Multi-Camera System Using PC-Cluster for Real-Time 3-D Pose Estimation

Kritsana Uttamang, Viboon Sangveraphunsiri
Department of Mechanical Engineering,
Faculty of Engineering, Chulalongkorn University, Bangkok 10330
E-mail:viboon.s@eng.chula.ac.th

Abstract

Tracking an object in three dimensional space is a major issue in computer vision which is normally solved through the extraction of representative features of the object, and two-dimension coordinates of the series of these image features are used to compute the position of the object. Typical system uses a binocular stereovision system. For environment with obstruction, only two cameras is not practical, multiple cameras are used instead. When multiple cameras are used, a certain similarity measure among extracted features from any two stereoscopic images helps to match the correspondences. In this way, three-dimensional measurement can be obtained from the 2-D coordinate of the features extracted from the different cameras. In this paper, a multiple cameras system (four cameras) and PC-cluster (two microcomputers) are used for estimating both position and velocity of a specified moving object. Noise filtering and features extraction of images are performed in the PC-cluster at video rate. Then, the extracted features from every camera will be used to locate the object. This is done in the main computer. The synchronization mechanism between computers has been developed using PCI-to-PCI data movers with fiber optic connection. We purpose a modified distortion model of Zhang's calibration method to reduce the computation time in 3-D reconstruction process. In our experiments, we setup the system to track 3-D paths which are generated by the PA10 robotic arm. The results show that the system can track both position and velocity of moving object in real-time with acceptable accuracy. Moreover, we show that the system can be adapted to be used for the reverse engineering application.

Keywords: multiple cameras system, vision based real-time tracking, 3D pose estimation, pc-cluster

1. Introduction

A lot of effort has been carried out to reconstruct the spatial geometry of scene using binocular stereovision systems. Most of the algorithms used a certain similarity measure between both stereoscopic images in order to match the correspondences. Unfortunately, matching homologous points between images is not always possible and false matches may appear. Thus, given a point in one image prediction algorithms must be used to fine the position of the homologous point in the second image. A computationally effective solution to overcome these difficulties relies on the use of multiple cameras vision system to reduce the amount of false matches. This is done by using epipolar geometry to predict correspondences. However, using multiple cameras represents a great deal of effort, since there are a lot of image data to be processed and there are many cameras to be calibrated. To realize a vision-based multiple cameras system which satisfies real-time tracking requirement, a high-performance hardware and fast image processing algorithm are needed. To overcome this requirement, a PC-cluster using fiber optic connection through PCI-to-PCI data mover interface is used.

In the early period, most of the work for real-time 3-D pose estimation used a binocular or stereovision. Tanaka, Maru and Miyazaki [1] proposed a 3-D object tracking technique using active stereo vision system. The object and corresponding coordinate were extracted from the background for each camera (time delay or latency between cameras exist) and epipolar geometry was used to calculate the 3-D coordinate of the object.

To improve the visual information, multi-camera system will be used. Yonemoto, Arita, Matsumoto and Taniguchi [2,3] developed a real-time coordinate capture system of a specified 3-D object based on multi-camera system using color marker. To improve performance of computation, the system implemented on a PC-cluster with network time protocol for PC-PC communication. Each camera connected to a dedicated PC. So that the real-time 3-D tracking was possible. However, this work did not concentrate on the tracking accuracy. The main purpose of the work is for tracking human motion behavior.

Garcia, Batlle and Salvi [4] developed a trinocular stereovision system for real-time pose detection. Each camera embedded with a real-time image processing hardware to perform object labeling and noise filtering at video rate. Both 3-D position and velocity could be obtained with this method.

Camera Calibration plays very important role for our application. Most of the vision-based systems for pose estimation of a scene require accurate prior known of system parameters, which can be estimated through a camera calibration process. The camera calibration process is based on the analysis of image feature of one or more views. A number of camera calibration methods have been proposed for the best result. They can be classified into two categories as the photogrammetric calibration and the self-calibration or auto-calibration. The photogrammetric calibration is performed by observing a calibration pattern whose geometries in 3D space are known accurately. The 2-D coordinates, with correspondence 3-D data, obtained from each camera, are used to calculate the camera parameters. There are many calibration methods can be done very efficiently, such as: Tsai' calibration method [5] used monoview with coplanar or non-coplanar set of points, of the known pre-specify object, to compute camera parameters including the radial lens distortion using the projective geometry and Taylor's series expansion. A three-step camera calibration method [6], proposed by Bacakoglu and Kamel, used linear least-squares to approximate camera parameters in the first step. In the second step, Bacakoglu and Kamel develop the alternative formulation to obtain an optimal rotation matrix from approximated parameters. Then translational and perspective transformations were optimized based on the optimized rotation matrix. In the third step, non-linear optimization is performed to handle lens distortion. Batista, Araujo and de Almeida [7] proposed the iterative multi-step, explicit camera calibration method, which is based on iterative approach to avoid the singularities obtained by the calibration equations when monoplane calibration points are used. Zhang [8] proposed the calibration procedure based on known coplanar points in 3-D of the calibration object. The object was taken in different view points using the same camera. Using the homography, both intrinsic and extrinsic parameters can be found.

