Internal report number: khw_irep_070509
The CSE Dept., The Chinese University of Hong Kong, Internal report, 2007-May-9

A robust head pose tracking system based on multiple cameras

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

Michael Ming Yuen Chang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: mchang@ie.cuhk.edu.hk
*Kin Hong Wong, Ying Kin Yu, Siu Hang Or are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {khwong,ykyu,shor}@cse.cuhk.edu.hk
*corresponding author

Abstract

Tracking human head pose has many applications such as building input devices for game players and for the disabled. Traditional vision methods detect the image of the head to deduce its pose. Our approach is the reverse, we attach cameras to the head so head pose tracking becomes camera pose tracking. The main problem with camera centered vision based pose tracking is that, if one camera is used, translation and rotation will produce very similar effects on the images hence the pose cannot be correctly detected; this problem is called the translation/rotation ambiguity problem. The novelty of our approach is that we attach multiple cameras to the head to tackle the ambiguity problem and obtain very accurate pose result. As the costs, sizes and weights of cameras are reducing quickly, this is in fact a very practical approach for making a head pose input device. Experiments on simulation and real data have shown the effectiveness of the method.

1. Introduction

Most traditional head motion tracking systems use the images of the head and computer vision method to find the pose [1][3][4][5][6][9]. Part of the difficulty is that the image of the head only occupies a portion of the whole image and reliable face features are difficult to find. Hence the system cannot be very robust. We take a different and novel approach that the camera is not looking at the head but mounted on the head to look at surrounding patterns. It has the advantage that all the information in the images is used for tracking, and head tracking becomes the same as camera pose tracking. However, it is well known that the camera centered pose estimation method by one camera suffers from a problem called translation/rotation ambiguity [7],[8]. For example, by just looking at the image of a moving camera, it is not easy to find whether the 2-D motion is created by a rotation around the vertical axis or translation along the horizontal axis. Some researchers have used multiple cameras for pose estimation [10][12][13], however, our contributions are: (1) give an analysis of why this ambiguity happens and (2) solve the translation/rotation ambiguity problem by using a pair of back-to-back cameras and create a head pose input system as in Figure 1. As the costs, sizes, and weights of the cameras are dropping fast, it is in fact a practical and feasible scheme for head tracking.

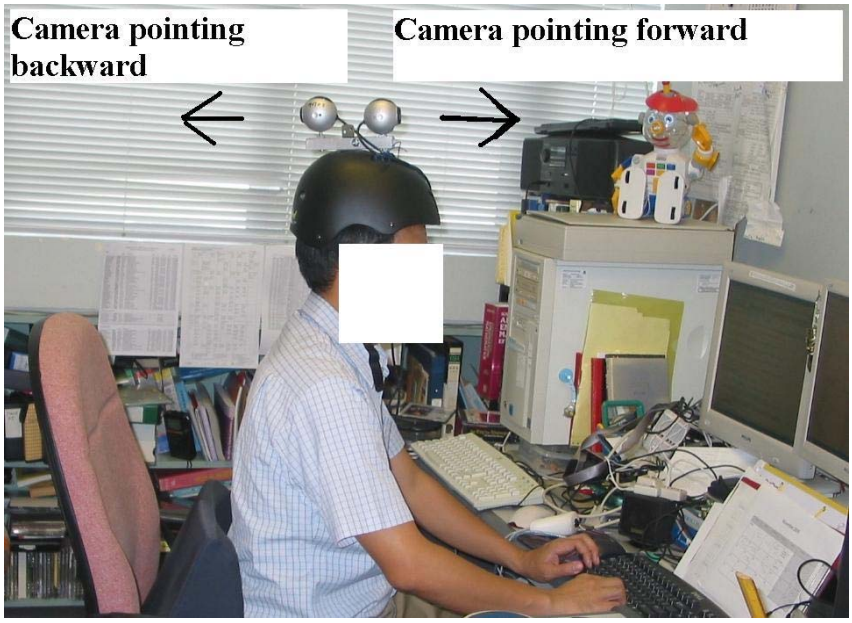

Figure 1a: A user with the head mount with one camera pointing at the forward direction (right), another pointing at the backward direction (left).

Figure 1b: Close-up of the head mount with two cameras

The organization of the paper is follows: in section 2, we study the ambiguity problem for a single camera; we then extend

our work to include multiple cameras and propose a method to remove the ambiguity. In section 3, we describe the experimental results, and in section 4 we conclude our work with a discussion.

