Calibration of an Active Vision System and Feature Tracking Based on 8-Point Projective Invariants

By

Chen Zhi-Yi

陈志义

A Master Thesis

Submitted in partial fulfilment of the requirements

for the degree of

Master of Philosophy

in

Department of Electronic Engineering

The Chinese University of Hong Kong

June 1997

香港中文大學
大學
圖書館系統
22 MAY 1998
UNIVERSITY
LIBRARY SYSTEM
THE CHINESE UNIVERSITY OF HONG KONG

Abstract

An active vision system needs to be calibrated before it can be used in active vision research. The active vision system calibration problem can usually be decomposed into three subproblems: robot manipulator calibration, robot hand-eye (head-eye) calibration and camera calibration. Since our active vision system has been fabricated and assembled very accurately, only the latter two problems are investigated. We have accomplished the camera calibration and head-eye calibration for our active vision system with a real image sequence.

Projective invariants can be defined as a set of geometric properties which remained unchanged under projective transformation. A new tracking algorithm for the feature points on a polyhedral object using planar projective invariants is proposed. This algorithm has been verified on real data acquired by an active vision system. The results of both feature tracking and shape reconstruction are good.

Feature tracking method based on more general 8-point projective invariants are performed on several real image sequence acquired by the active vision system. Experimental results are reported and analyzed .

Table of Contents

List of Symbols S.1

Chapter 1 Introduction

1.1 Active Vision Paradigm and Calibration of Active Vision System 1.1
1.1.1 Active Vision Paradigm 1.1
1.1.2 A Review of the Existing Active Vision Systems 1.1
1.1.3 A Brief Introduction to Our Active Vision System 1.2
1.1.4 The Stages of Calibrating an Active Vision System 1.3
1.2 Projective Invariants and Their Applications to Feature Tracking 1.4
1.3 Thesis Overview 1.4
References 1.5

Chapter 2 Calibration for an Active Vision System: Camera Calibration

2.1 An Overview of Camera Calibration 2.1
2.2 Tsai's RAC Based Camera Calibration Method 2.5
2.2.1 The Pinhole Camera Model with Radial Distortion 2.7
2.2.2 Calibrating a Camera Using Monoview Noncoplanar Points 2.10
2.3 Reg Willson's Implementation of R. Y. Tsai's RAC Based Camera Calibration Algorithm 2.15
2.4 Experimental Setup and Procedures 2.20
2.5 Experimental Results 2.23
2.6 Conclusion 2.28
References 2.29

Chapter 3 Calibration for an Active Vision System: Head-Eye Calibration

3.1 Why Head-Eye Calibration 3.1
3.2 Review of the Existing Head-Eye Calibration Algorithms 3.1
3.2.1 Category I Classic Approaches 3.1
3.2.2 Category II Self-Calibration Techniques 3.2
3.3 R.Tsai's Approach for Hand-Eye (Head-Eye) Calibration 3.3

3.3.1 Introduction 3.3
3.3.2 Definitions of Coordinate Frames and Homogeoeous Transformation Matrices 3.3
3.3.3 Formulation of the Head-Eye Calibration Problem 3.6
3.3.4 Using Principal Vector to Represent Rotation Transformation Matrix 3.7
3.3.5 Calculating $\boldsymbol{R}_{cg}$ and $\boldsymbol{T}_{cg}$ 3.9
3.4 Our Local Implementation of Tsai's Head Eye Calibration Algorithm 3.14
3.4.1 Using Denavit - Hartternberg's Approach to Establish a Body-Attached Coordinate Frame for Each Link of the Manipulator 3.16
3.5 Function of Procedures and Formats of Data Files 3.23
3.6 Experimental Results 3.26
3.7 Discussion 3.45
3.8 Conclusion 3.46
References 3.47
Appendix I Procedures 3.48

Chapter 4 A New Tracking Method for Shape from Motion Using an Active Vision System

4.1 Introduction 4.1
4.2 A New Tracking Method 4.1
4.2.1 Our approach 4.1
4.2.2 Using an Active Vision System to Track the Projective Basis Across Image Sequence 4.2
4.2.3 Using Projective Invariants to Track the Remaining Feature Points . 4.2
4.3 Using Factorisation Method to Recover Shape from Motion 4.11
4.4 Discussion and Future Research 4.31
References 4.32

Chapter 5 Experiments on Feature Tracking with 3D Projective Invariants

5.1 8-point Projective Invariant 5.1
5.2 Projective Invariant Based Tranfer between Distinct Views of a 3-D Scene .. 5.4
5.3 Transfer Experiments on the Image Sequence of an Calibration Block 5.6
5.3.1 Experiment 1. Real Image Sequence 1 of a Camera Calibration Block 5.6
5.3.2 Experiment 2. Real Image Sequence 2 of a Camera Calibration Block 5.15
5.3.3 Experiment 3. Real Image Sequence 3 of a Camera Calibration Block

. . . . . 5.22
5.3.4 Experiment 4. Synthetic Image Sequence of a Camera Calibration Block 5.27
5.3.5 Discussions on the Experimental Results 5.32
5.4 Transfer Experiments on the Image Sequence of a Human Face Model 5.33
References 5.44

Chapter 6 Conclusions and Future Researches

6.1 Contributions and Conclusions 6.1
6.2 Future Researches 6.1

Bibliography B.1

Acknowlements

It's my pleasure to express my gratitude to my supervisor, Dr. H. T. Tsui for his effective and inspring guidance throughout the course of this research work.

Part of this work was done in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Professor S. D. Ma provided opportunity for this work to be continued.

I am grateful to all of the people who have helped me with this work.

List of Symbols

Symbol	Explanation	Chapter
d_x	the center to center distance between adjacent sensor element in X (scan line) direction.	2
d_y	the center to center distance between adjacent CCD sensor in Y direction.	2
(C_x, C_y)	the row and column numbers of the center of computer frame memory.	2
(x_w, y_w, z_w)	3D coordinates of control points described in object world coordinate system.	2
(x, y, z)	3D coordinates of control points described in camera 3D coordinate system.	2
(X_u, Y_u)	the ideal (undistorted) image coordinate.	2
(X_d, Y_d)	the distorted image coordinate .	2
f	the effective focal length.	2
κ_1	the first term radial distortion coefficient.	2
N_{cx}	the number of sensor elements in the X direction.	2
N_{fx}	the number of pixels in a line as sampled by the computer.	2
X	$X_f - C_x$	2
Y	$Y_f - C_y$	2
R	the rotation matrix which represensts the rotion transformation from the object world coordinate system (x_w, y_w, z_w) to the camera 3D coordinate system (x, y, z).	2
T	the translation vector which represents the translation from the object world coordinate system (x_w, y_w, z_w) to the camera 3D coordinate system (x, y, z).	2
G_i	The head coordinate frame at the ith observing location.	3

C_i	The camera coordinate frame at the *i*th observing location.	3
CW	The calibration block world coordinate frame.	3
RW	The robot world coordinate frame.	3
$\boldsymbol{H}_{gi}$	The coordinate transformation from G_i to RW.	3
$\boldsymbol{H}_{ci}$	The coordinate transformation from CW to C_i.	3
$\boldsymbol{H}_{gij}$	The coordinate transformation from G_i to G_j.	3
$\boldsymbol{H}_{cij}$	The coordinate transformation from C_i to C_j.	3
$\boldsymbol{H}_{cg}$	The coordinate transformation from C_i to G_i.	3
$\boldsymbol{R}$	rotation matrix.	3
$\boldsymbol{r}_i = (r_{1i}, r_{2i}, r_{3i})^T$	the *i*th column vector of $\boldsymbol{R}$.	3
$[n_1, n_2, n_3]^T$	the unit vector of equivalent rotation axis of $\boldsymbol{R}$.	3
θ	the equivalent rotation angle of $\boldsymbol{R}$.	3
$\boldsymbol{P}_r$	principal axis of $\boldsymbol{R}$, $\boldsymbol{P}_r = 2\sin\frac{\theta}{2}[n_1, n_2, n_3]$ where $0 \le \theta \le \pi$	3
$\boldsymbol{P}_{gij}$	Principal axis or rotation axis for $\boldsymbol{R}_{gij}$.	3
$\boldsymbol{P}_{cij}$	Rotation axis for $\boldsymbol{R}_{cij}$.	3
$\boldsymbol{P}_{cg}$	Rotation axis for $\boldsymbol{R}_{cg}$.	3
$\boldsymbol{P}'_{cg}$	$\frac{1}{\sqrt{4-\lvert\boldsymbol{P}_{cg}\rvert^2}}\boldsymbol{P}_{cg}$	3
X, Y, A, D, E	the joints of the active vision system.	3
$(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$	the body-attached coordinate system for *i*th link.	3
θ_i	the angle between the adjacent links.	3

d_i	the distance between the adjacent links.	3
a_i	the length of the link *i*.	3
α_i	the joint angle of the link *i*.	3
${}^{i-1}\boldsymbol{A}_i$	the D-H transformation matrix for adjacent coordinate frames *i* and *i* ` 1.	3
${}^{0}\boldsymbol{T}_i$	the transformation from the *i*th coordinate frame to the robot base coordinate system.	3
$\boldsymbol{H}_{RC}$	the homogeneous transformation from the robot world frame *RW* to the camera calibration world frame *CW*	3
$\Delta\boldsymbol{H}_{ck}$	the error matrix between the measured camera pose relative to the robot world frame and the predicted camera pose relative to the robot world frame the *k* the observing position.	3
$[A, B, C, D]$	the cross ratio of four collinear points *A*, *B*, *C*, *D*.	4
(x_1, x_2, x_3)	the projective co-ordinates of a point *M* in the projective basis (*A*, *B*, *C*, *D*).	4
k_1, k_2, k_3	the cross ratios of three pencils of 4 lines going through *C*, *A*, and *B* respectively.	4
ψ_1, ψ_2	two distinct images of the same set of object points.	5
O, O'	camera centers for ψ_1, ψ_2.	5
υ, υ'	The epipoles υ and υ' are defined to be at the intersection of the line $\overline{OO'}$ with both image planes.	5
$\cong$	the equality up to a scale.	5
$\boldsymbol{p} \in \psi_1, \boldsymbol{p}' \in \psi_2$	two corresponding points.	5
$\boldsymbol{l}, \boldsymbol{l}'$	corresponding epipolar lines of $\boldsymbol{p}'$ and $\boldsymbol{p}$.	5
$[\upsilon']$	the skew-symmetric matrix of υ' (i.e., $[\upsilon']\boldsymbol{x} = \upsilon' \times \boldsymbol{x}$ for any vector x).	5
$\boldsymbol{A}$	a 3×3 projective transformation matrix ,which can transform ψ_1 to ψ_2.	5

$\boldsymbol{F}$	the fundamental matrix. The matrix $\boldsymbol{F}$ is a 3×3 matrix of rank 2.	5
$\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3$	three column vectors of $\boldsymbol{F}$.	5
$\boldsymbol{b}$	$[x_p x_{p'}, x_p y_{p'}, x_p, y_p x_{p'}, y_p y_{p'}, y_p, x_{p'}, y_{p'}, 1]^T$	5
$\boldsymbol{f}$	$[f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33}]^T$	5
$\boldsymbol{B}$	a 9×9 matrix.	5
ψ_{R_1} and ψ_{R_2}	two different reference image of a 3D scene.	5
ψ_T	a target image of the same scene.	5
(p_j, q_j)	eight corresponding points in ψ_{R_1}.	5
$(\bar{p}_j, \bar{q}_j)$	eight corresponding points in ψ_{R_2}.	5
(w_j, s_j)	eight corresponding points in ψ_T.	5
(p, q)	any other corresponding reference image point in ψ_{R_1}.	5
$(\bar{p}, \bar{q})$	any other corresponding reference image point in ψ_{R_2}.	5
(w, s)	target image point in ψ_T.	5

Chapter 1

Introduction

1.1 Active Vision Paradigm and Calibration of Active Vision System

1.1.1 Active Vision Paradigm

The concept of *active vision* was proposed by Aloimonos *et al.*[1],[11]. The main idea of this visual perception paradigm is that an observer has the ability to actively control the camera parameters. It has been shown that such an active observer can solve many basic visual problems in a much more efficient way than a passive one. One of the reasons for this is that additional constraints can be introduced by the activities of the observer.