The self-calibration methods do not use any calibration object. Just by moving a camera in a static scene, the rigidity of the scene provides constraints to intrinsic parameters. The correspondences between images, which are captured by the same camera in different view point, are sufficient to recover both intrinsic and extrinsic parameters. 3-D pose can be reconstructed up to a similarity. However, we cannot always obtain reliable results because there are many parameters to be estimated.

In addition to camera calibration, the 3D-reconstruction routine is needed in the 3-D pose estimation. In multiple cameras vision-based system for pose estimation, when number of image points (from two or more cameras used) of the calibration object are precisely known, as well as the intrinsic and extrinsic parameters of the calibrated cameras, the 3-D coordinate can be determined from the intersection in space of back projection rays. Each ray passes through the optical center and the known 2-D point in the image plane of the corresponding camera. These rays will intersect at the same point. Due to the presence of noise, these rays are not guarantee to intersect at a single point. There are some commonly-suggested methods to overcome this problem as:

Midpoint of the common perpendicular to the two rays:

This method compute 3D-point by minimizing the sum of the square distances of the 3D-point to each projected ray. However, this method strictly valid only in a Euclidean coordinate frame. Beardsley and Zisserman [9] suggest an alternative method based on Quasi-Euclidean to find the average of midpoint of common perpendicular between any two rays. This method consumes less computation and acceptable result especially in Euclidean Frame, otherwise the error still exists but the result of reconstruction is better than original midpoint method.

Least-Squares and Iterative Least-Squares:

This method uses less computation and gives high accuracy. For N cameras, the 2N linear equations are obtained from the relationship between camera model and points in the image planes. Least-Squares method uses Singular Value Decomposition (SVD) or Pseudo-invert matrix to solve the 2N linear equations with 3 unknown to obtain 3D-point. This method has no geometrical meaning [9,10] and its results vary with the weights upon its linear equations. Hartley and Sturm [10] proposed an alternative method called iterative least-squares method. The original least-squares method is modified by adding weighting factor to the linear equations. The suitable weighting factor is adjusted in each iteration. The result is more accurate but consumes more computation than the original least-squares method.

Liu et al. [11] use least-squares method to reconstruct 3D-point from corrected image points. The first-order maximum likelihood estimation use to correct image points, which assumed a Gaussian noise distribution embedded in measurement. This method can be reconstructed 3D-point more efficient than both original least-squares method and iterative least-squares method.

Bundle Adjustment:

Bundle adjustment is the method to solve the problem of refining a visual reconstruction to produce jointly optimal 3D-structure and viewing parameter [12] by using some optimization method such as Levenberg-Maquardt. There are many optimization methods used in bundle adjustment as shown in [12].

Hartley and Zisserman [13] seek the maximum likelihood solution assuming that the measurement noise is Gaussian. They minimize the image distance between the detected image points and reprojected points.

Bartoli [14] introduces an algorithm for bundle adjustment based on quasi-linear optimization to obtain 3D model from long image sequences.

In this paper, we purpose the vision system that uses multiple cameras for tracking a moving object in 3-D space. The developed system uses multiple computers in order to increase speed and efficiency. It can tracks the object in real-time.

2. System Overview

In this paper, we developed a real-time tracking system using multiple cameras. The system implemented on PC-cluster (in our case, we are using two computers and four cameras) connected through fiber cable. The synchronization mechanism between PCs is through PCI-to-PCI data mover interface. The flow of the conceptual process is as following:

i. Cameras calibration.
ii. Two-dimension features extraction for each view.
iii. Three-dimension pose estimation for the object.
iv. Real-time rendering.
v. Perform i-iv for each frame

Fig. 1 shows the arrangement of the processing modules developed. Each processing module has been designed as follows:

1. *Image Capturing Module*: This module consists of image capturing and resizing (1280x1024-320x240) for each camera. The captured image data is sent to 2-D image processing module.

2. *2-D Image Processing Module*: The image data received from the Image Capturing Module is used for 2-D image feature extraction. This 2-D image feature will be used by the 3-D Pose Estimation Module.

3. *3-D Pose Estimation Module*: This module is used for estimating 3-D position and velocity of the object.

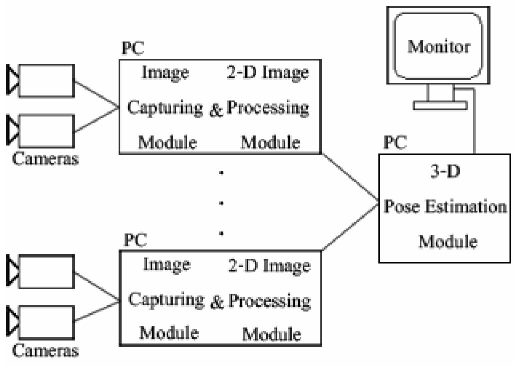

Fig.1. An arrangement of the processing modules on PC-cluster.

3. Camera Calibration

3.1Camera Model

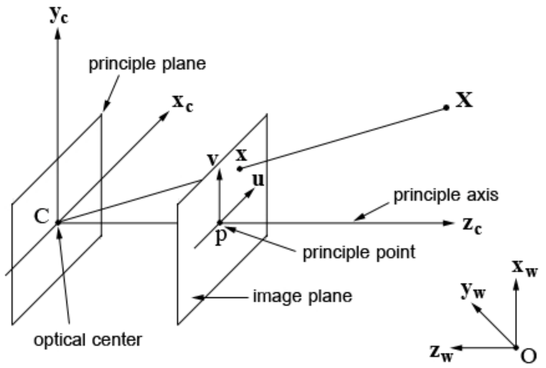

Fig. 2. The pinhole camera model.

