & 0 & 0 & 0\\ Y_np_{13}-Z_np_{12} & Z_np_{11}-X_np_{13} & X_np_{12}-Y_np_{11} & 0 & 0 & 0\\ Y_np_{23}-Z_np_{22} & Z_np_{21}-X_np_{23} & X_np_{22}-Y_np_{21} & 0 & 0 & 0\end{bmatrix} \quad (12)$$

Of the obtained matrix (12), the right three columns as a sub-matrix has a rank of 2 at most, because there are only 2 nonzero rows; the right three columns as a sub-matrix has a rank of 3 at most. So maxim rank of matrix (12) is 5. Elementary row transformation operations do not change the rank of matrix, so matrix A has a rank of 5 at most.

From the above we know motion estimation of rigid-body need at least 2 cameras. Let the projection matrix of the second camera be:

$$P=\begin{bmatrix}p_{11}' & p_{12}' & p_{13}' & p_{14}'\\ p_{21}' & p_{22}' & p_{23}' & p_{24}'\\ 0 & 0 & 0 & 1\end{bmatrix}$$

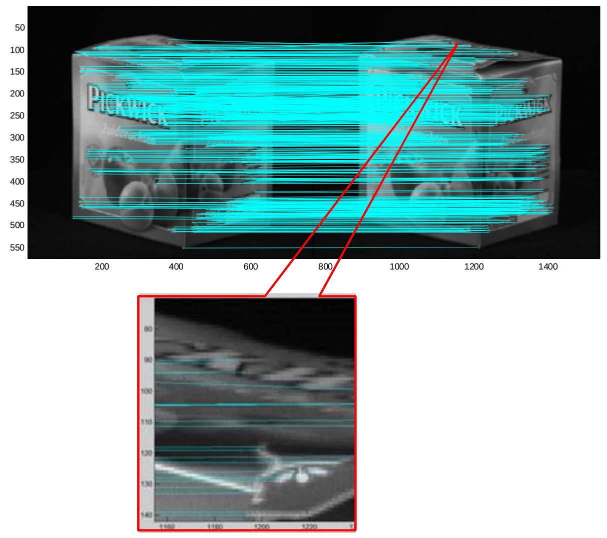
We have:

$$\begin{bmatrix}x_1'-p_{14}-X_1p_{11}-Y_1p_{12}-Z_1p_{13}\\ y_1'-p_{24}-X_1p_{21}-Y_1p_{22}-Z_1p_{23}\\ .\\ .\\ x_n'-p_{14}'-X_np_{11}'-Y_np_{12}'-Z_np_{13}'\\ y_n'-p_{24}'-X_np_{21}'-Y_np_{22}'-Z_np_{23}'\end{bmatrix}=\begin{bmatrix}Y_1p_{13}-Z_1p_{12} & Z_1p_{11}-X_1p_{13} & X_1p_{12}-Y_1p_{11} & p_{11} & p_{12} & p_{13}\\ Y_1p_{23}-Z_1p_{22} & Z_1p_{21}-X_1p_{23} & X_1p_{22}-Y_1p_{21} & p_{21} & p_{22} & p_{23}\\ & . & & & & \\ & . & & & & \\ Y_np_{13}'-Z_np_{12}' & Z_np_{11}'-X_np_{13}' & X_np_{12}'-Y_np_{11}' & p_{11}' & p_{12}' & p_{13}'\\ Y_np_{23}'-Z_np_{22}' & Z_np_{21}'-X_np_{23}' & X_np_{22}'-Y_np_{21}' & p_{21}' & p_{22}' & p_{23}'\end{bmatrix}\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\\ t_1\\ t_2\\ t_3\end{bmatrix}$$