1.1.2 A Review of the Existing Active Vision Systems

There is a quite good definition of an active vision system which is given by C. Fermüler, Y. Aloimonos [3] ,"an active vision system is actually a camera resting on a platform and being controlled through motors by a computer that has access to the images sened by the camera in real time. The platform can move freely in the envrionment. If this machine can fixate on targets being in relative motion to it, it can solve visual tasks in an efficient and robust manner".

Actually, an active vision system has one or more cameras mounting on a platform (or robot end effector) by which the camera motion can be controlled through servo motors by a computer.
Recently, considerable efforts have been devoted toward the development of active vision system. Several existing prototypes are listed as follows.

Institution	Developer	Finishing Year
University of Pennsylvania	Krotokov[4]	1988
University of Illinois	Abbot & Ahuja[5]	1990
MIT	Pogio[6]	1988
Unversity of Rochester	Brown[7]	1989
The Harvard University	Clark & Ferrier[8]	1988
Royal Institute of Technology in Sweden	Pahlavan & Eklundh[9]	1992
National Institute of Standards and Technology	Fiala *et al.*[10]	1993

Table 1.1 A number of existing prototypes of active vision system.

1.1.3 A Brief Introduction to Our Active Vision System.

Our active vision system is a camera platform with high accuracy which has 7 degrees of freedom. The active observer has two linear axes which can provide translational motion in the horizontal plane, one common pan rotary axis around which a pair of steoreo cameras can pan from side to side, one independent vergence axis and one independent tilt axis for each cameras. In our project, 5 degrees of freedom are available when we conduct research in the context of monocular vision. We are now equipped with a highly flexible and camera platform which have 7 degrees of freedom to move. Its accuracy enables us to use the reading of odometers as the known motion parameters in the active vision approach once calibrated precisely.

Figure 1.1 depicts the schematic configuration of the observer robot working in CVIP Lab, Electronic Engineering Department The Chinese University of Hong Kong.

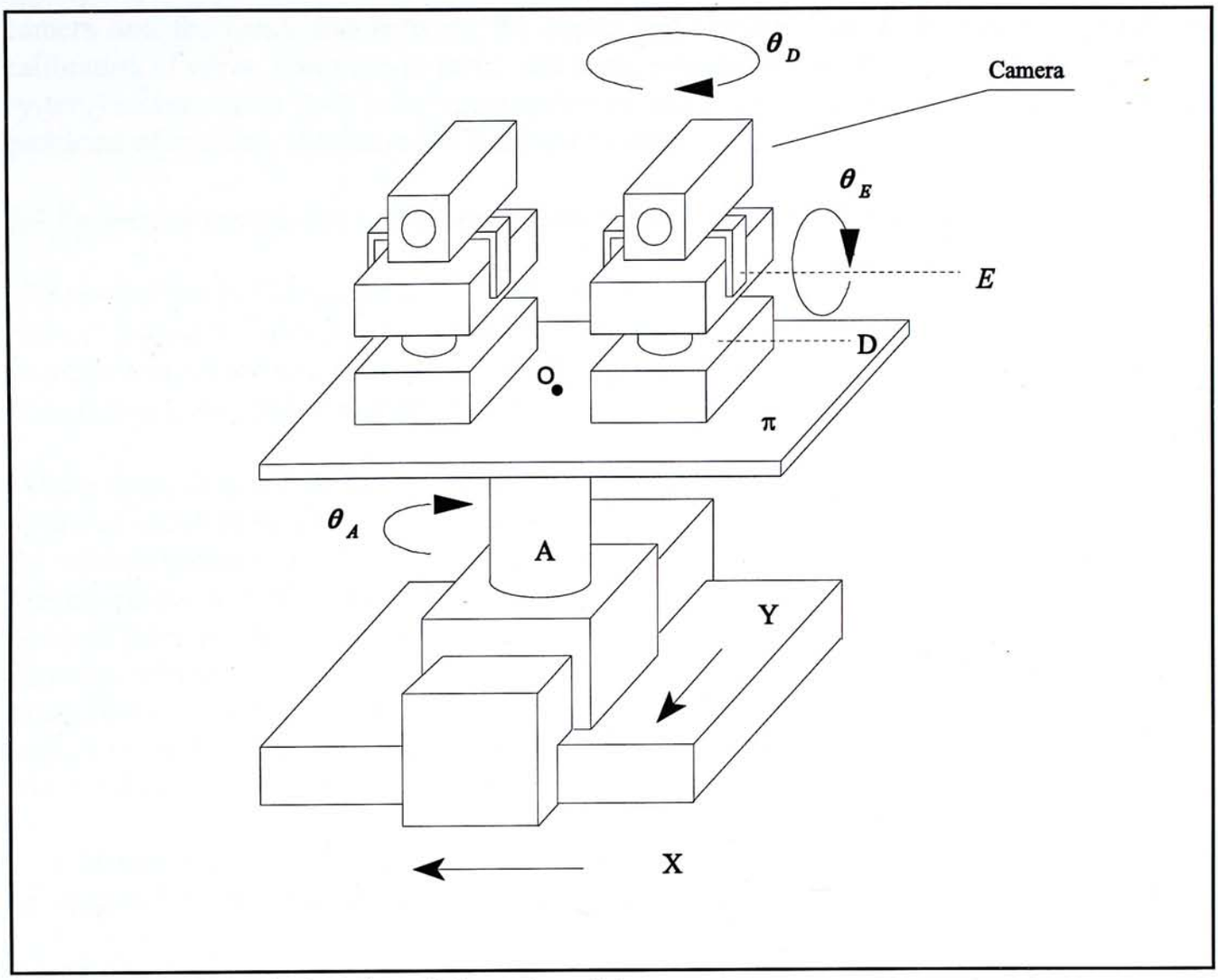


Figure 1.1 Schematic drawing of the active vision system

Table 1.2 shows the accuracy and the repeatability of the active vision system.

	Linear Axes	Rotational Axes
Accuracy	2.0 microns	0.02 degree
Repeatability	1.0 micron	0.01 degree

Table 1.2 Accuracy and Repeatability of the Active Vision System.

1.1.4 The Stages of Calibrating an Active Vision System

In order to have control over the motion of camera at will, it is necessary to know the relative position and orientation between the hand (robot end effector) and the robot base, between the

camera and the hand, and between the object and camera. These three tasks demand the calibration of robot, robot eye-to-hand, and camera. Because our observer robot (active vision system) distinguishes itself in its assemblying accurary , attention will be concentrated on the problems of camera calibration and hand-eye calibration.

1.2 Projective Invariants and Their Applications to Feature Tracking

As pointed out by Felix Klein in1872 [13], the basis for geometry is the set of properties which remain invariant under a given transformation group. Being defined as a set of geometric properties which remained unchanged under projective transformation, projective invariants had been derived and applied in many vision tasks[2].

Many shape from motion methods depend on the accurate tracking of feature points in an image sequence acquired by a moving camera. In Chapter 4, we present a new tracking algorithm for the feature points using projective invariants. After the correspondence of a small number of feature points between two images are established, no correspondence is required for the remaining points. These points can be mapped from one image to the other using plane projective invariant even if the points are not detected in the second image.This algorithm has been verified on real data acquired by an active vision system. Experiments of shape recovery from motion using a technique due to Tomasi and Kanade[12] are performed using our tracking algorithm. The results of both feature tracking and shape reconstruction are good.

A feature tracking method based on more general 8-point projective invariants is also developed. Experimental results on real image data are reported and analyzed .

1.3 Thesis Overview

The camera calibration for our active vision system is presented in Chapter 2. The head-eye calibration for our active vision system is discussed in Chapter 3. In Chapter 4, a new tracking method based on planar projective invariants is presented. In Chapter 5, we rederive the 8-point projective invariant, and analyze the performance of the feature tracking method based on 8-point projective invariant. In Chapter 6, we discuss several future research direcions.

References

[1] Aloimonos, I. Weiss and A. Bandyopadhyay, "Active vision" in *Proc. First Int. Conf. Comput. Vision*, June 1987, pp. 306-314.

[2] A.Shashua, "Projective structure from uncalibrated images: structure from motion and recognition," In *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 8, pp. 778-790, 1994.

[3] C. Fermüler, Y. Aloimonos, "The role of fixation in visual motion analysis," *Int. J. Comput. Vision*, 11:2, pp. 165-186, 1993.

[4] E. Krotkov, F. Fuma, J. Summers, "An agile stereo camera system for flexible image aquisition," *IEEE J. Robot. Autom.*, vol.4, no. 1, pp. 108-113,1988.

[5] A. L. Abbot, N. Ahuja, "Active surface reconstruction by integrating focus, vergence, stereo, and camera calibration," *Proc. 3rd Intern. Conf. Comput. Vis. Osaka, Japan*, pp. 489 - 492, 1990.

[6] T. Poggio *et al.*, "The MIT vision machine," *Proc. DARPA Image Understanding Workshop*, pp.177-198, 1988.

[7] C. M. Brown, "Kinematic and 3D motion prediction for gaze control," *Proc. Workshop on Interpretation of 3D scenes*, Austin, TX, pp. 145-141, 1989.

[8] J. J. Clark, N. J. Ferrier, "Modal control of atterntive viison system," *Proc. 2nd Intern. Conf. Comput. Vis.*, Tampa, FL, pp. 514-523, 1988.

[9] K. Pahlavan, J. O. Eklundh, "A head-eye system - Analysis and design," *Comput. Vis., Graph., Image Process.: Image Understanding Workshop*, vol. 56, no. 1, pp. 41-56, 1992.

[10] J. C. Fiala *et.al* , "TRICLOPS: A tool for studying active vision," *Int. J. Comput. Vision*, vol. 12, no. 2/3, pp. 231-250, 1994.

[11] Y.Aloimonos,I.Weiss,and A. Bandopadhay, "Active vision", *Int. J. Comput. Vision* 2(1), 1988, 333-356

[12] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method," *Technical Report CMU-CS-91-172*, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[13] Marvin J. Greenberg , Euclidean and Non-Euclidean Geometry , Development and History, W. H. Freeman, San Francisco, 1980, pp. 367.

Chapter 2

Calibration for an Active Vision System: Camera Calibration

2.1 An Overview of Camera Calibration

The problem we address in this chapter is the camera calibration for our active vision system. As an important issue in computer vision, camera calibration has attracted much of the attention of the vision community over the last decade. The calibration problem can be formulated briefly as follows: given a set of 3D coordinates of control points and their corresponding images on the camera plane, "camera calibration is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and /or 3D position and orientation of the camera frame relative to a certain world coordinate system."[5]. The camera calibration problem has long been considered as a crucial stage of the overall calibration of the active vision system since its accuracy determines directly the accuracy of the ensuing hand-eye calibration stage.

Existing camera calibration techniques can be categorized into two groups, group A using pin-hole model and group B adopting two planes model. R. Tsai gave a brief cross comparison between the existent appoaches at that time, and we summarize it in the Table 2.1 and Table 2.2.

	Group A Pin-hole Model				Group B Two Plane model
	Category I	**Category II**	**Category III**	**Category IV**	
Description	Techniques Involving Full-Scale Nonlinear Optimizatio n	Techniques Involving Computing Perspective Transformatio n Matrix First Using Linear Equation.	Geometric Technique	RAC Based Two Stage method. RAC is the abbreviation of Radial Alignment Constraint.	Two-Plane Model
Advantage	It allows easy adaptation of any arbitrarily accurate yet complex model for imaging.	No nonlinear optimization is needed	No nonlinear search is needed.	Radial alignment constraint is used to reduce the dimensionality of the unknown parameter space. 1) Radial distortion of lense is considered. 2) s_x (uncertainty factor) and most of extrinsic parameters (relative rotation and translation between the camera and calibration point) except T_z, can be obtained in a simple and efficient way. Only the calculation of f(effective focal length),κ_1 (distortion coefficient) and T_z, involves nonlinear search.	Only linear equations need be solved.