2. Theory and Algorithm

2.1. The reference camera *k*=1

Using the first camera *k*=1 as the reference (the world coordinate coincides with the camera coordinate of the camera *k*=1. Since *k*=1 is the reference, so the index *k* is not shown here for simplicity. $\boldsymbol{P}_{ij}$ is the 3D position of the given model feature point $\boldsymbol{M}_i$ observed by the camera (*k*=1) at the j^{th} frame.

$$\boldsymbol{P}_{ij} = \begin{bmatrix} \boldsymbol{X}_{ij} & \boldsymbol{Y}_{ij} & \boldsymbol{Z}_{ij} \end{bmatrix}^T = \boldsymbol{R}_j(\boldsymbol{M}_i - \boldsymbol{d}_j),$$

$\boldsymbol{R}_j$ and $\boldsymbol{d}_j$ are rotational matrix and the translation vector, respectively, of the camera at the j^{th} frame; and Let $\boldsymbol{p}_j = \begin{bmatrix} \boldsymbol{t}_X & \boldsymbol{t}_Y & \boldsymbol{t}_Z & \boldsymbol{\theta}_X & \boldsymbol{\theta}_Y & \boldsymbol{\theta}_Z \end{bmatrix}^T$ be the pose vector of the camera at the j^{th} frame, where $\boldsymbol{d}_j = \begin{bmatrix} \boldsymbol{t}_X & \boldsymbol{t}_Y & \boldsymbol{t}_Z \end{bmatrix}^T$ and $\begin{bmatrix} \boldsymbol{\theta}_X & \boldsymbol{\theta}_Y & \boldsymbol{\theta}_Z \end{bmatrix}^T$ are the rotational angles that produce $\boldsymbol{R}_j$.

If $\boldsymbol{f}$ is the focal length of the camera the image coordinates of the feature are

$$\boldsymbol{x}_{ij} = \boldsymbol{f}\frac{\boldsymbol{X}_{ij}}{\boldsymbol{Z}_{ij}}, \quad \text{and} \quad \boldsymbol{y}_{ij} = \boldsymbol{f}\frac{\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}}$$

Set $\boldsymbol{f} = 1$, the change of pose $\boldsymbol{\delta p}$ will result in change of the image features $\begin{bmatrix} \boldsymbol{\delta x}_{ij} & \boldsymbol{\delta y}_{ij} \end{bmatrix}^T$, it is related by

$$\begin{bmatrix} \boldsymbol{\delta x}_{ij} \\ \boldsymbol{\delta y}_{ij} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial \boldsymbol{x}_{ij}}{\partial \boldsymbol{X}_{ij}}\frac{\partial \boldsymbol{X}_{ij}}{\partial \boldsymbol{t}_X} + \frac{\partial \boldsymbol{x}_{ij}}{\partial \boldsymbol{Z}_{ij}}\frac{\partial \boldsymbol{Z}_{ij}}{\partial \boldsymbol{t}_X}\right) & \cdots \\ \left(\frac{\partial \boldsymbol{y}_{ij}}{\partial \boldsymbol{Y}_{ij}}\frac{\partial \boldsymbol{Y}_{ij}}{\partial \boldsymbol{t}_X} + \frac{\partial \boldsymbol{y}_{ij}}{\partial \boldsymbol{Z}_{ij}}\frac{\partial \boldsymbol{Z}_{ij}}{\partial \boldsymbol{t}_X}\right) & \cdots \end{bmatrix} \boldsymbol{\delta p} \qquad (1)$$

$$= \boldsymbol{j}_i \, \boldsymbol{\delta p}$$

where $\boldsymbol{j}_i = \begin{bmatrix} \frac{1}{\boldsymbol{Z}_{ij}} & 0 & -\frac{\boldsymbol{X}_{ij}}{\boldsymbol{Z}_{ij}^2} \\ 0 & \frac{1}{\boldsymbol{Z}_{ij}} & -\frac{\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}^2} \end{bmatrix} \begin{bmatrix} -\boldsymbol{R}_j & \begin{matrix} 0 & -\boldsymbol{Z}_{ij} & \boldsymbol{Y}_{ij} \\ \boldsymbol{Z}_{ij} & 0 & -\boldsymbol{X}_{ij} \\ -\boldsymbol{Y}_{ij} & \boldsymbol{X}_{ij} & 0 \end{matrix} \end{bmatrix}$

and $\boldsymbol{\delta p} = \begin{bmatrix} \boldsymbol{\delta t}_X & \boldsymbol{\delta t}_Y & \boldsymbol{\delta t}_Z & \boldsymbol{\delta \theta}_X & \boldsymbol{\delta \theta}_Y & \boldsymbol{\delta \theta}_Z \end{bmatrix}^T$, $\boldsymbol{N}$ is the number of image points ,

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{\delta x}_{1j} \\ \boldsymbol{\delta y}_{1j} \\ \vdots \\ \vdots \\ \boldsymbol{\delta x}_{Nj} \\ \boldsymbol{\delta y}_{Nj} \end{bmatrix} = \begin{bmatrix} \boldsymbol{j}_1 \\ \vdots \\ \boldsymbol{j}_N \end{bmatrix} \boldsymbol{\delta p} = \boldsymbol{J}\boldsymbol{\delta p} \qquad (2)$$