Fig. 2 shows the pinhole camera model defined as the 3x4 homogeneous camera projection matrix **P**. This projection matrix **P** is used to transform a 3-D point in world coordinate to computer image point. The 3-D point will be represented by a homogeneous 4-vector as $\mathbf{X_w} = (x_w \quad y_w \quad z_w \quad 1)^\mathbf{T}$. And the computer image point will be represented by a homogeneous 3-vector as $\mathbf{x} = (u \quad v \quad 1)^\mathbf{T}$.

The transformation can be written as:

$$\mathbf{x} = \mathbf{PX_w} \tag{1a}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{T}\right]\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{1b}$$

where

T is the translation vector.

R is 3x3 rotation matrix which can be expressed as Rodrigues's formula with rotational 3-vector $\mathbf{r} = \left[r_x \quad r_y \quad r_z\right]^T$ whose direction is rotating axis and magnitude is rotating angle.

K is 3x3 camera calibration matrix which consist of five camera parameters describe as follow:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

α_x, α_y represent the focal length of the camera in term of pixel dimensions in the x and y direction, respectively.

u_0, v_0 are the principle points in terms of pixel dimensions.

s is referred to the skew parameter.

The parameters used in the transformation can be categorized into two following classes:

1. *Extrinsic Parameters:* These parameters are used in the transformation from the world coordinate system to the 3-D camera coordinate system. The origin of the camera coordinate system is at the optical center. There are six extrinsic parameters or components: three components for the rotation vector and the three components for the translation vector **T**.

2. *Intrinsic Parameters:* These parameters are used in the transformation from 3-D object coordinate represented in the camera coordinate system (x_c, y_c, z_c), to the computer image frame buffer coordinate (u_f, v_f). For the pinhole camera model, there are five intrinsic parameters as $\alpha_x, \alpha_y, u_0, v_0$ and s.

3.2 Computation of the Camera Matrix P

Camera calibration means to compute the camera intrinsic and extrinsic parameters which relate the 3-D world coordinate system (x_w, y_w, z_w) to the 2-D computer image coordinate system (u, v). By giving a number of points whose coordinates in the world coordinate are known and whose image coordinates are measured. Zhang [4] uses a set of images of a calibration pattern which are capturing from different view points. The calibration pattern provides a set of points on the same plane whose 3-D coordinate can be measured accurately. Zhang relates 2-D image point position with 3-D point in world coordinate system with homography **H** which defines up to scale factor.

$$s\mathbf{m} = \mathbf{HM} \tag{3}$$

where

$$\mathbf{H} = \mathbf{K}\left[\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{t}\right] \tag{4}$$

given:

$$\mathbf{H} = \left[\mathbf{h_1} \quad \mathbf{h_2} \quad \mathbf{h_3}\right] \tag{5}$$

then

$$\left[\mathbf{h_1} \quad \mathbf{h_2} \quad \mathbf{h_3}\right] = \lambda\mathbf{K}\left[\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{t}\right] \tag{6}$$

λ is arbitrary constant and $\mathbf{h_i}$ is 3-vector of column i of homography **H** which can be calculated directly from given an enough number of 2-D image points and its corresponding 3-D position with respect to world coordinate system.

By using orthonornal constraint between $\mathbf{r_1}$ and $\mathbf{r_2}$:

$$\mathbf{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \quad (7)$$

$$\mathbf{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_1 - \mathbf{h}_2^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \quad (8)$$

$\mathbf{K}^{-T}\mathbf{K}^{-1}$ is known as image of absolute conic:

$$\mathbf{B} = \mathbf{K}^{-T}\mathbf{K}^{-1} \quad (9)$$

$$\mathbf{K}^{-T}\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{\alpha_x^2} & -\frac{s}{\alpha_x^2\alpha_y} & \frac{v_0 s - u_0\alpha_y}{\alpha_x^2\alpha_y} \\ -\frac{s}{\alpha_x^2\alpha_y} & \frac{s^2}{\alpha_x^2\alpha_y^2} + \frac{1}{\alpha_y^2} & -\frac{s\left(v_0 s - u_0\alpha_y\right)}{\alpha_x^2\alpha_y^2} - \frac{v_0}{\alpha_y^2} \\ \frac{v_0 s - u_0\alpha_y}{\alpha_x^2\alpha_y} & -\frac{s\left(v_0 s - u_0\alpha_y\right)}{\alpha_x^2\alpha_y^2} - \frac{v_0}{\alpha_y^2} & \frac{\left(v_0 s - u_0\alpha_y\right)^2}{\alpha_x^2\alpha_y^2} + \frac{v_0^2}{\alpha_y^2} + 1 \end{bmatrix} \quad (10)$$

From equation (8) and (9), a single image of a calibration plane gives 2 sets of equations. But the calibration matrix $\mathbf{K}$ has 5 degrees of freedom, so at least 3 images of calibration pattern needed.