Table 2.1

	Group A Pin-hole Model				Group B Two Plane model
	Category I	**Category II**	**Category III**	**Category IV**	
Limitations	A good initial guess is required to start the non-linear search.	1) Lens distortion can not be considered. 2)Unknowns to be solved are not linearly independent.	1) No lens distortion can be considered. 2) Focal length is assumed given. 3) Uncertainty of image scale factor is not allowed.	RAC method does not give the global solution in the case of noisy data. [6]	1) The number of unknowns is at least 24(12 for each plane),much higher than the degrees of freedom. 2) The formula used for the transformation between image and object coordinates is empirically based only.
Accuracy	Using faig's technique[3] , the rms error can as good as 0.1 mil.	Yakimovsky and Cunningham [8] reported that the 3D measurement accuracy of ± 5 mm at a distance of 2m.		A Fuji 25 mm camera was calibrated using a one inch by one inch calibration pattern .The total range of depth in the camera coordinate system is about from 4 in to 4.5 in.Average error is 0.5 mil and maximum error is 1.4 mil.	Martin's *et al.* [4] reported that the average error is about 4 mil with a distance of 25 in.

Table 2.2

Recently, many efforts have been devoted to the research into the field of camera calibration. Among them, Wei and Ma [7] generalized the two plane method in terms of the generalized projective mappings between image planes and two calibration planes. After comparing the experimental results, they argues that classic methods outperform Tsai's RAC method and their method outperforms the classic ones in terms of yielding better accuracy in both the 3-D measurement and the generation of image coordinates. This method is more computational intensive than the pin hole model based method owing to the more unknowns involved.

Camera calibration techniques based on the properties of vanishing point and/or vanishing line has been developed[2][1]. Caprile and Torre[2] proposed a method in which focal length and principal point can be recovered from a single image of a cube. Then the extrinsic parameters which describe the rigid motion between the two camera coordinate systems can be obtained from an image stereo pair of a suitable planar pattern. Beardley and Murray[1] reported a method with which the principal point and focal length of the given camera can be solved for. It is pointed out that the vanishing point based camera techniques degrades in longer focal length cameras.

In our research project, we chose to make use of Tsai's [5] RAC based camera calibration technique for the following reasons:

1. Radial distortion can be considered in Tsai's method so that adequate accuracy in resulting parameters can be ensured.

2. Most of extrinsic parameters can be solved for linearly, only three parameters (f, κ_1, T_z) entail non linear search.

3. A camera calibration package has been developed by Reg. Willson. It has been testified by many computer vision researchers such that the calibration result is reliable.

In Tsai's camera calibration technique, The knowledge of the following intrinsic parameters is assumed:

Principal point C_x, C_y (intersection between optic axis and image plane).

Center to center distance between adjacent sensor elements, d_x , d_y

Number of sensor elements in the X direction, N_{cx}.

Number of pixels in a line as sampled by the image aquisition machanism, N_{fx}.

All of them but C_x, C_y should be provided by the camera manufacturer.

2.2 Tsai's RAC Based Camera Calibration Method

Figure 2.1 describes the camera geometry with perspective projection and radial lens distortion.

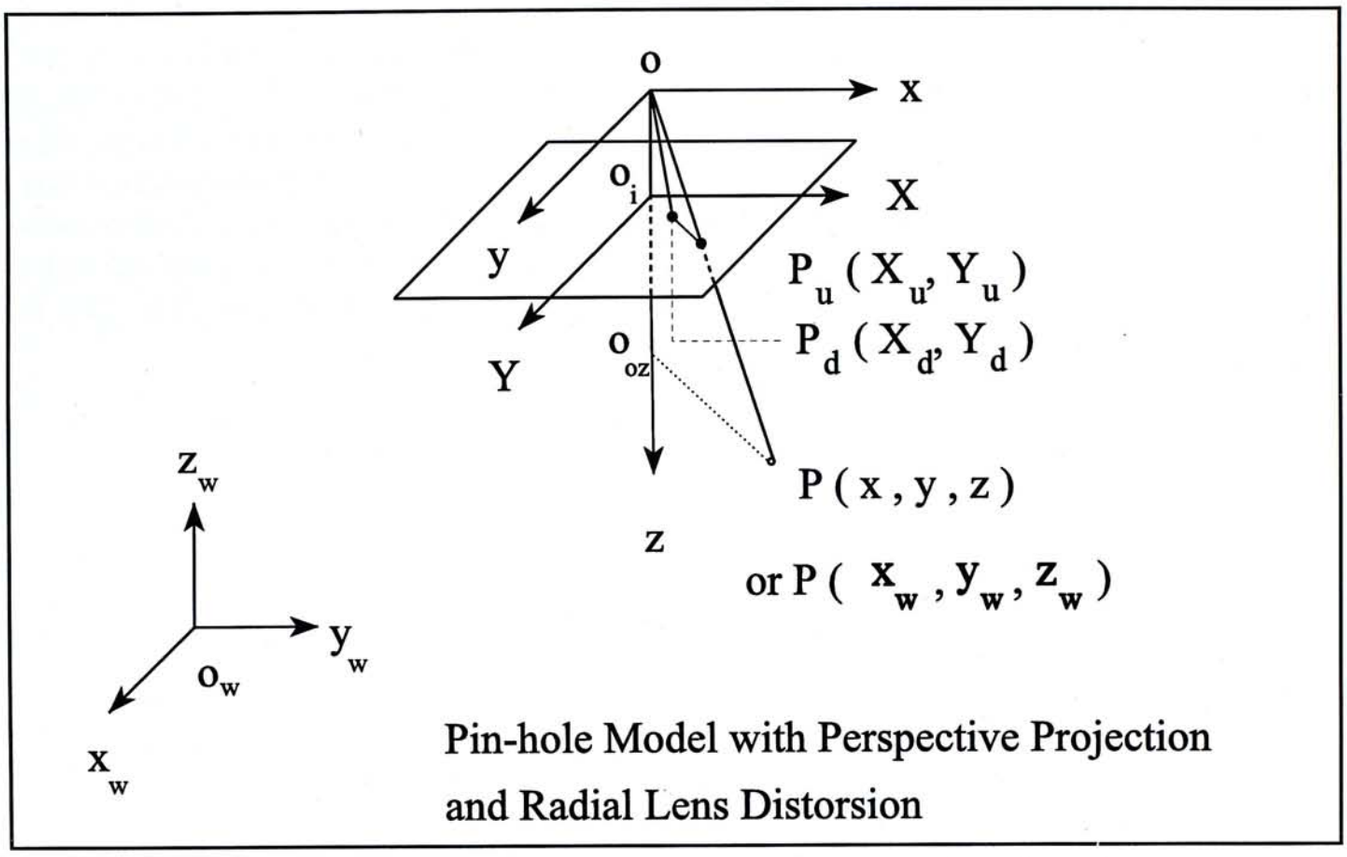


Figure 2.1 Pin-hole model with perspective projection and radial lens distorsion.

In R. Y. Tsai's two stage camera calibration algorithm, radial alignment constraint is used to reduce the dimensionality of the unknown parameter space. Radial distortion of lens, which is the dominant part of distorsion of lens, is considered in that approach. In the first stage of the approach, s_x (uncertainty factor) and most of extrinsic parameters (relative rotation and translation between the camera and calibration point) except T_z can be obtained in a simple and efficient way on the basis of RAC (radial alignment constraint). In the second stage, the calculation of f (effective focal length), κ_1 (distortion coefficient) and T_z involves nonlinear search.

The Radial Alignment Constraint is derived from the observation that radial distortion of a lens does not alter direction of vector from principal point of the image plane to image points, which leads to

$O_iP_u // O_iP_d // O_{oz}P$.

That is $O_{oz}P$, which is the perpendicular from the 3D object point $P(x, y, z)$ to the optical axis , must be parallel to O_iP_d, which is the perpendicular drawn from distorted image point (X_d, Y_d) to the optical axis. The point O_{oz} is the intersection of $O_{oz}P$ with the optical axis. This parallelism leads to a zero cross product of the direction vectors of two straight lines and yields the RAC constraint immediately.

Various coordinate systems which are involved in the process of calibrating a camera are illustrated in Figure 2.1. Among them, O_w-X_w-Y_w-Z_w is the 3D world coordinate system. *o-x-y-z* is 3D camera coordinate system where o is the optical center, z axis coincides with optical axis , and frontal image plane is perpendicular to the z axis and principal point O_i is the intersection between them. Distance OO_i is the focal length. O_i-X-Y is the image coordinate system. (X_u , Y_u) is the ideal image coodinate. (X_d , Y_d) is the actual image coordinate which differs from (X_u , Y_u). (X_f, Y_f) is the corresponding frame buffer coordinate.

2.2.1 The Pinhole Camera Model with Radial Distortion

The image formation process can be modelled as a transformation from 3D world coordinate system to computer frame buffer coordinate system which can be broken down into the following four subtransformations:

1. Rigid body transformation from the object world coordinate system (x_w, y_w, z_w) to the camera 3D coordinate system (x, y , z). This Transformation is an Euclidean Transformation.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{2.1}$$

Where R is the rotation matrix and T is the translation vector

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} , \quad T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{2.2}$$

R and T are the extrinsic prarameters to be calibrated.

2. Transformation from 3D camera coordinate (x, y, z) to ideal (undistorted) image coordinate (X_u, Y_u) using perspective projection with pinhole camera geometry. This transformation is a perspective transformation.

$$X_u = f\frac{x}{z} \qquad Y_u = f\frac{y}{z} \tag{2.3}$$

The parameter to be calibrated is the effective focal length f.

3. Transformation from ideal (undistorted) image coordinate (X_u, Y_u) to distorted image coordinate (X_d, Y_d) . Only radial components of lens distortion is considered in this transformation.

$$
\begin{aligned}
X_d + D_x &= X_u \qquad Y_d + D_y = Y_u \\
D_x &= X_d(\kappa_1 r^2 + \kappa_2 r^4 + \cdots) \\
D_y &= Y_d(\kappa_1 r^2 + \kappa_2 r^4 + \cdots) \\
r &= \sqrt{X_d^2 + Y_d^2}
\end{aligned}
\tag{2.4}
$$

Where (X_d, Y_d) is the distorted or true image coordinate on the image plane. The parameters to be calibrated are distortion coefficient. Only one term radial distortion are considered in Tsai's method.

4. Transformation from real image coordinate (X_d , Y_d) to computer image coordinate (X_f, Y_f).

$$
\begin{aligned}
X_f &= s_x d_x'^{-1} X_d + C_x \\
Y_f &= d_y^{-1} Y_d + C_y \\
d_x' &= d_x \frac{N_{cx}}{N_{fx}}
\end{aligned}
\tag{2.5}
$$

Where

(X_f, Y_f) denotes row and column numbers of the image pixel in computer frame memory,

(C_x , C_y) indicates row and column numbers of the center of computer frame memory,

d_x is the center to center distance between adjacent sensor element in X (scan line) direction,

d_y is the center to center distance between adjacent CCD sensor in Y direction,

N_{cx} represents the number of sensor elements in the X direction,

N_{fx} represents the number of pixels in a line as sampled by the computer.

The parameter to be calibrated is the uncertainty image scale factor s_x.

Let $X = X_f - C_x$, $Y = Y_f - C_y$, then,

the equations relating the 3D camera coordinates to the 2D computer image coordinates can be obtained by combining the last three steps, the (X, Y) computer coordinate is related to (x, y, z), the 3D coordinate of the object point in camera coordinate system by the following equations:

$$s_x^{-1} d_x' X + s_x^{-1} d_x' X \kappa_1 r^2 = f\frac{x}{z}$$

$$d_y Y + d_y Y \kappa_1 r^2 = f\frac{y}{z} \qquad (2.6)$$

where

$$r = \sqrt{(s_x^{-1} d_x' X)^2 + (d_y Y)^2}$$

Substituting (2.1) into (2.6) gives the equations relating the 3D world coordinates to the 2D computer image coordinates:

$$s_x^{-1} d_x' X + s_x^{-1} d_x' X \kappa_1 r^2 = f \frac{r_1 x_w + r_2 y_w + r_3 z_w + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_z}$$

$$d_y Y + d_y Y \kappa_1 r^2 = f \frac{r_4 x_w + r_5 y_w + r_6 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \tag{2.7}$$

where

$$r = \sqrt{(s_x^{-1} d_x' X)^2 + d_y Y^2}$$

The above set of equations characterize the image formation process using a pinhole model with one term radial lens distorsion.