$\boldsymbol{e}$ can be measured since it is the motion of the image features, we want to find $\boldsymbol{\delta p}$ that best fits the above equation. It is a non-linear least squares problem and the normal equation is

$$\boldsymbol{J}^T\boldsymbol{e} = \boldsymbol{J}^T\boldsymbol{J}\ \delta\boldsymbol{p}.$$

If we study closely the Jacobian in equation (2) we will see the problem of translation/rotation ambiguity. Without loss of generality, we simplify our analysis by choosing the j^{th} frame as the reference frame, so that $\boldsymbol{R}_j = \boldsymbol{I}$. The Jacobian has the following explicit form

$$\boldsymbol{e} = \begin{bmatrix} \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ -\frac{1}{\boldsymbol{Z}_{ij}} & 0 & \frac{\boldsymbol{X}_{ij}}{\boldsymbol{Z}_{ij}^2} & -\frac{\boldsymbol{X}_{ij}\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}^2} & 1+\frac{\boldsymbol{X}_{ij}^2}{\boldsymbol{Z}_{ij}^2} & -\frac{\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}} \\ 0 & -\frac{1}{\boldsymbol{Z}_{ij}} & \frac{\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}^2} & -\left(1+\frac{\boldsymbol{Y}_{ij}^2}{\boldsymbol{Z}_{ij}^2}\right) & \frac{\boldsymbol{X}_{ij}\boldsymbol{Y}_{ij}}{\boldsymbol{Z}_{ij}^2} & \frac{\boldsymbol{X}_{ij}}{\boldsymbol{Z}_{ij}} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix} \begin{bmatrix} \delta t_X \\ \delta t_Y \\ \delta t_Z \\ \delta\theta_X \\ \delta\theta_Y \\ \delta\theta_Z \end{bmatrix} \quad (3)$$

The ambiguity arises if the object is a planner and the field of view is small. That means there is little variation in $\boldsymbol{Z}_{ij}$, so that $\boldsymbol{Z}_{ij}$ is close to a constant value $\bar{\boldsymbol{Z}}$. When the field of view is small $X_{ij} \ll Z_{ij}$ and $Y_{ij} \ll Z_{ij}$. From (3), it can be seen that the confounding between translation in x-axis and rotation in y-axis is due to the high correlation between the 1st and the 5th columns in $\boldsymbol{J}$. Given that $\boldsymbol{Z}_{ij} \approx \bar{\boldsymbol{Z}}$, $\boldsymbol{X}_{ij} \ll \boldsymbol{Z}_{ij}$ and $\boldsymbol{Y}_{ij} \ll \boldsymbol{Z}_{ij}$, the two columns differ only by a constant factor $-\bar{\boldsymbol{Z}}$. Similarly, the 2nd and the 4th columns are almost linearly dependent, which explains the confounding between translation in y-axis and rotation in x-axis. Under these conditions, the rank of $\boldsymbol{J}$ is close to 4. A larger field of view can reduce this ambiguity problem as the images can include points with larger values of $\boldsymbol{X}_{ij}$ and $\boldsymbol{Y}_{ij}$, which has the desired effect of reducing the correlation between the confounded columns.

2.2. Multiple Cameras case

If we have K cameras (k=1,,,K) tighted together we assume that they have the same pose change $\delta\boldsymbol{p}$. The relative pose amongst cameras are represented by $\boldsymbol{R}_k$ and $\boldsymbol{D}_k$. The points observed by the k^{th} camera, based on the coordinate system of first camera k=1, is

$$\begin{aligned} \boldsymbol{P}_{ijk} &= \begin{bmatrix} \boldsymbol{X}_{ijk} & \boldsymbol{Y}_{ijk} & \boldsymbol{Z}_{ijk} \end{bmatrix}^T \\ &= \boldsymbol{R}_k\boldsymbol{R}_j(\boldsymbol{M}_i - \boldsymbol{d}_j - \boldsymbol{D}_k). \end{aligned} \quad (4)$$

Similar to equation (4) the motion of the image features for the k-th camera is

$$\boldsymbol{e}_{ijk} = \begin{bmatrix} \frac{1}{\boldsymbol{Z}_{ijk}} & 0 & -\frac{\boldsymbol{X}_{ijk}}{(\boldsymbol{Z}_{ijk})^2} \\ 0 & \frac{1}{\boldsymbol{Z}_{ijk}} & -\frac{\boldsymbol{Y}_{ijk}}{(\boldsymbol{Z}_{ijk})^2} \end{bmatrix} \begin{bmatrix} -\boldsymbol{R}_k\boldsymbol{R}_j \\ \boldsymbol{R}_k\boldsymbol{W}_X\boldsymbol{R}_k^T\boldsymbol{P}_{ijk} \\ \boldsymbol{R}_k\boldsymbol{W}_Y\boldsymbol{R}_k^T\boldsymbol{P}_{ijk} \\ \boldsymbol{R}_k\boldsymbol{W}_Z\boldsymbol{R}_k^T\boldsymbol{P}_{ijk} \end{bmatrix}^T \delta\boldsymbol{p} \quad (5)$$

,where $\boldsymbol{W}_X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, $\boldsymbol{W}_Y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$ and $\boldsymbol{W}_Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

For all feature points, $\boldsymbol{e}_k = \begin{bmatrix} \cdots & \boldsymbol{e}_{ijk} & \cdots \end{bmatrix}^T = \boldsymbol{J}_k\delta\boldsymbol{p}$.