When homography of each images are known, all intrinsic and extrinsic parameters can be determined as follow:

$$v_0 = \frac{\left(B_{12}B_{13} - B_{11}B_{23}\right)}{\left(B_{11}B_{22} - B_{12}^2\right)} \quad (11)$$

$$\zeta = B_{33} - \frac{B_{13}^2 + v_0\left(B_{12}B_{13} - B_{11}B_{23}\right)}{B_{11}} \quad (12)$$

$$\alpha_x = \sqrt{\frac{\zeta}{B_{11}}} \quad (13)$$

$$\alpha_y = \sqrt{\frac{\zeta B_{11}}{\left(B_{11}B_{22} - B_{12}^2\right)}} \quad (14)$$

$$s = -\frac{B_{12}\alpha_x^2\alpha_y}{\zeta} \quad (15)$$

$$u_0 = \frac{sv_0}{\alpha_x} - \frac{B_{13}\alpha_x^2}{\zeta} \quad (16)$$

$$\mathbf{r}_1 = \lambda\mathbf{K}^{-1}\mathbf{h}_1 \quad (17)$$

$$\mathbf{r}_2 = \lambda\mathbf{K}^{-1}\mathbf{h}_2 \quad (18)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (19)$$

$$\mathbf{t} = \lambda\mathbf{K}^{-1}\mathbf{h}_3 \quad (20)$$

when

$$\lambda = \frac{1}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|} = \frac{1}{\left\|\mathbf{K}^{-1}\mathbf{h}_2\right\|} \quad (21)$$

Where each B_{ij} is element of matrix $\mathbf{B}$ at the i-th row and j-th column.

In this paper, 8×8 of 2 cm. square chessboard will be used as the calibration pattern as shown in Fig. 3, whose 49-point correspondences have been used in order to compute camera matrix. The plane of the pattern is called model plane.

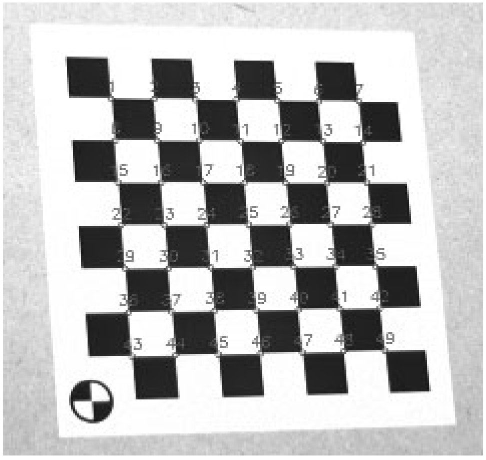

Fig. 3. The chessboard pattern for calibration

3.3 Radial Distortion

From the camera model described in the early section, the point in the world coordinate, image point, and optical center are collinear. For real or actual lenses, this assumption will not true. The most important deviation is generally a radial distortion [5]. In practice, this error becomes more significant as the focal length of the lens decreases.

The actual projected point is related to the ideal point by radial displacement. Zhang modeled actual projected point as function of distortion factor, which is considered as Taylor's expansion function, multiplied by ideal point position as follow:

$$\mathbf{x}_d = \mathbf{x}_u\left(1 + k_1' r_u^2 + k_2' r_u^4\right) \qquad (22)$$

where

$\mathbf{x}_u$ is the ideal normalize image coordinate (which obeys linear projection) $\mathbf{x}_u = \left(x_u, y_u\right)$.

$\mathbf{x}_d$ is actual normalize image coordinate, after radial distortion, $\mathbf{x}_d = \left(x_d, y_d\right)$.

r_u is the radial distance $\sqrt{x_u^2 + y_u^2}$ from the center for radial distortion.

k_1', k_2' are first and second order of lens distortion coefficients.

For our developed system, due to an object is being tracked in real-time, so we need to minimize the computation time in the reconstruction process. In the reconstruction process, the actual image position, from image view, of the object is known but the ideal image position (in 3-D world coordinate) needs to be calculated. From equation (22), it is difficult to calculate the ideal image position from the given actual image position. The polynomial of degree 5 need to be solved and it will consume a lot of computational time. To reduce computational time, we modeled the ideal image position as function of actual image position instead as:

$$\mathbf{x}_u = \mathbf{x}_d\left(1 + k_1 r_d^2 + k_2 r_d^4\right) \qquad (23)$$

where

r_d is the radial distance $\sqrt{x_d^2 + y_d^2}$ from the center for radial distortion.

k_1, k_2 are first and second order of lens distortion coefficients.

Let $\left(u_u, v_u\right)$ and $\left(u_d, v_d\right)$ is ideal and actual pixel image point respectively. Our strategy is to estimate k_1 and k_2 after having estimated the other parameters, From (23), we have two equations for each point in each image:

$$\begin{bmatrix} \left(u_d - u_0\right)\left(x_d^2 + y_d^2\right) & \left(u_d - u_0\right)\left(x_d^2 + y_d^2\right)^2 \\ \left(v_d - v_0\right)\left(x_d^2 + y_d^2\right) & \left(v_d - v_0\right)\left(x_d^2 + y_d^2\right)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \left(u_u - u_d\right) \\ \left(v_u - v_d\right) \end{bmatrix} \qquad (24)$$

Given m points in n images, we can stack all equations together to obtain in total $2mn$ equations then linear least-squares optimization is used to obtain k_1 and k_2 from (24).