2.2.2 Calibrating a Camera Using Monoview Noncoplanar Points

In our project, control points scatter on two mutually orthogonal plane. The calibration techniques using a noncoplanar set of calibration points should be used. Tsai's calibration approach solves the calibration problem in two stages. In the first stage, the linear matrix equation derived from the RAC yields solutions for seven unknowns from which one intrinsic parameter s_x and most of the extrinsic parameters (rotation matrix R , components of translation vector in x, y direction T_x and T_y) can be solved for. In the second stage, effective focal length f , one term radial distorsion coefficient κ_1 and T_z are calculated using non-linear optimization technique.

Stage 1. *Compute 3D Orientation, Position (x and y) and Scale Factor*

a) Compute alternate distorted image coordinate (X_d', Y_d') from (X_f, Y_f).

Rearranging (2.5) yields

$$X_{di} = s_x^{-1} d_x' (X_{fi} - C_x)$$

$$Y_{di} = d_y (Y_{fi} - C_y) \tag{2.8}$$

for i = 1 , … , N , where N is the total number of calibration points.

Let $X_d' = s_x X_d$, $Y_d' = Y_d$,

the alternate distorted image coordinates can be computed as following

$$X_{di}' = d_x'(X_{fi} - C_x)$$
$$Y_{di}' = d_y(Y_{fi} - C_y) \tag{2.9}$$

b) Compute $T_y^{-1}s_xr_1$, $T_y^{-1}s_xr_2$, $T_y^{-1}s_xr_3$, $T_y^{-1}s_xT_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$, $T_y^{-1}r_6$:

The following relation was derived from the radial alignment constraint that constrains the outer product of two vectors(x, y, 0) and (X, Y, 0) to be zero. we have

$$\frac{Y_d}{X_d} = \frac{y}{x} \tag{2.10}$$

That is

$$\frac{s_x Y_d'}{X_d'} = \frac{y}{x} \tag{2.11}$$

For each calibration point i with (x_{wi}, y_{wi}, z_{wi}) as the 3D world coordinate and (X_d', Y_d') as the modified image coordinate computed in a) above, set up the following linear equation with $T_y^{-1}s_xr_1$, $T_y^{-1}s_xr_2$, $T_y^{-1}s_xr_3$, $T_y^{-1}s_xT_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$, $T_y^{-1}r_6$ as unknowns:

$$\begin{bmatrix} Y_{di}'x_{wi} & Y_{di}'y_{wi} & Y_{di}'z_{wi} & Y_{di}' & -X_{di}'x_{wi} & -X_{di}'y_{wi} & -X_{di}'z_{wi} \end{bmatrix} \begin{bmatrix} T_y^{-1}s_x r_1 \\ T_y^{-1}s_x r_2 \\ T_y^{-1}s_x r_3 \\ T_y^{-1}s_x T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \\ T_y^{-1}r_6 \end{bmatrix} = X_{di}' \tag{2.12}$$

With N (the number of object points) much larger than seven, an overdetermined system of linear equations can be established and solved for the seven unknowns $T_y^{-1}s_x r_1$, $T_y^{-1}s_x r_2$, $T_y^{-1}s_x r_3$, $T_y^{-1}s_x T_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$ and $T_y^{-1}r_6$.

c) Compute ($r_1, \cdots, r_9, T_x, T_y$) from $T_y^{-1}s_x r_1$, $T_y^{-1}s_x r_2$, $T_y^{-1}s_x r_3$, $T_y^{-1}s_x T_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$ and $T_y^{-1}r_6$.

1) Compute $|T_y|$ from $T_y^{-1}s_x r_1$, $T_y^{-1}s_x r_2$, $T_y^{-1}s_x r_3$, $T_y^{-1}s_x T_x$, $T_y^{-1}r_4$, $T_y^{-1}r_5$ and $T_y^{-1}r_6$.

Let a_i, $i = 1, \cdots, 7$ be defined as $a_1 = T_y^{-1}s_x r_1$, $a_2 = T_y^{-1}s_x r_2$, $a_3 = T_y^{-1}s_x r_3$, $a_4 = T_y^{-1}s_x T_x$, $a_5 = T_y^{-1}r_4$, $a_6 = T_y^{-1}r_5$ and $a_7 = T_y^{-1}r_6$. Note that all a_i, $i = 1, \cdots, 7$ are determined in b). Compute $|T_y|$ using the following formula:

$$|T_y| = (a_5^2 + a_6^2 + a_7^2)^{-1/2} \tag{2.13}$$

2) Determine s_x:

Using the following formula to compute s_x:

Note that

$$s_x = (a_1^2 + a_2^2 + a_3^2)^{1/2} |T_y| \tag{2.14}$$

3) Determine the sign of T_y :

i) pick an object point i whose computer image coordinate (X_{fi}, Y_{fi}) is away from the image center (C_x, C_y); the object world coordinate is (x_{wi}, y_{wi}, z_{wi}).

ii) Pick the sign of T_y to be + 1.

iii) Compute the following

$$r_1 = (T_y^{-1} s_x r_1) \cdot (T_y s_x^{-1}) \qquad r_2 = (T_y^{-1} s_x r_2) \cdot (T_y s_x^{-1})$$

$$r_3 = (T_y^{-1} s_x r_3) \cdot (T_y s_x^{-1}) \quad r_4 = (T_y^{-1} r_4) \cdot T_y$$

$$r_5 = (T_y^{-1} r_5) \cdot T_y \qquad r_6 = (T_y^{-1} r_6) \cdot T_y$$

$$T_x = (T_y^{-1} s_x T_x) \cdot (T_y s_x^{-1})$$

$$x = r_1 x_w + r_2 y_w + r_3 z_w + T_x \qquad y = r_4 x_w + r_5 y_w + r_6 z_w + T_y$$

where $T_y^{-1} s_x r_1$, $T_y^{-1} s_x r_2$, $T_y^{-1} s_x T_x$, $T_y^{-1} r_4$ and $T_y^{-1} r_5$ are determined in b)

iv) IF ((x and X have the same sign) and (y and Y have the same sign)), THEN $\mathrm{sgn}(T_y) = +1$, ELSE $\mathrm{sgn}(T_y) = -1$.

4) Compute the 3D rotation matrix R, or r_1, r_1, ···, r_9:

Compute r_1, r_2, r_3, r_4, r_5, r_6 and T_x with the following formula:

$$r_1 = a_1 \cdot T_y / s_x \quad r_2 = a_2 \cdot T_y / s_x \quad r_3 = a_3 \cdot T_y / s_x$$

$$r_4 = a_5 \cdot T_y \qquad r_5 = a_6 \cdot T_y \qquad r_6 = a_7 \cdot T_y$$

$$T_x = a_4 \cdot T_y / s_x$$

where a_i, $i = 1, \cdots, 7$ are defined in (1) and are the seven variables computed in b).

Given r_i, $i = 1, \cdots, 6$, which are the elements in the first two rows of R, the third row of R can be computed as the cross product of the first two rows, using the orthonormal property of R and the right-handed rule (determinant of $R = 1$, not -1)

Stage 2. *Compute Effective Focal Length, Distortion Coefficients, and z position*

a) Compute an approximation of f and T_z by ignoring lens distortion

For each calibration point i, establish the following linear equation with f and T_z as unknowns:

$$\begin{bmatrix} y_i & -d_y Y_i \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} = w_i d_y Y_i$$

where (2.15)

$$y_i = r_4 x_{wi} + r_5 y_{wi} + r_6 \cdot z_{wi} + T_y$$

$$w_i = r_7 x_{wi} + r_8 y_{wi} + r_9 \cdot z_{wi}$$

With several object calibration point, this yields an overdetermined system of linear equations that can be solved for the unknowns f and T_z. The control points must not be allowed to distributed on a plane which is exactly parallel to the image plane, otherwise (15) becomes linearly dependent.

b) Compute the exact solution for f, T_z, κ_1

Solve (2.7) with f, T_z, κ_1 as unknowns using standard optimization scheme such as steepest descent. Use the approximation for f and T_z computed in d) as initial guess, and zero as the initial guess of κ_1.

2.3 Reg Willson's Implementation of R. Y. Tsai's RAC Based Camera Calibration Algorithm

In our project, the camera calibration package developed by Reg Willson was used to calibrate the camera mounted on the active vision system. This camera calibration package contains a fully self contained implementation of Roger Tsai's camera calibration algorithm using public domain MINPACK optimization routines. A brief specification of this camera calibraiton package is shown in Figure 2.2.

In order to use the Tsai's RAC calibration package developed by CMU, several parameters needed to be set in advance. The subroutine added in camera.c are given below:

```
void initialize_pulnix_tm6_c540_parms( )
{

  strcpy(camera_type, "TM6+C540" );

  cp.Ncx = 752;                          /* [sel] Number of sensor elements in camera's x direction */

  cp.Nfx = 512;                          /* [pix]  Number of pixels in frame grabber's x direction */

  cp.dx = 0.0086;                        /* [mm/sel] X dimension of camera's sensor element (in mm) */

  cp.dy = 0.0083;                        /* [mm/sel] Y dimension of camera's sensor element (in mm)  */

  cp.dpx =  cp.dx * cp.Ncx / cp.Nfx ;     /* [mm/pix] Effective X dimension of pixel in frame grabber */

  cp.dpy = cp.dy;                                /*[mm/pix] Effective Y dimension of pixel in frame
                                                  grabber */

  cp.Cx = 512/2;                         /*[pix]  Z axis intercept of camera coordinate system   */

  cp.Cy =  512/2;                        /*[pix]    Z axis intercept of camera coordinate system  */

  cp.sx  = 1.0                           /* []      Scale factor to compensate for any error in dpx  */
}
```

Then, we replace initialize_photometrics_parms( ) by initialize_pulnix_tm6_c540_parms in the following files:

C_CAL.C	N_CAL.C
C_CAL_FO.C	N_CAL_FO.C

C_SYNTHETIC.C N_SYNTHETIC.C
CAMERA_CALIBRATION.C

Camera Calibration using Tsai's Method - revision 2.1

Reg Willson <rgwillson@mmm.com> 04-Jun-94

This revision includes a fully self contained implementation of Roger Tsai's camera calibration algorithm using *public domain* MINPACK optimization routines (the code may also be built using commercial IMSL optimization routines). Tsai's algorithm is documented in several places including:

"A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", Roger Y. Tsai, Radiometry -- (Physics-Based Vision), L. Wolff, S. Shafer, G. Healey, eds., Jones and Bartlett, 1992,

"A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", Roger Y. Tsai, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pages 323-344.

"An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", Roger Y. Tsai, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pages 364-374.

The *actual calibration code* is contained in the files:

camera_calibration.c
camera_calibration.h
extrinsic_calibration.c
cc_utils.c
matrix.c

The subdirectory minpack contains the FORTRAN source code for the MINPACK routines:

dpmpar.f, enorm.f, fdjac2.f, lmdif.f, lmpar.f, qrsolv.f.

Five *test programs* that make use of the calibration code are:

c_cal.c	basic coplanar calibration
c_cal_fo.c	coplanar calibration, full optimization
n_cal.c	basic noncoplanar calibration
n_cal_fo.c	noncoplanar calibration, full optimization
ep_cal.c	extrinsic parameter calibration

Two programs are included to generate *synthetic data* to test the calibration code:

c_synthetic.c	coplanar calibration data generation program
n_synthetic.c	noncoplanar calibration data generation program
gasdev.c	random number routine for above

Four *test data* files are also included:

c_test.cd.data	synthetic coplanar test data
n_test.cd.data	synthetic noncoplanar test data
c_test.cpcc.data	calibrated camera model from coplanar test data
n_test.cpcc.data	calibrated camera model from noncoplanar test data

Two *makefiles* are provided to build the code:

makefile.IMSL	to use with the IMSL optimization routines
makefile.MINPACK	to use with the MINPACK optimization routines

Figure 2.2 Brief description of Reg Willson's implementation of R. Y. Tsai's two stage RAC based camera calibration approach.