To add more cameras to the equation, we can just stack the matrices. For example for a two camera system the total image

features error $\boldsymbol{e}$ is related to the change of the pose $\delta \boldsymbol{p}$ by

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \delta \boldsymbol{p} = \boldsymbol{J}\ \delta \boldsymbol{p}$$

and ($\boldsymbol{e}_1$, $\boldsymbol{J}_1$) and ($\boldsymbol{e}_2$, $\boldsymbol{J}_2$) are the flow vector and Jacobian matrix for the 1st and 2nd cameras respectively. For example, consider a specific configuration of a two-camera system, in which the cameras are configured back to back. The orientation of the second camera is obtained by rotating the first camera by 180° around the y-axis, *i.e.* $\boldsymbol{R}_k = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$. To simplify our analysis, we take the j^{th} frame as the reference frame, so that $\boldsymbol{R}_j$ is an identify matrix. Substitute $\boldsymbol{R}_j$ and $\boldsymbol{R}_k$ in (4), the combined Jacobian for this particular configuration is

$$\boldsymbol{J} = \begin{bmatrix} \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ -\frac{1}{\bar{Z}} & 0 & \frac{X_{ij1}}{\bar{Z}^2} & -\frac{X_{ij1}Y_{ij1}}{\bar{Z}^2} & 1+\frac{X_{ij1}^2}{\bar{Z}^2} & -\frac{Y_{ij1}}{\bar{Z}} \\ 0 & -\frac{1}{\bar{Z}} & \frac{Y_{ij1}}{\bar{Z}^2} & -\left(1+\frac{Y_{ij1}^2}{\bar{Z}^2}\right) & \frac{X_{ij1}Y_{ij1}}{\bar{Z}^2} & \frac{X_{ij1}}{\bar{Z}} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \frac{1}{\bar{Z}_2} & 0 & -\frac{X_{ij2}}{\bar{Z}_2^2} & \frac{X_{ij2}Y_{ij2}}{\bar{Z}_2^2} & 1+\frac{X_{ij2}^2}{\bar{Z}_2^2} & \frac{Y_{ij2}}{\bar{Z}_2} \\ 0 & -\frac{1}{\bar{Z}_2} & -\frac{Y_{ij2}}{\bar{Z}_2^2} & 1+\frac{Y_{ij2}^2}{\bar{Z}_2^2} & \frac{X_{ij2}Y_{ij2}}{\bar{Z}_2^2} & \frac{X_{ij2}}{\bar{Z}_2} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix}$$

Examine the columns in the combined Jacobian, one can see that the 1st and 5th columns, and likewise the 2nd and 4th columns, are now uncorrelated. In the special case that the distances between the scene and the cameras are roughly the same, so that $\bar{Z} \approx \bar{Z}_2$, and if the distribution of the 3D points are random so that X_{ijk} and Y_{ijk} are uncorrelated, then the columns in the combined Jacobian have the interesting property that they form a set of orthogonal basis. Since the column vectors are uncorrelated from each other, the translation/rotation ambiguity is completely removed. A back-to-back camera configuration is therefore an ideal setup for a two-camera system if the distance between the scene and the cameras is more or less uniform in all directions. To add more cameras into the system, it is advantageous to stack up 'pairs' of back-to-back cameras into the system, so as to maintain the orthogonality of the columns in the combined Jacobian.

3. Experimental results

3.1. Simulation results with known model

In this section, we will study the performances of different camera systems by simulation. In the experiment the 3D feature points of the scene are distributed randomly on the surface of a sphere, while the camera system is located at centre of that sphere. A sphere model has the advantages that data can be generated easily; features can be observed by cameras at any directions; and if the field of view is small, the images obtained resemble planar surfaces that have small depth variation.