In order to obtains the best solution of intrinsic parameters, extrinsic parameters and radial distortion coefficients, Levenberg-Marquardt optimizer [5,7,8,15] has been used. The optimization will start with the intrinsic and extrinsic parameter values, computed by Zhang's calibration method. The minimize function of Levenberg-Marquardt is:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathbf{x}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)\right\|^2 \qquad (25)$$

where

n is number of images of model plane.

m is number of points on model plane.

$\mathbf{R}_i$ is rotation matrix which corresponding to image *i*.

$\mathbf{t}_i$ is translation vector which corresponding to image *i*.

$\hat{\mathbf{x}}(\mathbf{K}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)$ is projection of point $\mathbf{X}_j$ in image *i*.

4. Feature Extraction

The accurate detection of image features is required in applications of 3-D reconstruction. In this paper, the spherical ball whose image always circle (2-D image) has been used as tracking target. The application of the Hough transformation to detection of circular objects has been employed to detect center point and radius of tracking object as shown in Fig. 4.

Fig. 4. The circle detection using Hough transformation

5. Pose Estimation

Consider the multiple cameras system which has *n* cameras, given $\mathbf{P_i}$ is the i-th camera matrix, and $\mathbf{x_i}$ is point in the i-th image of the 3-D world point $\mathbf{X}$ and corresponding to camera matrix $\mathbf{P_i}$, then we have $\mathbf{x_i} = \mathbf{P_i X}$. By using relation of vector cross product:

$$\mathbf{x_i} \times \left(\mathbf{P_i X}\right) = 0 \tag{26}$$

Then

$$\begin{aligned} x_i\left(\mathbf{p}_i^{3T}\mathbf{X}\right) - \left(\mathbf{p}_i^{1T}\mathbf{X}\right) &= 0 \\ y_i\left(\mathbf{p}_i^{3T}\mathbf{X}\right) - \left(\mathbf{p}_i^{2T}\mathbf{X}\right) &= 0 \\ x_i\left(\mathbf{p}_i^{2T}\mathbf{X}\right) - y_i\left(\mathbf{p}_i^{1T}\mathbf{X}\right) &= 0 \end{aligned} \tag{27}$$

where

$\mathbf{p}_i^{jT}$ is the j-th row of camera matrix $\mathbf{P_i}$

From each camera, only 2 equations are independent, let choose 2 equations from each camera. A matrix **L** can be obtained by stacking up equation (27) as [10,13,16]:

$$\mathbf{LX} = \begin{bmatrix} x_1\mathbf{p}_1^{3T} - \mathbf{p}_1^{1T} \\ y_1\mathbf{p}_1^{3T} - \mathbf{p}_1^{2T} \\ x_2\mathbf{p}_2^{3T} - \mathbf{p}_2^{1T} \\ y_2\mathbf{p}_2^{3T} - \mathbf{p}_2^{2T} \\ \vdots \\ x_n\mathbf{p}_n^{3T} - \mathbf{p}_n^{1T} \\ y_n\mathbf{p}_n^{3T} - \mathbf{p}_n^{2T} \end{bmatrix} \mathbf{X} = \mathbf{0} \tag{28}$$

Because of noise embedded in the captured data point, so, we can not obtain the exact solution of the above equation. The solution for $\mathbf{X}$ can be obtained by least-square solution (LS) of equation (27). LS gives an accurate result but it has no geometrical meaning. The other method is the bundle adjustment with Levenberg-Marquardt optimization (LM). This method tries to minimize geometric image distance between measured image point and reprojected image point of the estimated 3D world point. LM gives a better accuracy solution but it too slow because LM is an iteration-based method. In contrast, LS is faster than LM but the error is also more than LM.

6. Software Implementation

In this section, we proposed the software architecture follow the outline of the developed system mention earlier. This prototype application is suitable for real-time tracking of an object (spherical ball) using multiple cameras. The developed software is implemented in PC-cluster connected via PCI-to-PCI using fiber optic cable.

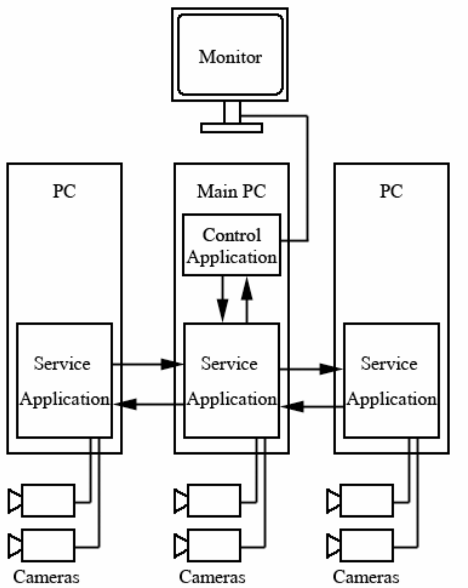

Fig. 5. An arrangement of the developed software

Fig. 5 shows a diagram of the proposed software which can be divided into the following application processes as:

1. *Service Application*: Service application is a Windows-based application which runs in the background of Windows NT operating system. In each PC, a service application will be responsible for image capturing and 2-D image processing, so an image capturing module and a 2-D image processing module are contained in this service application. Moreover, the service application has been developed to perform communicate between PCs in PC-cluster system via fiber optic cable with PCI-to-

PCI data mover card and it also provides interfaces to other Windows application.