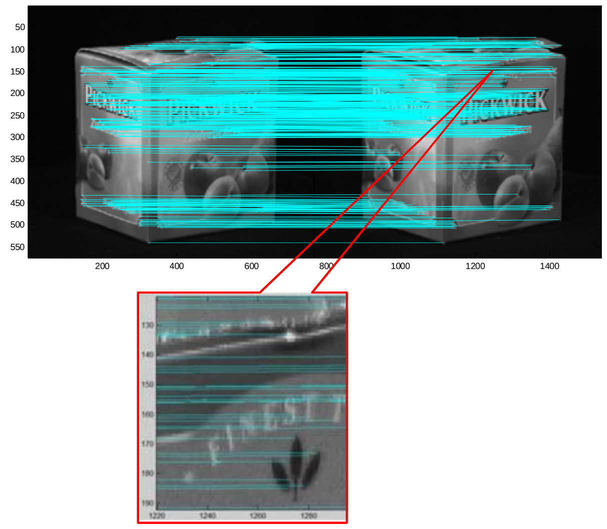
(13)

The matrix on the left is denoted as b . The matrix on the right is denoted as A , x successively. The rank of A can be 6 at most now. Thus, estimating motion is converted to solving equation set

$$b=Ax \quad (14)$$



(a) Matched point pairs in image t0 (left) and t1 (right) from camera 1



(b) Matched point pairs in image t0 (left) and t1 (right) from camera 2
Figure 2. Matched point pairs in image t0 and t1 from camera1 and 2

Use m point pairs from camera 1 (or view 1) and k point pairs from camera 2 (or view 2), on the condition that $n = m + k, (m, n \geq 1)$ ($m, n \geq 1$). In theory, 3 point pairs from two views are enough for solving the equation set. When n$\geq$3, we need to solve an equation set to derive the results. After 3D reconstruction with image t0 and image matching between image t0&t1, we can obtain a large number of feature point pairs. To exploit the information we have, we first use all point pairs to estimate motion. To enhance the algorithm's robustness, use the initial motion matrix for reprojection, compare the results with real 2D coordinates in image t1, and eliminate outliers whose errors are larger than threshold 10 (experiment shows error is normally less than 5, if error >10 appears, it is probably an outlier). Appearance of outliers may be the result of wrong matching.

Second step : solve the equation set by least mean square error principle.

To solve the equation set by least mean square error principle, QR decomposition method is adopted in this paper.

Because of the existence of error, $Ax = b + \varepsilon$. The problem equates solving x that minimizes norm $\|\varepsilon\|_2^2$. We can find Q that obeys $QA = \begin{pmatrix} R \\ O \end{pmatrix}$, where Q is an orthogonal matrix and R is a nonsingular upper triangular matrix.

$$\|Ax - b\|_2^2 = \|Q(Ax - b)\|_2^2 = \|QAx - Qb\|_2^2 = \left\| \begin{bmatrix} R \\ O \end{bmatrix} - Qb \right\|_2^2$$

Write $Qb = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, then we have $\|Ax - b\|_2^2 = \left\| \begin{bmatrix} Rx - b_1 \\ b_2 \end{bmatrix} \right\|_2^2$. It is a column vector, and thus $= \|Rx - b_1\|_2^2 + \|b_2\|_2^2$. Notice that $\|b_2\|_2^2$ is constant; therefore, the original problem of minimizing $\|Ax - b\|_2^2$ converts to minimizing $\|Rx - b_1\|_2^2$, namely $Rx - b_1$ =0.