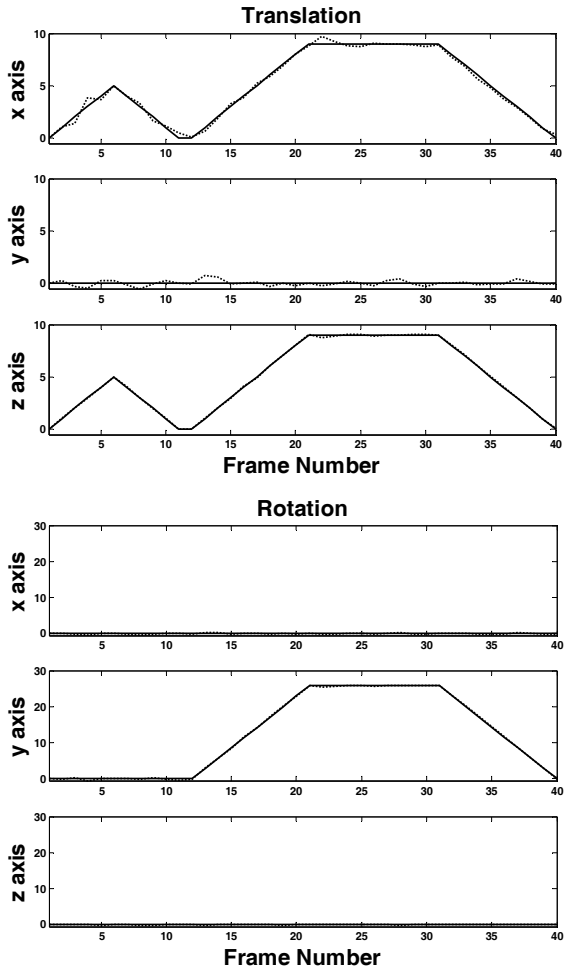

Figure2: Pose tracking of a known model by a single camera with small field of view (30°) and additive noise=*N*(0,2) : (top) tracking of translation; (bottom) tracking of rotation. Solid lines are the ground truth; dotted lines are the estimated results. Translations are in pixels and rotations are in degrees.

In our experiment, a single camera system captures around 100 points per frame; and for a two-camera system, the number of points captured is reduced to 50 points per frame per camera. This creates a fair basis for comparison between the two approaches. Zero mean Gaussian noise is added to the 2D feature points and the rotation speed of the system is limited to 0.05 radian/frame ($\sim 3^\circ$/frame). The algorithms used by the single camera and two-camera systems are essentially the same. Both are based on the Lowe's method [2]. The Jacobian matrices used are shown in the previous section.

Figure 2 shows a typical tracking result of a known model using a single camera with a small field of view (30°) and a moderate amount of added noise (zero mean Gaussian noise with 2 pixels standard deviation). The bold and the dotted line represent the ground truth and the estimation respectively. Translation is along the x and z-axis, and rotation (in degree) is around the y-axis. The tracking result of the single camera system is satisfactory, despite some small observable errors as indicated by the dotted line.

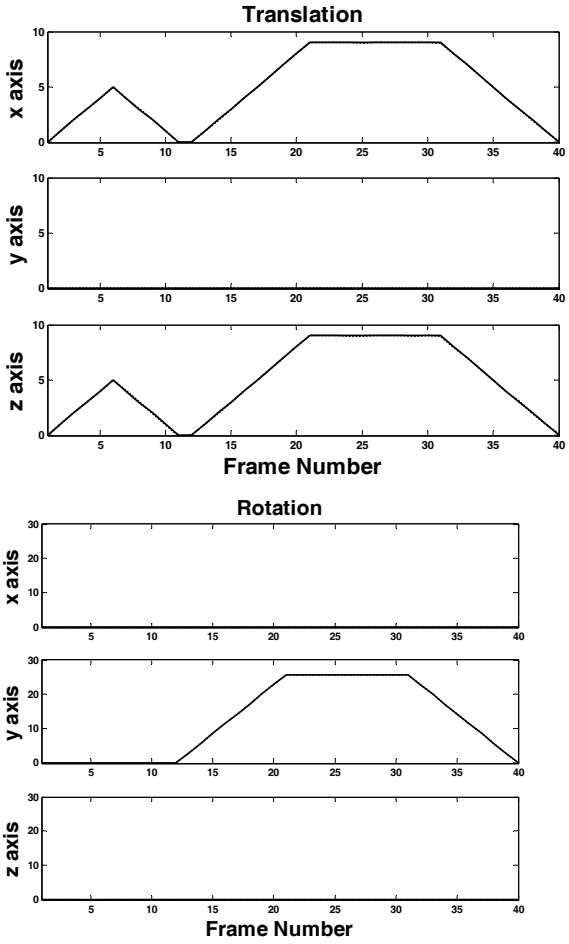

Figure 3: Pose tracking of a known model using two-camera system with same conditions as figure 2. Solid lines are the ground truth; dotted lines are the estimated results. Translations are in pixels and rotations are in degrees.

Figure 3 shows the tracking result using a two-camera system. The tracking result is excellent.

3.2. Simulation results with unknown model

The good result obtained by the single camera system showed that if the model of the scene is known, the ego-motion of the camera system can be tracked reliably. However, in applications such as robotic navigation and structure-from-motion, a model of the scene is usually not available. In that case, reliable tracking is difficult to achieve because a small pose tracking error will produce a slightly distorted model in the reconstruction, which in turn will introduce error in subsequent pose estimation, making tracking difficult over an extended period of time. In fact, the model can be found by a two-pass alternating structure-from-motion method described in [14]. Modifying the multiple camera methods for structure from motion can be done accordingly.