2. *Control Application*: In PC-cluster system, one PC has been used as main PC, it controls other remote PCs for tracking the target object. In main PC, control application has been developed up on top of service application described earlier by using Component Object Modeling (COM) architecture. Control application receives 2-D image coordinates of the target from service application and estimate for 3-D position of the target in the world coordinate. So, the 3-D pose estimation module is contained in this application. Moreover, the control application also provides application user interfaces. User can control tracking system through this application.

7. Experiment

Our setup system consists of 2 PCs, the main PC has dual-CPUs which are Pentium IV 3.2 GHz and 2GB of RAM installed. The remote PC has single CPU which is Pentium IV 2.0 GHz with 512 MB of RAM and they are connected together via fiber optic cable with dataBLIZZARD PCI-to-PCI data mover card. Each PC has 2 CCD cameras, PixeLINK PL-A741-BL, with 16mm lens connected through IEEE 1394 ports.

Fig. 6. The system setup for testing accuracy of measurements

The first experiment is to demonstrate the accuracy of the multi-camera system by detecting the known locations (corners of the small squares of the calibration pattern). Fig. 6 shows the arrangement of the system. Images appear in each camera are shown in Fig. 7 as labeling by camera serial no. Table 1 shows the error, standard deviation, and the maximum error obtained from the experiments. If the object is viewed by all the cameras, we obtain around 1 mm. accuracy. The accuracy will reduce if some cameras are obstructed. The better result will be obtained if we use better cameras.

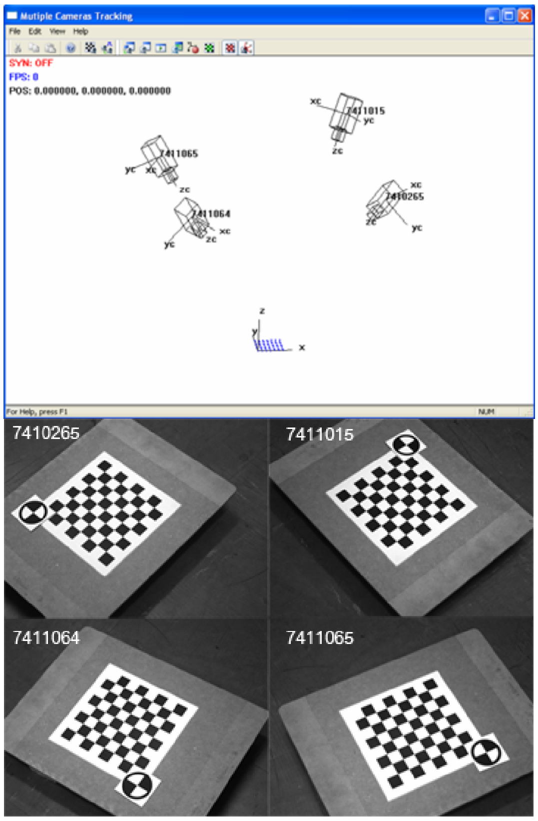

Fig. 7. Images appear in each camera

Table 1. The error obtained from the measurements

No. of cameras	Average error (mm.)	Standard deviation (mm)	Maximum error (mm)
4	0.101	0.044	0.187
3 (wo 7410265)	0.125	0.049	0.203
3 (wo 7411015)	0.123	0.048	0.220
3 (wo 7411064)	0.135	0.054	0.244
2 (w 7410265, 7411015)	0.188	0.067	0.328
2 (w 7410265, 7411064)	0.163	0.064	0.291
2 (w 7410265, 7411065)	0.148	0.046	0.278
2 (w 7411015, 7411064)	0.146	0.050	0.235
2 (w 7411015, 7411065)	0.169	0.112	0.664
2 (w 7411064, 7411065)	0.265	0.136	0.595

w= using, wo = without

The developed system can detects the moving target, which is a spherical ball attached to a Mitsubishi PA-10 robot arm as shown in Fig. 8. We generate motion for the robot arm and the system will track the moving target. We found that the calibrated cameras system can detects position of spherical ball with acceptable accuracy (position error smaller than 1 mm within a working volume) as shown in Fig 9, 10, 11, 12, 13, and 14. Fig. 9, 10, and 11 show the comparison of the result from the measurements and the robot arm move in XY-plane, YZ-plane, and ZX-plane, respectively. The speed of the robot arm is set to 0.05 rad/sec. Fig 11, 12, and 13 show the system detecting the target moving helically in Z-direction, Y-direction, and X-direction, respectively. Again the velocity of the robot arm is set as the same.

The developed system can be used as the coordinate measuring machine for reverse engineering applications. Fig. 15 shows the collection of points measured along a complex surface using the spherical ball as a probe. The tri-angular mesh created from the measurement data is shown in Fig. 16. It can be improved the measurement or the quality of the triangular mesh by reducing the size of the probe as well as increase the number of points measured.

Fig. 17 shows the tracking of the target object with s-curve velocity profile. The robot arm is program to move in y-direction with acceleration and deceleration set equal to 20 mm/s^2. The total distant is 300 mm with 40 mm/s constant velocity. The result shows that the system can track the target very well. We can improve the tracking by using faster cameras. A well-defined environment can be used to reduce exposure time of the camera.