Thus, we have obtained x vector. For equation set (13), x= $(\phi_1, \phi_2, \phi_3, t_1, t_2, t_3)^T$

D. Universal Algorithm

If there is no small-angle approximation, we can write equation set as follows:

$$\begin{bmatrix} x_1'-p_{14} \\ y_1'-p_{24} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n'-p_{14}' \\ y_n'-p_{24}' \end{bmatrix} = \begin{bmatrix} X_1p_{11} & X_1p_{12} & X_1p_{13} & Y_1p_{11} & Y_1p_{12} & Y_1p_{13} & Z_1p_{11} & Z_1p_{12} & Z_1p_{13} & p_{11} & p_{12} & p_{13} \\ X_1p_{21} & X_1p_{22} & X_1p_{23} & Y_1p_{21} & Y_1p_{22} & Y_1p_{23} & Z_1p_{21} & Z_1p_{22} & Z_1p_{23} & p_{21} & p_{22} & p_{23} \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & \cdot \\ X_np_{11}' & X_np_{12}' & X_np_{13}' & Y_np_{11}' & Y_np_{12}' & Y_np_{13}' & Z_np_{11}' & Z_np_{12}' & Z_np_{13}' & p_{11}' & p_{12}' & p_{13}' \\ X_np_{21}' & X_np_{22}' & X_np_{23}' & Y_np_{21}' & Y_np_{22}' & Y_np_{23}' & Z_np_{21}' & Z_np_{22}' & Z_np_{23}' & p_{21}' & p_{22}' & p_{23}' \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \\ r_{12} \\ r_{22} \\ r_{32} \\ r_{13} \\ r_{23} \\ r_{33} \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (15)$$

Use m point pairs from camera 1 (or view 1) and k point pairs from camera 2 (or view 2), on the condition that $n = m + k, (m, n \geq 1)$. The method of solving equation set is the same as Approximate Algorithm. For equation set (15), $x = (r_{11}, r_{21}, r_{31}, r_{12}, r_{22}, r_{32}, r_{13}, r_{23}, r_{33}, t_1, t_2, t_3)^T$.

For Universal Algorithm, to ensure A has full rank, at least 4+4=8 pairs of feature points are needed.

III. Experimental Validation

Validity of the proposed algorithm was verified by MATLAB experiments. We used images caught by 2 cameras.

Below are the steps of the algorithm.

```
------------------------------------------------------------------
1. {3D Reconstruction}
2. {Match image t0 and t1. Obtain 2D coordinates of feature points in t1.}
3. {Binocular Vision: 3D reconstruction and image matching for 2 cameras}
4. {Initialization: input m pairs of camera1 and k pairs of camera2)}
5. {Turn motion estimation into solving equation set. Every point pair
gives 2 equations. }
6. {Solve equation set $b = Ax$ under LSQ criterion (by QR decomposition).}
7. {Reproject with derived M, compare 2D results to real 2D coordinates.}
    If (Errors of all points are under threshold) then
    {Eliminate point pairs whose errors exceeds threshold.
     Go to 5, repeat 5, 6, and 7}
    Else {Give final results}
8. {Live Display of 3D Object}
------------------------------------------------------------------
```

A. 5 ° around axis x (both algorithms)

A box was known to have rotated an angle of 5° . For simplicity, the motion in our experiment only had rotation component. That is, t1, t2 and t3 in the translation matrix should be all zeros in theory, and the emphasis of result should be the rotation matrix. First, the Approximate Algorithm was tested. Second, the Universal Algorithm, which is rigorous, was tested. According to (6), $\theta_x = \arctan(-r_{23} / r_{33})$.

We adopted two images shown in Figure 2. Reconstruct image t0 in 3D space, and match image t0 and t1, we obtained 318 point pairs from camera 1in all. Similarly, we obtained 246 point pairs from camera 2 in all.

Experiments with different input numbers of point pair n were conducted. Input point pairs were results of equal-interval sampling from all point pairs. The calculated t1, t2, t3 were all approximately zeros; calculated θ_x is demonstrated in Figure 3.