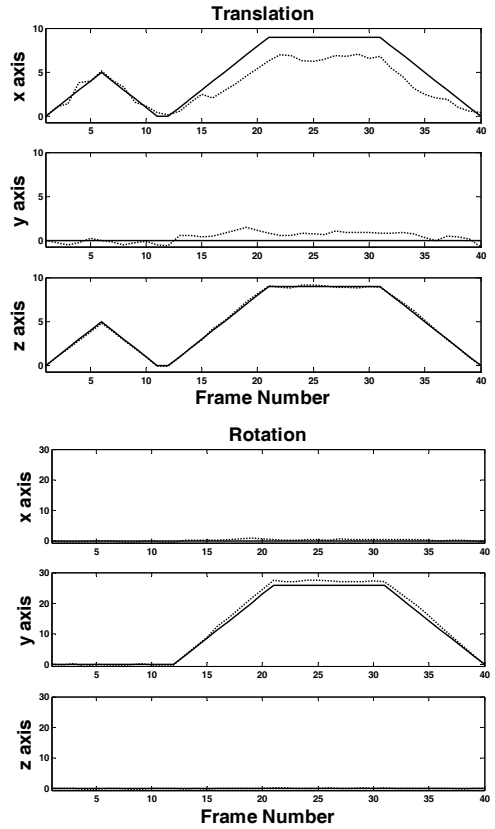

Figure 4: Pose tracking of an unknown model using a single camera under the same condition as in figure 2. Solid lines are the ground truth; dotted lines are the estimated results. Translations are in pixels and rotations are in degrees.

Fig. 4 shows the tracking result using a single camera for an unknown model using data generated by the same set of ground truth as in section 3.1. The first 10 frames have no rotation component, so that an initial model can be constructed directly. After the initialization, pose and model are estimated frame by frame using the interleaving algorithm. The translation/rotation ambiguity resulted in a small over-estimation of rotation in y-axis, and an under-estimation of translation in x-axis. The large translation error indicates that the reconstructed model is highly distorted. Tracking over an extended period of time is difficult to maintain because increasingly we are tracking the wrong model.

Fig. 5 shows a much improved tracking result using a two-camera system. The accurate pose tracking result obtained leaded to an accurate model reconstruction.

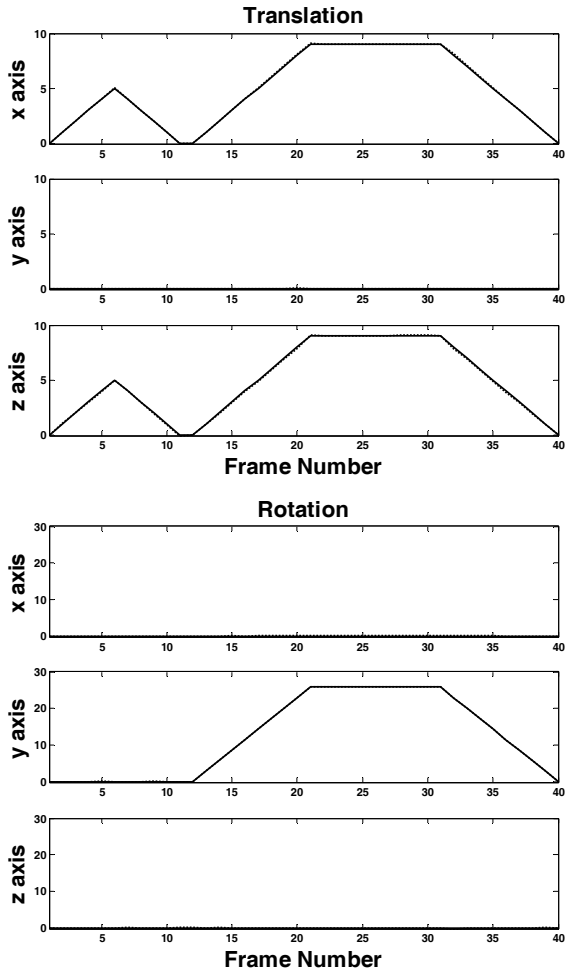

Figure 5: Pose tracking of an unknown model using two-camera system under the same condition as in figure 2. Solid lines are the ground truth; dotted lines are the estimated results. Translations are in pixels and rotations are in degrees.

3.3. Real scene experimental results

We have tested our algorithm using real data and the setup is shown in Figure 1. Two webcams (one pointing forward and the other pointing backward) were mounted on the head of a user and two video sequences of the cameras were captured. The motions of the images features were extracted by a feature tracker (such as the KLT algorithm [15]) and the first image and the features of one of the sequences are shown in Figure 6. The feature motions of two sequences were fed to our two-camera pose estimation algorithm and the front camera sequence is fed to our one-camera algorithm. The results or comparing the two algorithms is plotted in figure 7. Although we don't have the precise ground truth of the motion, but the direction of the pitch angle of the motion can be observed through the two video sequences. The result obtained by the two-camera method agrees with the motion sequences but not the one-camera method. In particular, the pitch angle of the head is observed to be turning into a positive direction from frame 35 to 48 but the one-camera method miscalculates the result. The reason is that without the information from the camera pointing backward the one-camera algorithm mixes up the horizontal translation and the pitch angles.