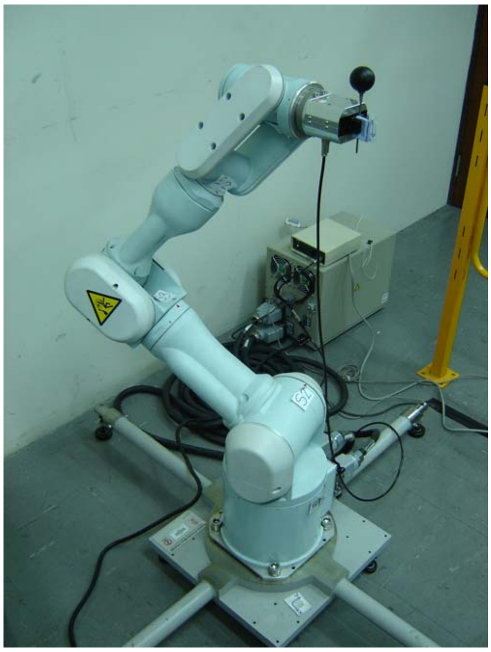

Fig 8. The robot arm holds a target

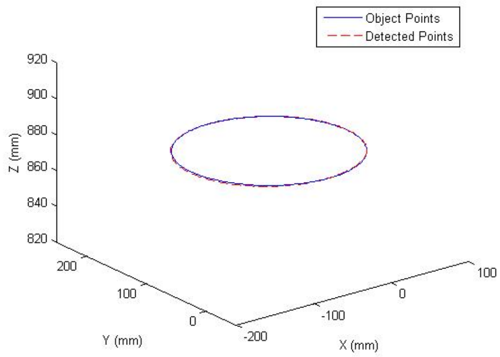

Fig. 9. The target moves in the XY-plane

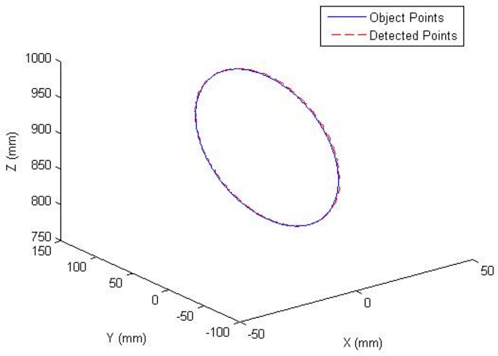

Fig. 10. The target moves in the YZ-plane

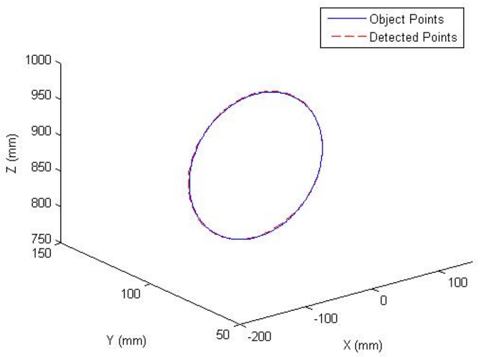

Fig. 11. The target moves in the ZX-plane

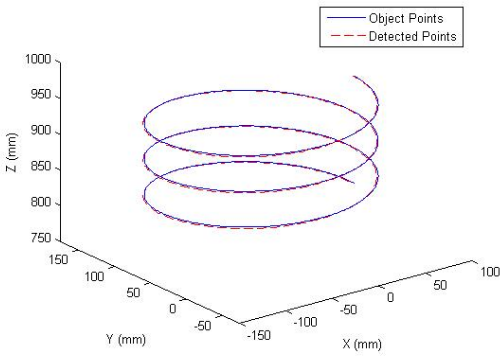

Fig. 12. Helical motion in z-direction

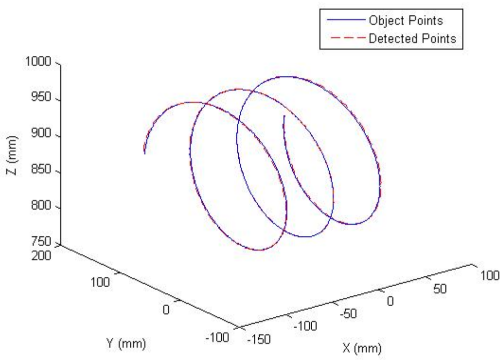

Fig. 13. Helical motion in X-direction

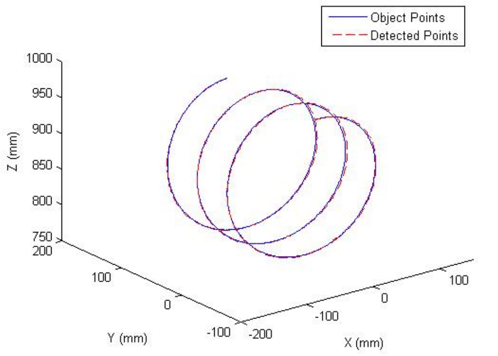

Fig. 14. Helical motion in Y-direction

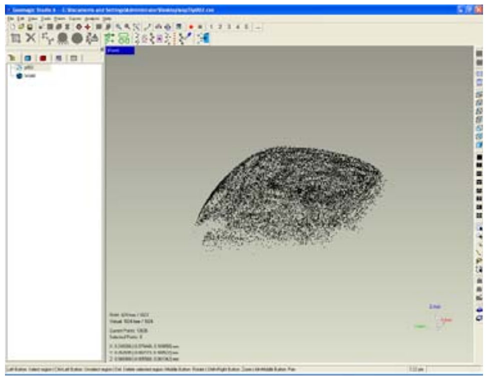

Fig. 15. The collection of measurement points of a complex surface

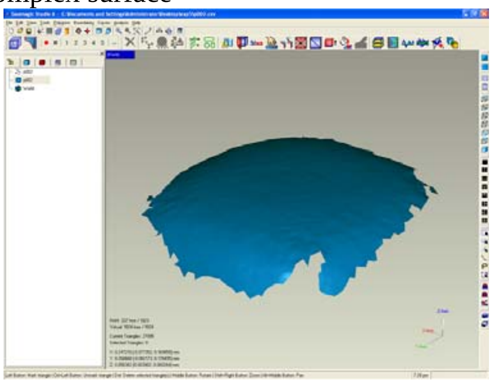

Fig. 16. The tri-angular mesh created from the measurement points

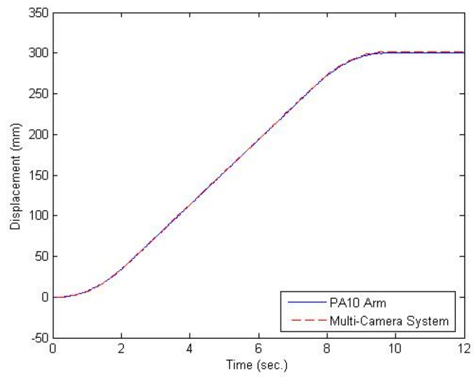

Fig. 17. Linear motion with s-curve velocity profile

8. Conclusions

In this paper, a real-time 3-D tracking system using multiple cameras has been developed. The developed system can track the moving target, which is a spherical object, with the acceptable accuracy. Four cameras are used in our experiments. The better quality cameras can be incorporated to the system, the better measurement can be obtained. And the more number of cameras, the more versatile of the measurement. The system can be adapted to track the target which is not a spherical object by changing the recognition algorithm. There are many method have been developed such as Generalized Hough Transform algorithm for detect arbitrary shapes. Besides 3-D pose estimation, the system can be adapted to be a coordinate measuring system for the reverse engineering application. The system will be much cheaper than the conventional system available in the market.

9. References

[1] Tanaka, M., Maru, N. and Miyazaki, F., 3-D Tracking of a Moving Object by an Active Stereo Vision System, 20th International Conference on Industrial Electronics, Control and Instrumentation, vol. 2. pp. 816-820, 1994.

[2] Yonemoto, S., Matsumoto, A., Arita, D. and Taniuchi, R., A real-time motion capture system with multiple camera fusion, IEEE International Conference on Image Analysis and Processing, September, pp. 600-605, 1999.