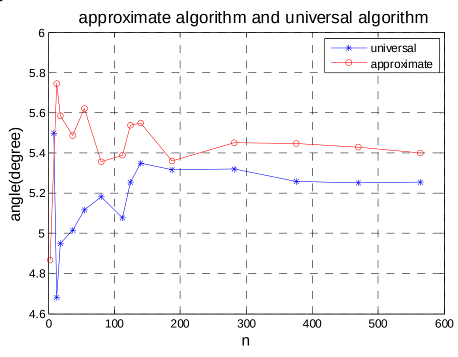


Figure 3. Experiments with different number of input point pairs

As shown, when input point pairs are barely enough for calculation, results are greatly influenced by the choice of input data, and are unreliable; however, when there are enough input data, both the Approximate Algorithm and the Universal Algorithm can give stable results (5.4 and 5.3). When input number of point pairs is greater than 200 (two cameras), for the Approximate Algorithm, fluctuations of results do not exceed 0.05 / 5.4=0.9%, and for the Universal Algorithm, fluctuations are within 0.06 / 5.2559= 1.2%.

It was also verified that reprojection errors were almost all less than 2 pixels along any axis in the image for 318+246=564 point pairs. Figure 4 shows an example of reprojection errors along x axis. In addition, the algorithm can eliminate outliers, whose reprojection errors are greater than threshold 10, thus guaranteeing stable and robust results.

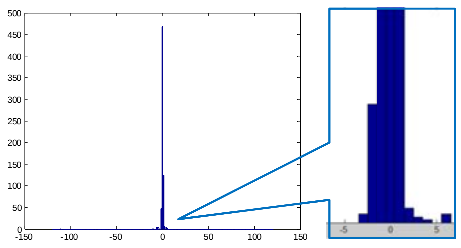
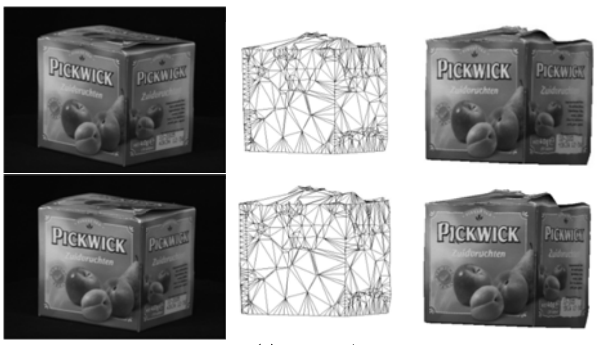


Figure 4. reprojection errors distribution

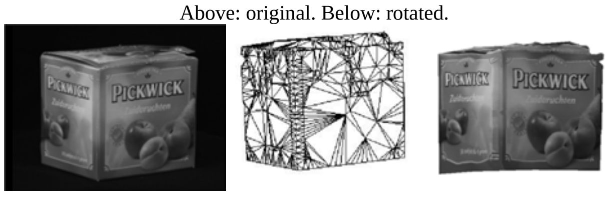
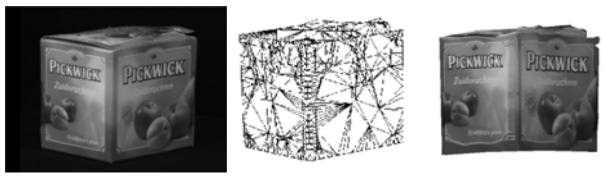
B. 10° around axis x (Universal Algorithm only)

The box was known to have rotated about 10° around axis x in Figure 1. Similar to experiment 1, there was merely rotation component of the motion. Only the Universal Algorithm can be applied. 210 point pairs from camera 1 and 168 pairs from camera 2 were used. The calculated angle was 10.6700.

C. Live display of 3D object



(a) camera 1
Above: original. Below: rotated.

(b) camera 2
Above: original. Below: rotated.
Figure 5. comparison of real object and displayed object

Our process of live display of 3D object is as follows: (i) Add the calculated motion to the original 3D object which has been reconstructed, i.e. apply equation (2) to all points on the object. (ii) Connect all points in meshes to obtain the surface. (iii) Attach the texture of the original object to the meshed surface.