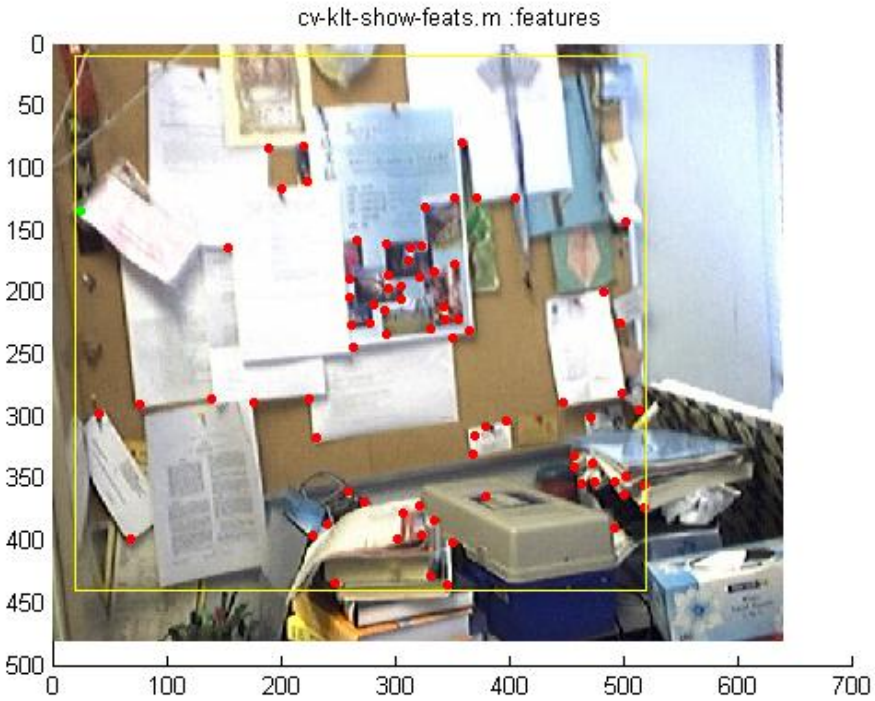

Figure 6: The first view with features marked (dots) for the camera pointing to the backward direction, it is a common office scene with no calibration objects.