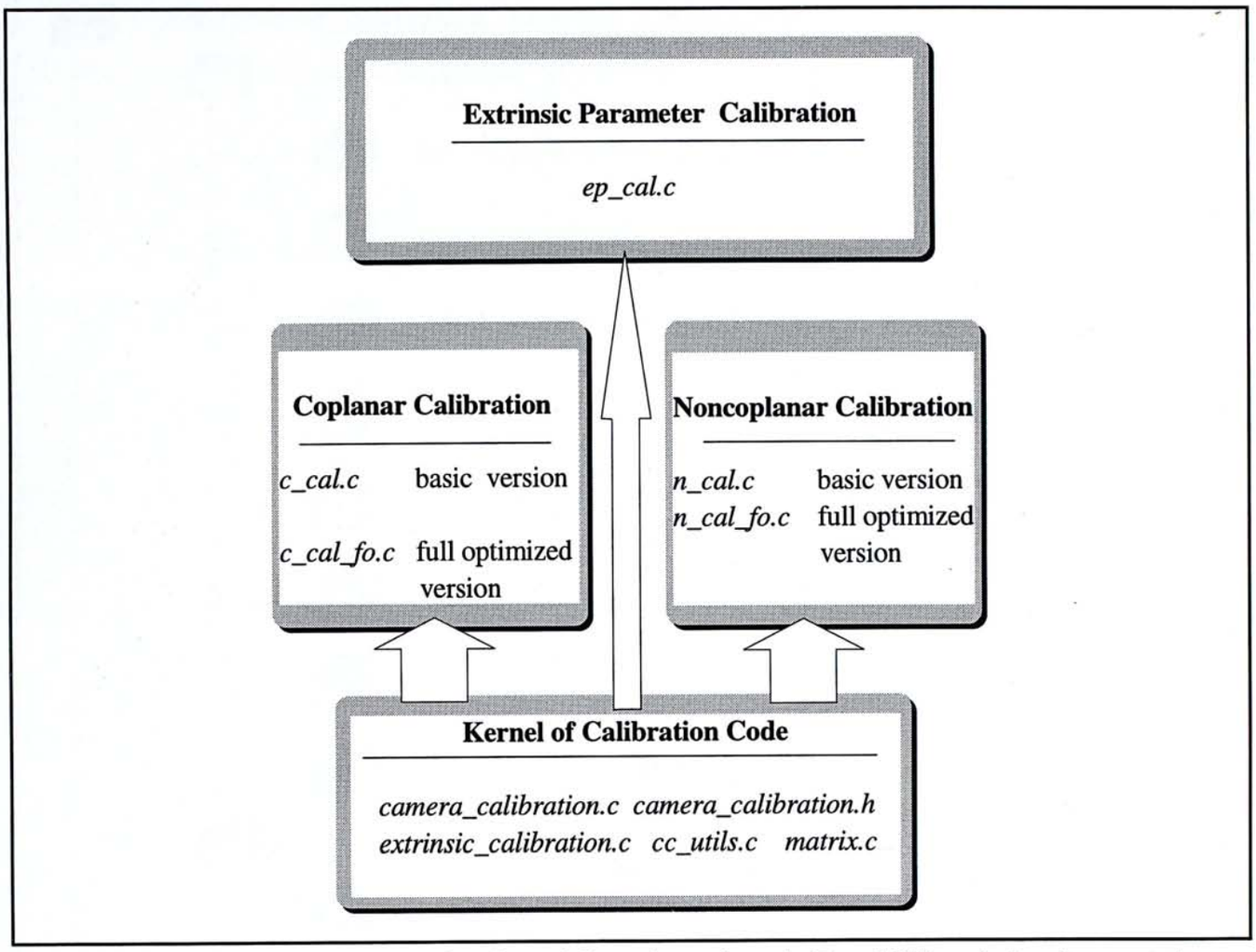


Figure 2.3 Relations and dependencies of the subroutines in Reg Willson's implementation.

Relations and dependcies of the various subroutines are illustrated in Fig. 2.3. The program we employed in camera calibration is n_cal_fo.c. It calls a subroutine named noncoplanar_calibration_with_full_optimation in its execution. The call graph of noncoplanar _calibration_with_full_optimization is shown in Fig 2.4. The subrountine ncc_three_parm_optimization estimates f, T_z, κ_1 in wich lmdif, a MINPACK optimization routines is called to solve a non-linear optimization problem.

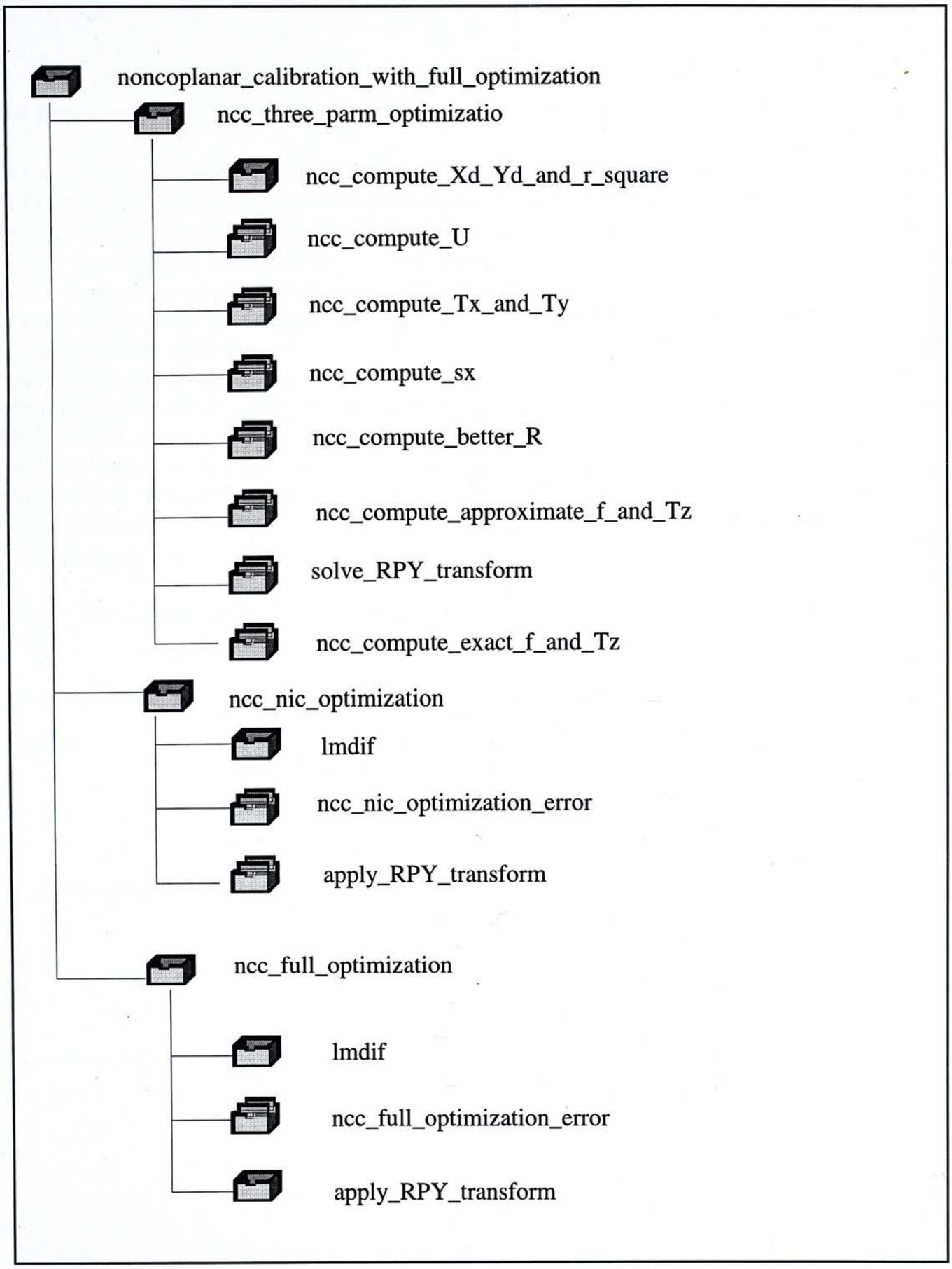


Figure 2.4 The call graph of noncoplanar_calibration with full optimizations, which is the kernel subroutine for noncoplanar calibration in Reg Willson's implementation.

2.4 Experimental Setup and Procedures

In the process of camra calibration, the world coordinate system is attached to a reference block. A PULNiX TM 6 CCIR CCD camera fixed rigidly on the end effector of the camera platform had been calibrated by employing the methods of R. Y. Tsai[5]. The calibration reference block (Fig. 2.5) is an accurately fabricated black aluminum cube with a 9x9 lattice of white discs on each of the three mutually orthogonal surfaces.The diameter of a white disc is 10.0mm. The Spacing separating the centres of the adjacent discs is 30.0mm. The positional accuracy is 0.1mm. several photographs of calibration reference block are shown in Fig. 2.6 During the process of experiments, the images of the calibration reference block were acquired from various viewing position and directions, taking advantage of the flexibility of our active vision system. A 16mm COSMICAR lens was used. After image segmentation, the centroids of ellipsis were used as the control points for the calibration procedures. With little loss in accuracy, the centroid of the ellipse can be taken as the projection of the centroid of the corresponding disc on the calibration reference block as the disc is very small comparing to the average distance between the camera and reference block.

Reg Willson's implementation requires the input data file formated as follows

	3D world coordinates of control points			computer image coordinates of control points	
50 control points	x_w	y_w	z_w	x_f	y_f
	⋮	⋮	⋮	⋮	⋮
	x_w	y_w	z_w	x_f	y_f

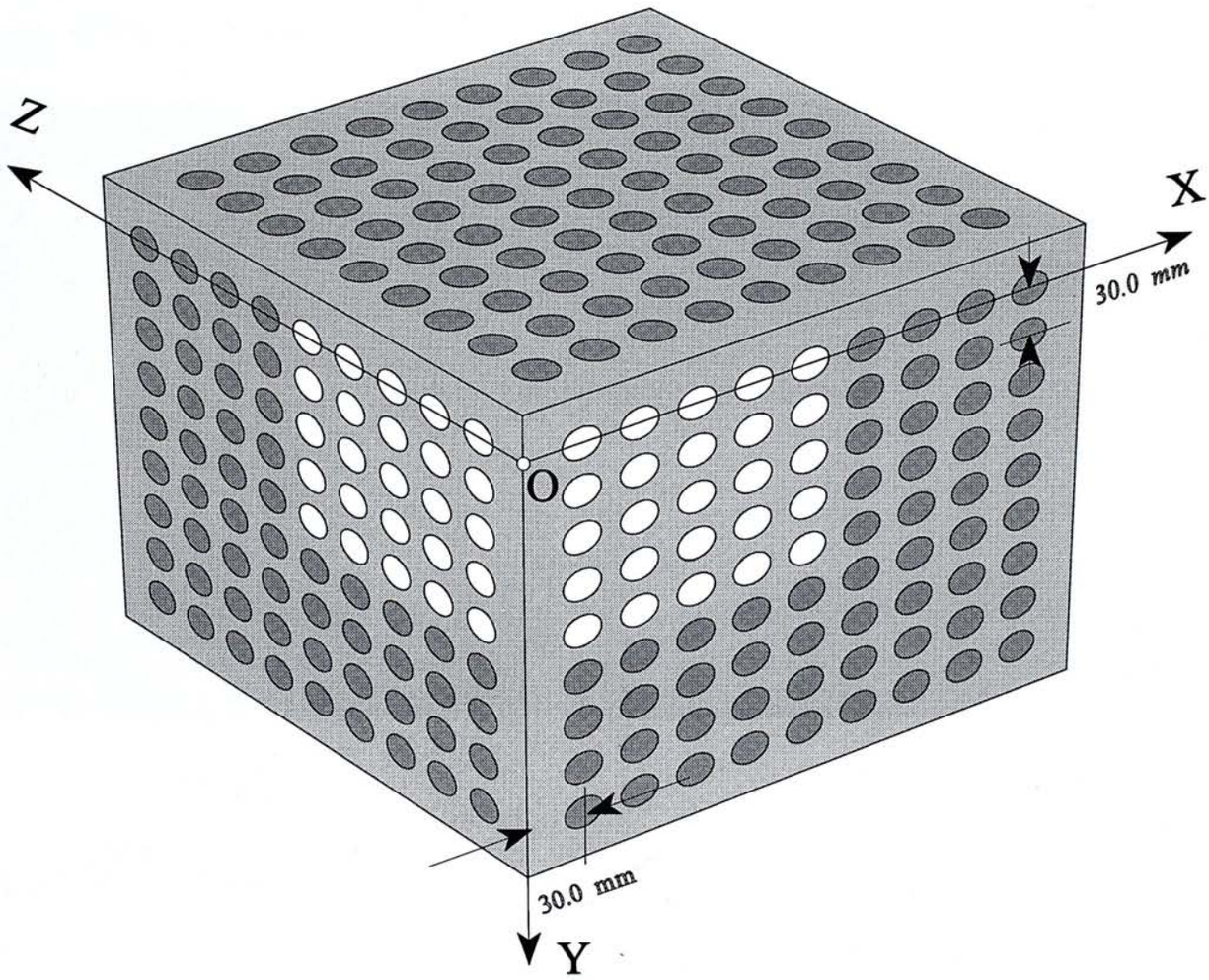


Figure 2.5 The schematic drawing of the calibration block. The centroids of the white circular discs are used as control points.