[3] Arita, D., Yonemoto, S. and Taniuchi, R., Real-time computer vision on PC-cluster and Its application to real-time motion capture, Proceedings. 5th IEEE International Workshop on Computer Architectures for Machine Perception, pp.205-214, 2000.

[4] Garcia, R., Batlle, J. and Salvi, J., A New Approach to Pose Detection using a Trinocular Stereovision System, Real-Time Imaging. pp.73-93, 2002.

[5] Tsai, R., A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, IEEE Journal of Robotics and Automation, vol. 3. no.4, pp.323-344, 1987.

[6] Bacakoglu, H. and Kamel, M. S., A Three-Step Camera Calibration Method, IEEE Transactions on Instrumentation and Measurement, vol. 46. no. 5, pp.1165-1172, 1997.

[7] Batista, J., Araújo, H. and de Almeida, A. T., Iterative Multistep Explicit Camera Calibration, IEEE Transactions on Robotics and Automation, vol. 15. no. 5., pp.897-917, 1999.

[8] Zhang, Z., A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22. no. 11, pp.1330-1334, 2000.

[9] Beardsley, P. A., Zisserman, A., and Murray, D. W., Sequential Updating of Projective and Affine Structure from Motion, International Journal of Computer Vision, vol. 23. no. 3, pp. 235-259, 1997.

[10] Hartley, R. and Sturm, P., Triangulation," Computer Vision and Image Understanding, vol. 68. no. 2. pp. 146-157, 1997.

[11] Liu, B., Yu, M., Maier, D. and Männer, R., An Efficient and Accurate Method for 3D-Point Reconstruct from Multiple Views, International Journal of Computer Vision, vol. 65. no. 3, pp. 175-188, 2005.

[12] Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., Bundle Adjustment – A Modern Synthesis, Vision Algorithms'99, LNCS 1883, pp. 298-372, Springer-Verlag, Berlin, 2000.

[13] Hartley, R. and Zisserman, A., Multiple View Geometry in Computer Vision, 2nd Cambridge University Press, 2003.

[14] Bartoli, A., A Unified Framewok for Quasi-Linear Bundle Adjustment, in Proceedings of the 16th International Conference on Pattern Recognition, vol. 2, pp.560-563, 2002.

[15] Sangveraphunsiri, V. and Longtantong, W., A 3D Particle Tracking System Using Stereo Vision, Thai Robot Society Conference on Robotics and Industrial Technology, pp 55-66, 2006.

[16] Faugeras, O. and Luong, Q., The Geometry of Multiple Images, Cambridge, MA: MIT Press, 2001.