Figure 5 shows 10° rotation of real object and displayed object from 2 views in experiment B. (a) presents the pictures from camera 1 and, and (b) presents those from camera 2. The left column shows real pictures shoot by corresponding cameras. The middle column shows 3D object grids before attaching texture. The right column shows displayed object with texture attached.

As shown, from both views, the displayed object accords with its prototype in real world.

3D object can be displayed in free-angle as long as the original object is entirely reconstructed. We can reconstruct the object using 3 frames of a video sequence, and display any frame within certain range of the video sequence. The range is decided by the maximum angle of image matching.

D. Comparison with recent work and merits discussion

To further study accuracy of the proposed method, control experiments were conducted from many different viewpoints. We adopted Han-Kanade method [17] to reconstruct a stereo-object as well as recovering motion, and then adopted the proposed method to calculate motion using same input points. As shown in Figure 6, our method is comparable to Han-Kanade method in terms of precision.

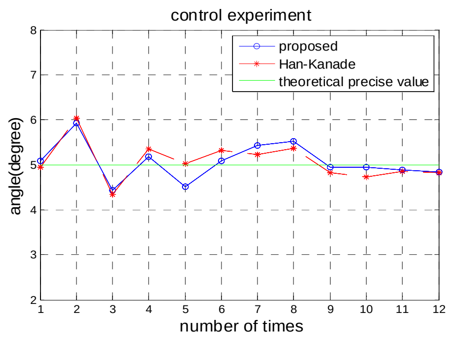


Figure 6. comparison with Han-Kanade method (same input point pairs)

Thorough recovery of 3D structure and motion from uncalibrated cameras is quite time-consuming. On a personal computer of 2.4GHz master frequency, it cost 5~10 minutes for one time of 3-frame reconstruction. Stereo reconstruction from 5 views will take much more time. Moreover, we cannot reduce the number of cameras, since fewer cameras will bring more difficulties on image matching because of wider baseline between every two camera. This directly affects 3D reconstruction. The proposed method, on the other hand, costs less than one second with same number of input points. (Same to Han-Kanade method, this does not include the time for display, making it a fair comparison) Evidently, it would save a great deal of time in case of long video sequence if we recover motion using proposed method, rather than recover structure and motion simultaneously over time.

E. *Simple error analysis*

Errors include: (1) Quantization error. Quantization error exists throughout the whole process of computer vision including motion calculation problem in this paper. (2) 3D reconstruction errors, including multi-view correspondence error, etc. As input of motion vector calculation, 3D reconstruction with errors will lead to errors in the motion vector. (3) Error in matching image t0 and t1. Finite pixel resolution may introduce error, and matching algorithm could also influence accuracy in 2D coordinate of image t1.

Additionally, camera calibration and computational accuracy have little effects on results of our method. In our method, 3D reconstruction is based on uncalibrated cameras, and motion calculation does not involve calibration either. Computational accuracy of MATLAB is more than 40 decimal places; therefore, truncation error has no influence on our results.

IV. CONCLUSION AND FUTURE WORK

Several conclusions can be drawn from the experiment results.

Firstly, the correctness and feasibility of the proposed algorithm are verified. For the Approximate Algorithm, exactly 3 point pairs are needed for motion estimation. Of course, the result depends largely on the selected 3 point pairs, and is too sensitive and unstable.

Secondly, the algorithm is stable and robust when there is enough input data, as results are obtained on a unanimous ground. The algorithm can eliminate outliers to a certain extent.

Thirdly, since the algorithm merely calculates motion vector but not simultaneously recover motion and structure, it is quite succinct. Time cost depends on the number of point pairs. For a typical experiment with about 200 point pairs each camera, both the Universal Algorithm and the Approximate Algorithm cost less than a few second. Thus, static 3D reconstruction can be performed only at intervals, while motion vector is calculated and added to the static model at any time during the intervals. Compared to conducting 3D reconstruction over time, this scheme will save much time.