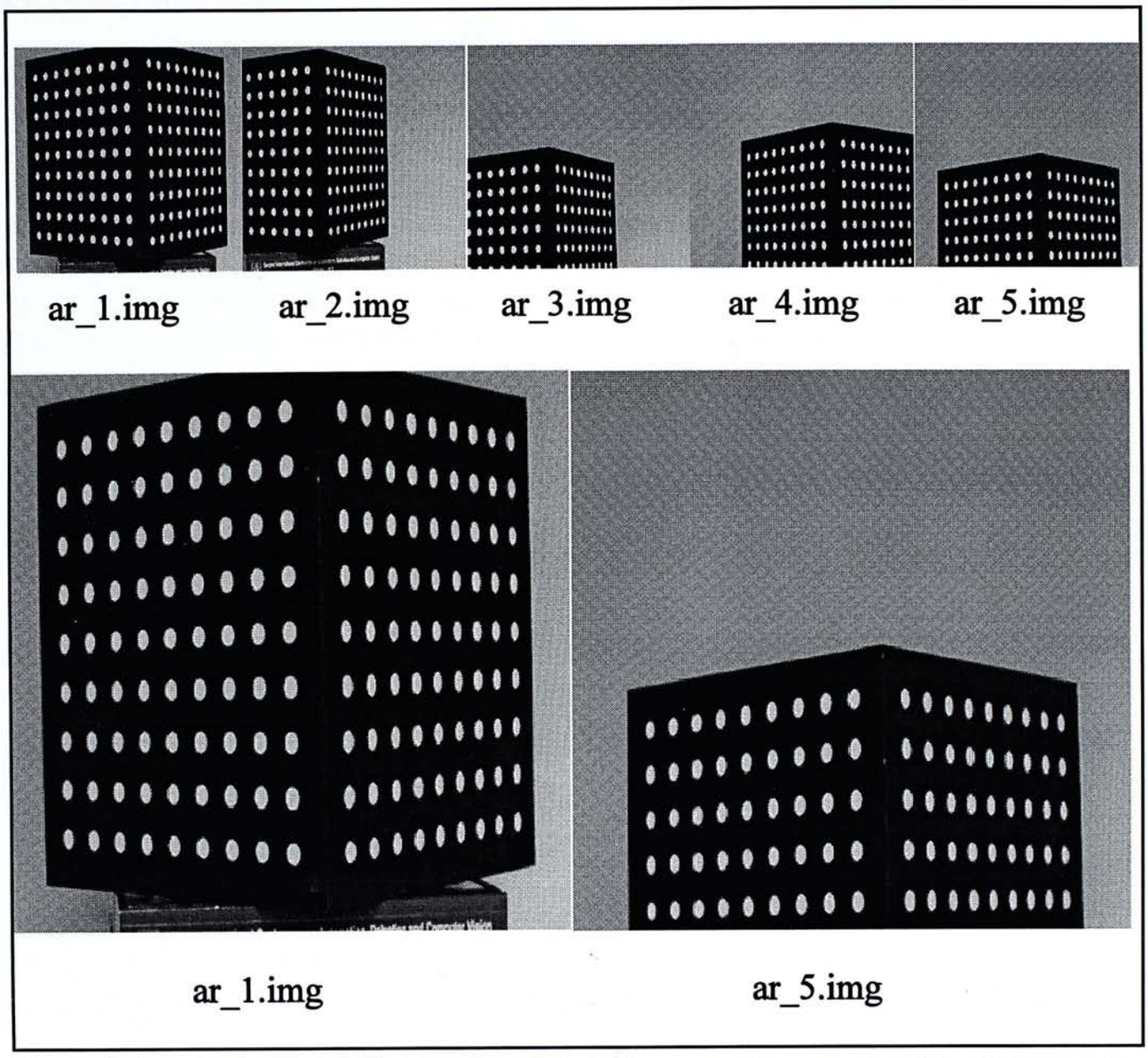


Figure 2.6 Images of calibration blocks which are used in camera calibration process

2.5 Experimental Results

Five images of calibration block are taken from different angle using the camera mounted on our active vision system. After detecting the centroids of the circular discs, we get the input data files which are required by Reg Willson' implementation. The contents of input data file for ar_1.img is shown in Table 2.1. Fifty control points are used in each situation. The Figure 2.7 shows the detected centroid of circular discs of ar_1.img. The output listing of Reg Willson's implementation of ar_1.img ... ar_5.img is illustrated in Table 2.2. The corresponding error statistics are presented in Table 2.3. The experimental results showed that the R. Y. Tsai's RAC based two stage camera calibration approach performed quite good.

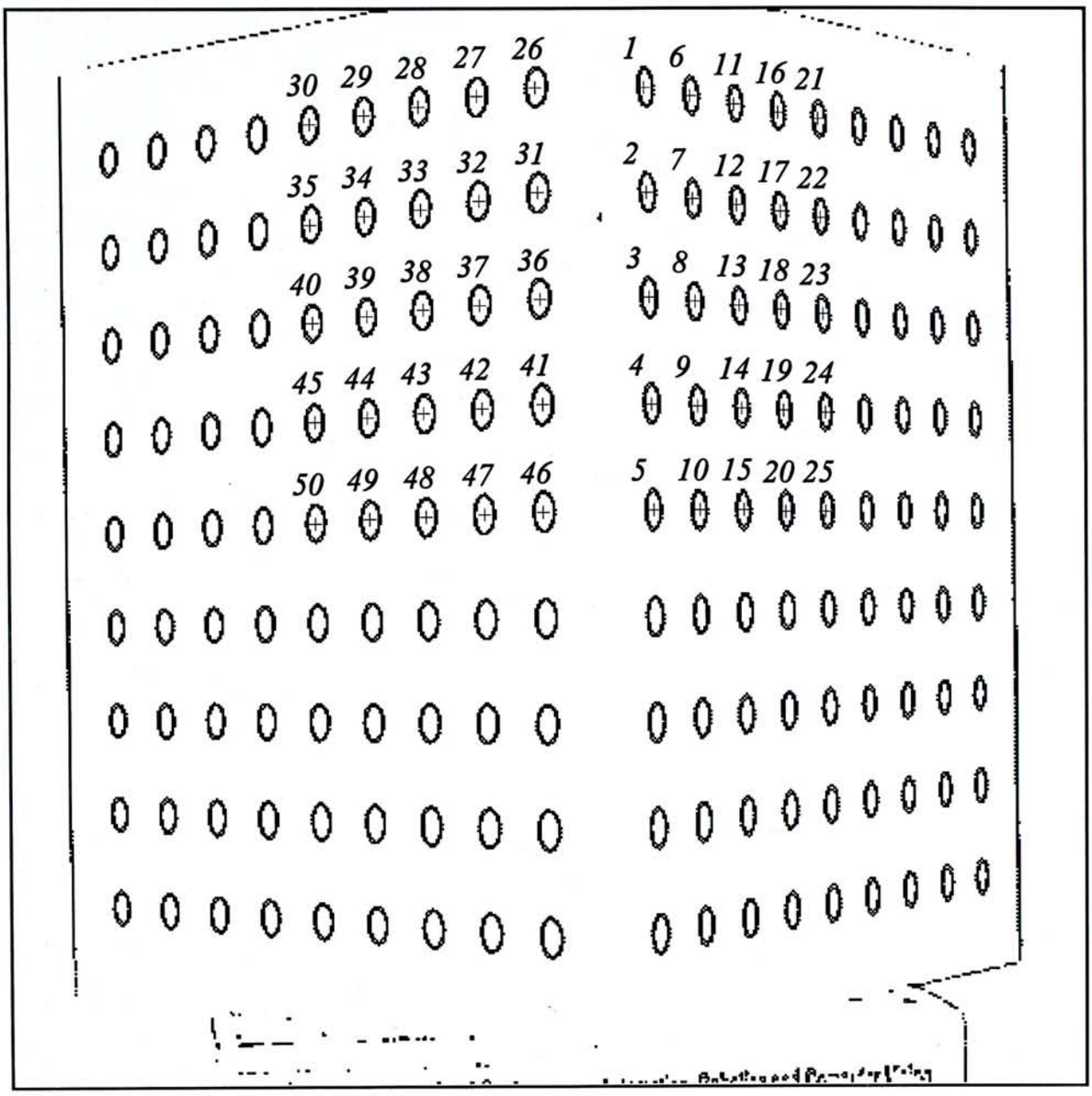


Figure 2.7 Detected Centroid of the white circular disc of image ar_1.img.

	x_w (mm)	y_w (mm)	z_w (mm)	x_i (pixel)	y_i (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.035894e+02	3.644575e+01
2	3.000000e+01	3.000000e+01	0.000000e+00	3.043773e+02	8.668257e+01
3	3.000000e+01	6.000000e+01	0.000000e+00	3.051673e+02	1.370488e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.059582e+02	1.874711e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.067521e+02	2.380875e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.256170e+02	4.052510e+01
7	6.000000e+01	3.000000e+01	0.000000e+00	3.263775e+02	8.981657e+01
8	6.000000e+01	6.000000e+01	0.000000e+00	3.271389e+02	1.391662e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.279020e+02	1.886268e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.286675e+02	2.382415e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.466598e+02	4.442206e+01
12	9.000000e+01	3.000000e+01	0.000000e+00	3.474386e+02	9.281679e+01
13	9.000000e+01	6.000000e+01	0.000000e+00	3.482172e+02	1.411975e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.489984e+02	1.897380e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.497814e+02	2.383899e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.670467e+02	4.819757e+01
17	1.200000e+02	3.000000e+01	0.000000e+00	3.678450e+02	9.572375e+01
18	1.200000e+02	6.000000e+01	0.000000e+00	3.686419e+02	1.431659e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.694423e+02	1.908147e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.702438e+02	2.385337e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.866859e+02	5.183460e+01
22	1.500000e+02	3.000000e+01	0.000000e+00	3.874406e+02	9.851519e+01
23	1.500000e+02	6.000000e+01	0.000000e+00	3.881929e+02	1.450500e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.889494e+02	1.918422e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.897065e+02	2.386705e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.513614e+02	3.636567e+01
27	0.000000e+00	0.000000e+00	6.000000e+01	2.227936e+02	4.099078e+01
28	0.000000e+00	0.000000e+00	9.000000e+01	1.951707e+02	4.546292e+01
29	0.000000e+00	0.000000e+00	1.200000e+02	1.684290e+02	4.979240e+01
30	0.000000e+00	0.000000e+00	1.500000e+02	1.424848e+02	5.399276e+01
31	0.000000e+00	3.000000e+01	3.000000e+01	2.520731e+02	8.677228e+01
32	0.000000e+00	3.000000e+01	6.000000e+01	2.234928e+02	9.056511e+01
33	0.000000e+00	3.000000e+01	9.000000e+01	1.958279e+02	9.423645e+01
34	0.000000e+00	3.000000e+01	1.200000e+02	1.690524e+02	9.778977e+01
35	0.000000e+00	3.000000e+01	1.500000e+02	1.430936e+02	1.012347e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.527876e+02	1.373669e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.241943e+02	1.403065e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.964871e+02	1.431549e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.696775e+02	1.459111e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.437037e+02	1.485813e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.535032e+02	1.880504e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.248973e+02	1.901535e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.971479e+02	1.921935e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.703042e+02	1.941670e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.443157e+02	1.960776e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.542188e+02	2.387244e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.256002e+02	2.399926e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.978085e+02	2.412242e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.709309e+02	2.424153e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.449277e+02	2.435676e+02

Table 2.1 World coordinates of control points and their corresponding image coordinates.