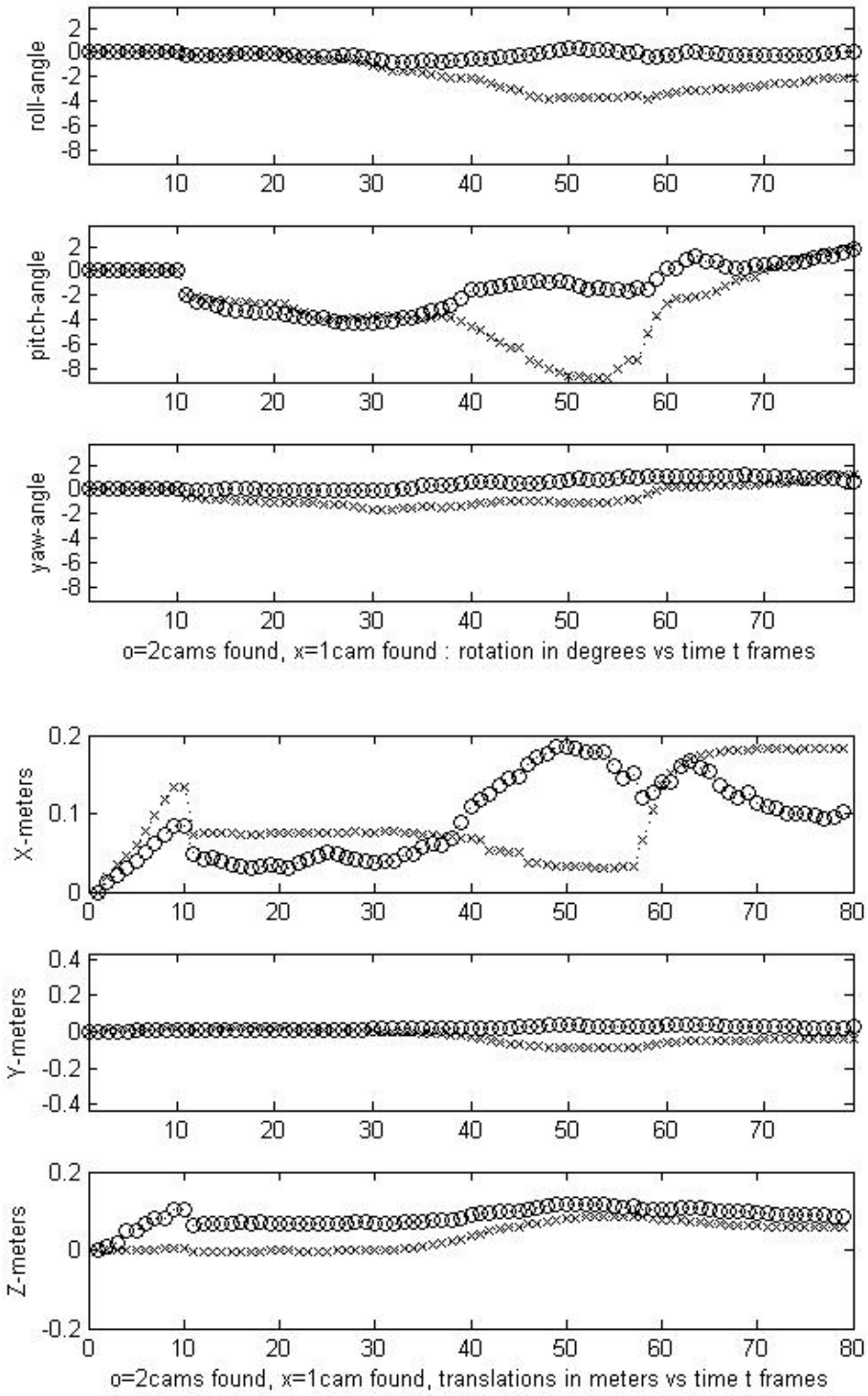

Figure 7 Pose tracking result of a real data experiment, 'O'=two-camera method; 'x' = one-camera method.

4. Conclusion and discussion

We presented a method for accurately tracking the motion of a human head by mounting multiple cameras on the head of the user. It is more accurate than the traditional method of tracking the images of a head, where reliable features are difficult to find and track. In contrary, for our approach of mounting cameras on the head, all the features surrounding the user are being used for tracking. We have also shown that employing two back-to-back cameras for tracking can solve the translation/rotation ambiguity problem and produces a very robust head tracking system. This has been demonstrated using mathematical analysis as well as experiments on synthetic and real data. With the prices, weights and sizes of digital cameras keep on decreasing this method is feasible, reliable and should cause very little disturbances to users.

5. References

[1] G. Adiv, "Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field**",** *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 5, pp. 477-489, May 1989.

[2] D.G. Lowe, "Fitting parameterized three-dimensional models to images", *IEEE Trans. Pattern Anal. Machine Intell.*, vol 13 , pp. 441-450, May. 1991.

[3] G. S. J. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14 , no. 10 , pp. 995-1013, Oct. 1992.

[4] D.J. Heeger and A.D. Jepson, "Subspace methods for recovering rigid motion i: Algorithm and implementation", *Int. Journal of Computer Vision*, 7:95-117, 1992.

[5] J. Weng, N. Ahuja, T.S. Huang, "Optimal motion and structure estimation**",** *IEEE Trans. Pattern Anal. Machine Intell.*, vol 15 , no. 9, pp. 864-884, Sept. 1993.

[6] S. Maybank, "*Theory of Reconstruction from Image Motion*", Springer-Verlag, 1993.

[7] K. Daniilidis and M.E. Spetsakis, "Understanding noise sensitivity in structure from motion", *Visual Navigation, Y. Aloimonos (Ed.)*, Lawrence Erlbaum Associates, pp. 61-88, 1996.

[8] R. Szeliski and S. B. Kang, "Shape Ambiguities in Structure From Motion", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 506-512, May 1997.

[9] C. Fermuller and Y. Aloimonos, "Observability of 3d motion", *International Journal of Computer Vision*, 37(1): 43-62, 2000.

[10] R.L. Thompson, I.D. Reid, L.A. Munoz, D.W. Murray, "Providing synthetic views for teleoperation using visual pose tracking in multiple cameras", *IEEE Trans. on Systems, Man and Cybernetics, Part A*, vol. 31, no. 1 , pp. 43-54, Jan. 2001.

[11] Baker, P, Fermuller, C, Aloimonos, Y, and Pless, R, *Computer Vision and Pattern Recognition, 2001*, Volumn 1, 8-14 Dec 2001, pp. 576-583.

[12] R. Pless, "Using Many Cameras as One", *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* , Volume: 2 , 18-20 June 2003, pp. 587-93.

[13] S. Tariq and F. Dellaert, "A Multi-Camera 6-DOF Pose Tracker**"**, *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on* , 02-05 Nov. 2004, pp. 296 – 297

[14] M. Chang and K.H. Wong, "Model Reconstruction and Pose Acquisition Using Extended Lowe's Method", IEEE Trans on Multimedia, vol 7, no. 2, April 2005, pp. 253-260.

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