Future work will concentrate on the following several aspects:

Selecting a proper interval between every two times of motion vector calculation: strike the balance between time consumption and necessity;

Improving the continuity of real-time display, perhaps by interpolating;

Testing cases when motion calculation is limited by wide-baseline image matching， in order to find out the limitation of application.

V. ACKNOWLEDGEMENT

This paper is supported by the Fundamental Research Funds for the Central Universities, Number: 1107021051, and National Natural Science Funds of Jiangsu Province of China, Number: BK2010386.

REFERENCES

[1] T. Papadimitriou, M. G. Strintzis, and M. Roumeliotis, “Robust Estimation of Rigid Body 3-D Motion Parameters Based on Point Correspondences”, IEEE Trans. on Circuits And Systems for Video Technology, 2000, pp. 541-549.

[2] J. Weng, T. S. Huang, N. Ahuja, “Motion and Structure From Two Perspective Views: Algorithms, Error Analysis, and Error Estimation”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989, pp. 451-476.

[3] R. Hartley, “In Defense of the 8-point Algorithm”, Proc. 5th ICCV, Cambridge, USA, 1995, pp. 1064-1070.

[4] H. Ebrahimnezhad, H. G. Robust, “Motion From Space Curves and 3D Reconstruction From Multiviews Using Perpendicular Double Stereo Rigs”, Image and Vision Computing, 2008, pp. 1397–1420.

[5] J. M. M. Montiel, J.D. TardoHs, and L. Montano, “Structure and Motion From Straight Line Segments”, Patten Recognition, 2000, pp. 1295-1307.

[6] E. G. Steinbach, P.Eisert, and B. Girod, “Model-based 3D Shape and Motion Estimation Using Sliding Textures”, Proc. Vision Modeling and Visualization Conference, 2001, pp. 375-382.

[7] J. Neumann, C. Fermuller, and Y. Aloimonos, “Polydioptric camera design and 3D motion estimation”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, volume 2, 2003, pp. 294-301.

[8] J. W. Roach and J. K. Aggarwal, “Determining the Movement of Objects From A Sequence of Images”, IEEE Trans. on Pattern Anal. Machine Intell, 1980, pp. 554–562.

[9] J. Oliensisa, “Multi Frame Structure from Motion Algorithm under Perspective Projection”, Int J. Comput Vis, 1999, pp. 163–192.

[10] P. Sand and S. Teller, “Particle Video: Long-Range Motion Estimation Using Point Trajectories”, Int J. Comput. Vis, 2008, pp. 72–91.

[11] B. Matei and P. Meer, “A General Method for Errors-in-Variables Problems in Computer Vision,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2000, pp. 18–25.

[12] P. Firoozfam and S. Negahdaripour, “Theoretical Accuracy Analysis of N-Ocular Vision Systems for Scene reconstruction, Motion Estimation, and Positioning Problems in Computer Vision”, Proc. 2nd Int. Symposium on 3D Data Processing, Visualization, and Transmission, 2004, pp. 888-895.

[13] Y. Sheikh, A. Hakeem, and M. Shah, “On the Direct Estimation of The Fundamental Matrix”. Proc. IEEE Conf. Comput Vis. Patt. Rec., 2007, pp. 1–7.

[14] P. Chen, “Why Not Use the Levenberg–Marquardt Method For Fundamental Matrix Estimation”, IET Comp. Vis., Vol. 4, Iss. 4, 2010, pp. 286–294.

[15] H. P. Wu and S. H. Chang, “Fundamental Matrix of Planar Catadioptric Stereo Systems”, IET Comput. Vis., Vol. 4, Iss. 2, 2010, pp. 85–104.

[16] R. Szeliski, “Computer Vision: Algorithms and Applications”, Chapter 7, 2010, pp. 343-374.

[17] M. Han and T. Kanade, “Creating 3D Models with Uncalibrated Cameras”, Proc. IEEE Computer Society Workshop on the Application of Computer Vision, 2000, pp. 178-185.