	ar_1	ar_2	ar_3	ar_4	ar_5
Ncx [sel]	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02
Nfx [pix]	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02
dx [mm/sel]	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03
dy [mm/sel]	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03
dpx [mm/pix]	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02
dpy [mm/pix]	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03
Cx [pix]	2.4776446285e+02	2.5045318605e+02	2.3915637605e+02	2.5128969952e+02	2.1917888404e+02
Cy [pix]	2.5469099832e+02	2.5841649935e+02	2.2364229052e+02	2.5808149743e+02	2.6454267555e+02
sx [ ]	1.1227882758e+00	1.1211101243e+00	1.1201509065e+00	1.1196347046e+00	1.1186478530e+00
f [mm]	1.6549503015e+01	1.6186245801e+01	1.6464527975e+01	1.6234382067e+01	1.6156441142e+01
kapp1 [1/mm^2]	8.1928222662e-04	6.3127568400e-04	8.8132895870e-04	2.0878829816e-04	2.4488778435e-05
Tx [mm]	2.6190461917e+01	-6.2233653245e+01	-4.4673168825e+01	6.6544516338e+01	5.8426623406e+01
Ty [mm]	-1.2990231907e+02	-1.3244508357e+02	3.6172454438e+01	-1.9655170712e+01	2.1660664493e+01
Tz [mm]	1.1601219277e+03	1.2448983789e+03	1.2835462809e+03	1.2507127362e+03	1.2495327582e+03
Rx [rad]	-1.7261899417e-02	-1.1574339177e-02	-2.6914326506e-01	-1.3210325652e-01	-1.9594023297e-01
Ry [rad]	-8.8600803250e-01	-1.0609916199e+00	-1.0375421206e+00	-7.4518660207e-01	-8.5583291935e-01
Rz [rad]	-4.3362310382e-03	-8.8180149164e-03	2.1230558538e-01	7.4088293275e-02	1.3148292402e-01
p1	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00
p2	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00

Table 2.2 Output list of Reg Willson's calibration package for the images of ar_1.img ... ar_5.img. Estimated extrinsic and intrinsic parameters in each situation can be found here.

	undistorted image plane error:				object space error			
	mean	stddev	max [pix]	sse [pix^2]	mean	stddev	max [mm]	sse [mm^2]
ar_1	0.16678	0.069218	0.317553	1.625548	0.113307	0.045439	0.206447	0.743093
ar_2	0.112294	0.055498	0.239516	0.781417	0.081477	0.036672	0.160035	0.397823
ar_3	0.105475	0.050694	0.229076	0.682169	0.07665	0.034119	0.148284	0.350805
ar_4	0.098536	0.0552	0.223456	0.634775	0.070044	0.035891	0.151723	0.308431
ar_5	0.11597	0.059682	0.258191	0.846994	0.087102	0.042827	0.168331	0.469214

Table 2.3 Camera calibration error statistics for ar_1.img ... ar_5.img. The sse is the abbreviation of the sum of squared error.

2.6 Conclusion

The resulting error statistics shows that both R. Y. Tsai's two stage RAC based camera calibration method and Reg Willson's implementation performs well in our experimental environment. Thus we have a sound basis for doing head-eye calibration of the active vision system.

References

[1] P. Beardsley, D. Murray, "Camera calibration using vanishing points," *Proc. British Machine Vision Conf.,* September, Leeds, pp. 22-24, 1992

[2] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vision*, 4, pp. 127-140, 1990.

[3] W. Faig, "Calibration of close-range photogrammetry systems: Mathematical formulation," *Photogrammetric Eng. Romote Sensing*, vol. 41, pp. 1479-1486,1975

[4] H. A. Martins, J. R. Birk, and R.B. Kelley, "Camera models based on data from two calibration planes," *Computer Graphics Image Processing*, vol. 17, pp. 173-180, 1981.

[5] R. Y. Tsai, "A versatile camera calibration techinque for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Trans. Robotics Automn.* vol. 3, no. 4, pp. 323-344, August, 1987.

[6] G. Q. Wei, S. D. Ma, "A complete two-plane camera calibration method and experimental comparisons," *Proc. 4th Intern. Conf. Comput. Vision, Berlin, May 11-14, 1993, pp. 133-138.*

[7] G. Q. Wei, S. D. Ma, "Implicit and Explicit Camera Calibration: Theory and Experiments," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol.16, no. 5, May,1994, pp. 469-480

[8] Y. Yakimovsky and R. Cunningham, "A system for extracting three dimensional measurements from a stereo pair of TV cameras," *Computer Graphics Image Processing*, vol. 7, pp. 195-210, 1978.

Chapter 3

Calibration for an Active Vision System : Head-Eye Calibration

3.1 Why Head Eye Calibration

When one starts to deal with a variety of fundamental vision problems utilizing the active vision paradigm, the most important concern is that one is able to control the geometric parameters of the camera including its intrinsic parameters and extrinsic parameters. In a wide range of situations one needs to be able to control the pose of camera in order to get the work done. Thus, it is essential to know the spatial relationship between the camera coordinate frame and the head coordinate frame which is fixed on the last link of the active vision system.

The problem of head-eye calibration is posed as determining the orientation and the position of a camera coordinate frame with respect to the head (the last link of the active vision system) coordinate frame of the observer robot on which the camera is rigidly fixed.

The head eye calibration for an active vision system provides us with a tool to describe the geometrical relationship between the camera coordinate frame and the head coordinate frame (or the hand coordinate frame). Once the head-eye calibration is accomplished, the pose of camera can be controled as being required by some active vision algorithm.

Previous research on this topic has reduced this problem to the solution to a equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$, [3] , where $\boldsymbol{A}$ is a 4×4 homogeneous transformation which denotes the relative motion of robot head coordinate system with respect to the robot world coordinate frame. and $\boldsymbol{B}$ is a 4×4 homogeneous transformation which denotes the corresponding relative motion of camera coordinate frame with respect to a camera world coordinate frame attached on the calibration block.

3.2 Review of the Existing Head-Eye Calibration Algorithms

There have been several approaches being developed to solve the head-eye calibration problem.

Existing head-eye calibration techniques fall into two categories.

3.2.1 Category I Classic Approaches

Shiu and Ahmad[3] pointed out that the head-eye calibration can be done by solving two simultaneous equations of the form $\boldsymbol{A}_1\boldsymbol{X} = \boldsymbol{XB}_1$ and $\boldsymbol{A}_2\boldsymbol{X} = \boldsymbol{XB}_2$, where $\boldsymbol{A}_i$ are the motion described in the coordinate frame of head, $\boldsymbol{B}_i$ represent the induced motion of the camera

described in the camera coordinate frame and $\boldsymbol{X}$ is the homogeneous transform from camera frame to coordinate frame fixed on the robot head. A closed form solution to this system of equations was developed.

The most famous one was proposed by R. Y. Tsai.[1]. In that algorithm, the rotation matrix was estimated at first, then the translation vector is estimated using the rotation matrix just obtained. Tsai using a principal vector to represent the rotation matrix, the principal vector of eye to head rotation was then estimated using least square methods on the basis of geometric concepts. The RAC (Radial Alignment Constraint) based camera calibration method [5] is utilized to calculate the extrinsic parameters of the camera so that radial distortion of lens can be considered in their method.

All of above mentioned techniques require a calibration object on which a camera world coordinate is fixed. These methods are not flexible enough in the situation where the intrinsic and extrinsic parameters of the camera change frequently.

3.2.2 Category II Self-Calibration Techniques

Ma[2] proposed a linear self calibration method which can calibrate both intrinsic and extrinsic parameters(head - eye) using a sequence of specially designed motion. If the camera intrinsic parameters are known, the orientation of the camera relative to the platform can be solved for by 3 non-coplanar pure translations. If the camera intrinsic parameters are unknown, 2 sequences of 3 orthogonal translations are sufficient to determine the camera orientation and its intrinsic parameters. The reported average reconstruction error is about 1mm at a distance of about 1.5m. A head-eye calibration technique was presented in the same paper. Three non-coplanar pure translations of the robot end-effector are needed to determine the relative orientation between the camera coordinate frame and robot head coordinate frame. And then, an additional general motion of the robot is needed to solve the relative position of camera coordinate frame with respect to the robot coordinate frame. Since only a solution of a linear equation system is used as computing the head-eye rotation matrix $\boldsymbol{R}_{cg}$, the orthonormality constraints can not be ensured.

Viéville[6] derived an auto-calibration procedure for visual sensor parameters on a robotic head. This method is based on the tracking of stationary targets, while the robotics performs a fixed axis rotation.

Horaud[7] has developed a head-eye calibration .method. The major advantage which distinguished this method from others is that only the whole perspective matrix is used in the process of estimating the head-eye transformation $\boldsymbol{H}_{cg}$. Since his method skips the problem of factoring the intrinsic and extrinsic parameters of the camera from the perspective matrix, it is expected to be rather stable theoretically. All the magic that made it possible to use only perspective matrix is its extraordinary selection of camera world coordinate frame. In his method, Horaud selected the camera world coordinate frame to be the same coordinate frame as robot head's.

3.3 R. Tsai's Approach for Hand-Eye (Head-Eye) Calibration

3.3.1 Introduction

During the head-eye calibration process, robot head moves to several different positions.The joint variables of the robot are recorded at each observing position. At the same time, a camera ,which is fixed on the robot head rigidly, grabs a picture of a stationary calibration block at each observing position. Given a set of control points expressed in an object world coordinate fame which is attached on the calibration block, corresponding image feature points are extracted and recorded so that camera extrinsic parameters can be determined at each observing position. The head-eye calibration for a robot can be made with the knowledge of robot motion and camera extrinsic parameters. It takes at least more than 3 observing positions to uniquely determine the transformation from the camera coordinate system to the robot head coordinate system.

3.3.2 Definitions of Coordinate Frames and Homogenous Transformation Matrices:

The various coordinate frames involving in the head-eye calibration process are listed as follows:

Definition 3.1 The head coordinate frame, denoted by G_i, is a right-handed 3D Cartesian coordinate frame which is attached rigidly on the last link of a manipulator of the robot, where the subscript i means the ith observing location.

Definition 3.2 The camera coordinate frame, denoted by C_i, is a right-handed 3D Cartesian coordinate frame which is fixed on a camera, where the subscript i means the ith observing location. The coordinate frame has its origin coincident at the focal point of the camera, the z axis lies along the optical axis of the camera.

Definition 3.3 The calibration block world coordinate frame,denoted by CW, is a right-handed 3D Cartesian coordinate frame which is established on a calibration block. The 3D coordinates of set of control points with respect to CW are measured accurately in order to perform camera calibration.

Definition 3.4 The robot world coordinate frame, denoted by RW, is a right-handed 3D Cartesian coordinate frame which is set on supporting base of the robot. Given a set of robot joint variables, the relative motion of the robot head (the last link of a manipulator of the robot) can be determined uniquely with respect to this coordinate frame.

Definition 3.5 The homogeneous coordinate transformation from a right-handed 3D Cartesian coordinate frame O_1-X_1-Y_1-Z_1 to another right-handed 3D Cartesian coordinate frame O_2-X_2-Y_2-Z_2, denoted by $\boldsymbol{H}$, is defined as a 4×4 homogeneous rotation-translation transformation matrix.

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \tag{3.1}$$

where, $\boldsymbol{R}$ is a 3×3 rotation matrix and $\boldsymbol{T}$ is a 3×1 translation vector. $\boldsymbol{R}$ is an orthonormal matrix . Hence , $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$, where $\boldsymbol{I}$ is a 3×3 identity matrix.

A point in the space, denoted by $\boldsymbol{p}$, can be represented by its homogeneous coordinates with respect to the O_1-X_1-Y_1-Z_1 and O_2-X_2-Y_2-Z_2 coordinate systems, respectively,as

$$\boldsymbol{p}_{O_1\text{-}X_1\text{-}Y_1\text{-}Z_1} = (p_{x_1}, p_{y_1}, p_{z_1}, 1)^T \quad and \quad \boldsymbol{p}_{O_2\text{-}X_2\text{-}Y_2\text{-}Z_2} = (p_{x_2}, p_{y_2}, p_{z_2}, 1)^T \tag{3.2}$$

where $\boldsymbol{p}_{O_1\text{-}X_1\text{-}Y_1\text{-}Z_1}$ and $\boldsymbol{p}_{O_2\text{-}X_2\text{-}Y_2\text{-}Z_2}$ represent the same point $\boldsymbol{p}$ in the space with reference to different coordinate frames. Then, $\boldsymbol{H}$ will transforms the homogeneous coordinates of $\boldsymbol{p}_{O_1\text{-}X_1\text{-}Y_1\text{-}Z_1}$ to the homogeneous coordinates $\boldsymbol{p}_{O_2\text{-}X_2\text{-}Y_2\text{-}Z_2}$ which is expressed with respect to the O_2-X_2-Y_2-Z_2 coordinate frame, That is,

$$\boldsymbol{p}_{O_2\text{-}X_2\text{-}Y_2\text{-}Z_2} = \boldsymbol{H}\boldsymbol{p}_{O_1\text{-}X_1\text{-}Y_1\text{-}Z_1} \tag{3.3}$$

Definition 3.6 The coordinate transformation from G_i to RW, denoted by $\boldsymbol{H}_{gi}$, is defined as a 4×4 homogeneous rotation-translation transformation matrix.

$$\boldsymbol{H}_{gi} = \begin{bmatrix} \boldsymbol{R}_{gi} & \boldsymbol{T}_{gi} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \tag{3.4}$$

where i specifies the ith observing station.

Definition 3.7 The coordinate transformation from CW to C_i, denoted by $\boldsymbol{H}_{ci}$, is defined as a 4×4 homogeneous rotation-translation transformation matrix.

$$\boldsymbol{H}_{ci} = \begin{bmatrix} \boldsymbol{R}_{ci} & \boldsymbol{T}_{ci} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \tag{3.5}$$

where i specifies the ith observing station .

Definition 3.8 The coordinate transformation from G_i to G_j, denoted by $\boldsymbol{H}_{gij}$, is defined as a 4×4 homogeneous rotation-translation transformation matrix.

$$\boldsymbol{H}_{gij} = \begin{bmatrix} \boldsymbol{R}_{gij} & \boldsymbol{T}_{gij} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \tag{3.6}$$

where i , j specifies the ith,and jth observing station respectively.

Definition 3.9 The coordinate transformation from C_i to C_j, denoted by $\boldsymbol{H}_{cij}$, is defined as a 4×4 homogeneous rotation–translation transformation matrix.

$$\boldsymbol{H}_{cij} = \begin{bmatrix} \boldsymbol{R}_{cij} & \boldsymbol{T}_{cij} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{3.7}$$

where i , j specifies the ith,and jth observing station respectively.

Definition 3.10 The coordinate transformation from C_i to G_i, denoted by $\boldsymbol{H}_{cg}$, is defined as a

$$\boldsymbol{H}_{cg} = \begin{bmatrix} \boldsymbol{R}_{cg} & \boldsymbol{T}_{cg} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{3.8}$$

4×4 homogeneous rotation–translation transformation matrix.

3.3.3 Formulation of the Head - Eye Calibration Problem

The relationships between the homogeneous matrices and the various coordinate frames are shown in the Fig. 3.1.

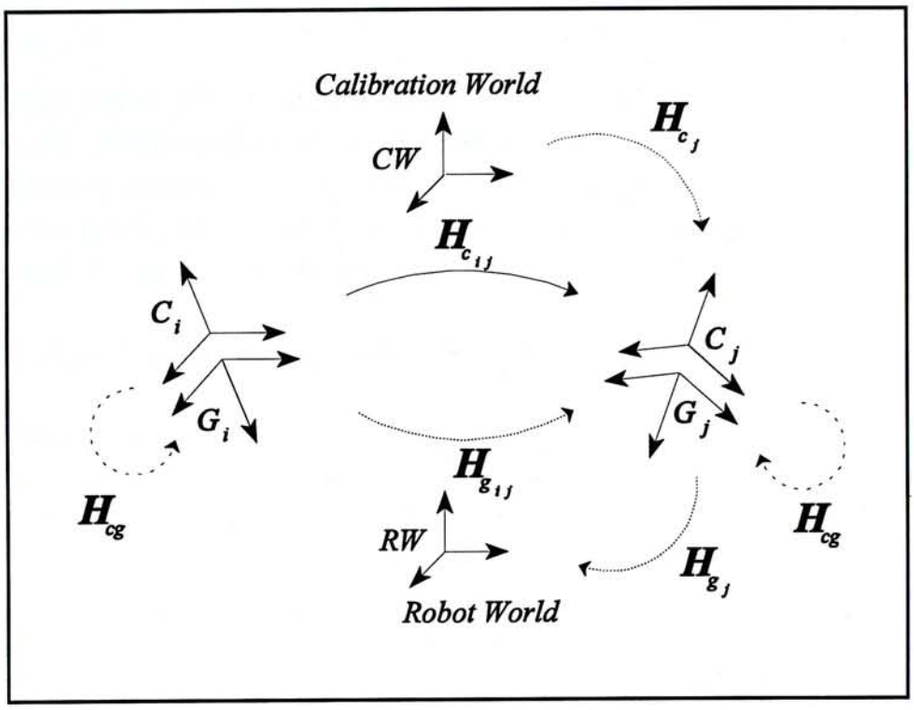


Figure 3.1 Relationship between the homogeneous matrices and the coordinate frames

When a robot head moves from ith observing position to jth observing position, we consider the coordinate transformation from the camera coordinate frame at the ith observing position, denoted by C_i, to G_j which is the robot head coordinate frame at the jth observing position. This coordinate transformation can be decomposed as two subtransformations, a coordinate transformation from the coordinate frame C_i to the coordinate frame G_i followed by a coordinate transformation from the coordinate frame G_i to the coordinate frame G_j. Thus, this coordinate transformation can be represented by $\boldsymbol{H}_{gij}\boldsymbol{H}_{cg}$. While this transformation can be decomposed in another way: a coordinate transformation from the coordinate frame C_i to the coordinate frame C_j followed by a coordinate transformation from the coordinate frame C_j to the coordinate frame G_j, that is $\boldsymbol{H}_{cg}\boldsymbol{H}_{cij}$. Since $\boldsymbol{H}_{gij}\boldsymbol{H}_{cg}$ and $\boldsymbol{H}_{cg}\boldsymbol{H}_{cij}$ are two different decompositions of a same coordinate transformation, $\boldsymbol{H}_{gij}\boldsymbol{H}_{cg}$ and $\boldsymbol{H}_{cg}\boldsymbol{H}_{cij}$ should be identical.

The head-eye calibration problem can then be formulated as: given a set of $\boldsymbol{H}_{gij}$s and the corresponding set of $\boldsymbol{H}_{cij}$ s, find $\boldsymbol{H}_{cg}$ using the following equation.

$$\boldsymbol{H}_{gij}\boldsymbol{H}_{cg} = \boldsymbol{H}_{cg}\boldsymbol{H}_{cij} \tag{3.9}$$

where $\boldsymbol{H}_{gij}$ and $\boldsymbol{H}_{cij}$ are calculated using the following equations,

$$\boldsymbol{H}_{gij} = \boldsymbol{H}_{gj}^{-1}\boldsymbol{H}_{gi} \tag{3.10}$$

$$\boldsymbol{H}_{cij} = \boldsymbol{H}_{cj}\boldsymbol{H}_{ci}^{-1} \tag{3.11}$$

In the above two equations , (3.10) and (3.11) , $\boldsymbol{H}_{ci}$, which is the the coordinate transformation from the camera calibration world coordinate frame CW to C_i, can be obtained from camera calibration (extrinsic parameters) , $\boldsymbol{H}_{gi}$ can be calculated usng kinematic equations of the manipulator when the robot joint variables are given. It describes the position and orientation of robot head coordinate frame relative to the robot world coordinate frame.

3.3.4 Using Principal Vector to Represent Rotation Transformation Matrix

Consider a rotation matrix $\boldsymbol{R}$,

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.12}$$

its characteristic equation is given by,

$$|\boldsymbol{R} - \lambda \boldsymbol{I}| = 0 \tag{3.13}$$

where, $\boldsymbol{I}$ is a 3×3 identity matrix.

Since $\boldsymbol{R}$ is a orthonormal matrix , its three column vectors, denoted by $\boldsymbol{r}_1 = (r_{11}, r_{21}, r_{31})^T$, $\boldsymbol{r}_2 = (r_{12}, r_{22}, r_{32})^T$, $\boldsymbol{r}_3 = (r_{13}, r_{23}, r_{33})^T$ respectively, satisfy the following orthonormal constraints:

$$\begin{aligned} &\boldsymbol{r}_i \cdot \boldsymbol{r}_j = 0 \\ &|\boldsymbol{r}_i|^2 = 1 \qquad \text{where} \qquad i, j = 1 \ldots 3 \\ &\boldsymbol{r}_1 \times \boldsymbol{r}_2 = \boldsymbol{r}_3 \qquad \boldsymbol{r}_2 \times \boldsymbol{r}_3 = \boldsymbol{r}_1 \qquad \boldsymbol{r}_3 \times \boldsymbol{r}_1 = \boldsymbol{r}_2 \end{aligned} \tag{3.14}$$

Substituting (3.12) and (3.14) into (3.13), the characteristic equation of $\boldsymbol{R}$ can be rewritten as

$$-\lambda^3 + (r_{11}+r_{22}+r_{33})\lambda^2 - (r_{11}+r_{22}+r_{33})\lambda + 1 = 0 \tag{3.15}$$

(3.15) is a polynomial of degree 3 in λ with real coefficients.Dividing both sides of (3.15) by (λ - 1), gives,

$$(\lambda-1)\left[-\lambda^2 + (K-1)\lambda - (K-1)\right] = 0$$

$$\text{where} \qquad K = r_{11}+r_{22}+r_{33} \tag{3.16}$$

Thus, (3.16) has a real root which is 1. In other words, one eigenvalues of $\boldsymbol{R}$ must be 1. Recalling the definitions of eigenvalues and eigenvectors, there must exist a real vector $\boldsymbol{P}_r$ for a given rotation matrix so that

$$\boldsymbol{R}\boldsymbol{P}_r = \boldsymbol{P}_r \tag{3.17}$$

(3.17) means that , for any given rotation matrx $\boldsymbol{R}$, there must be a real vector $\boldsymbol{P}_r$ which defines a 3D line pointing away from the origin so that all the points on this line remain invariant after the rotation specified by $\boldsymbol{R}$. This 3D line is called the equivalent axis of $\boldsymbol{R}$ .It has been shown that any given rotation matrix $\boldsymbol{R}$ can be modeled as a rotation by an angle θ about an equivalent axis through the origin with an unit vector being $[n_1, n_2, n_3]^T$. The representation of $\boldsymbol{R}$ in terms of θ and $[n_1, n_2, n_3]^T$ is given by,

$$\boldsymbol{R} = \begin{bmatrix} n_1^2+(1-n_1^2)\cos\theta & n_1n_2(1-\cos\theta)-n_3\sin\theta & n_1n_3(1-\cos\theta)+n_2\sin\theta \\ n_1n_2(1-\cos\theta)+n_3\sin\theta & n_2^2+(1-n_2^2)\cos\theta & n_2n_3(1-\cos\theta)-n_1\sin\theta \\ n_1n_3(1-\cos\theta)-n_2\sin\theta & n_2n_3(1-\cos\theta)+n_1\sin\theta & n_3^2+(1-n_3^2)\cos\theta \end{bmatrix} \tag{3.18}$$

Hence , $\boldsymbol{R}$ around axis defined by the unit vector $[n_1, n_2, n_3]^T$ can be completely specified by specifying θ and eigenvector(or principal vector) of $\boldsymbol{R}$ with its corresponding eigenvalue being 1. Modulating the magnitude of $[n_1, n_2, n_3]^T$ by a function of θ, gives the principal axis of $\boldsymbol{R}$,

$$\boldsymbol{P}_r = 2\sin\frac{\theta}{2}[n_1, n_2, n_3] \qquad \text{where} \quad 0 \le \theta \le \pi \tag{3.19}$$

It can be shown that $\boldsymbol{R}$ is a simple function of the principal axis $\boldsymbol{P}_r$.

$$\boldsymbol{R} = \left(1 - \frac{|\boldsymbol{P}_r|^2}{2}\right)\boldsymbol{I} + \frac{1}{2}(\boldsymbol{P}_r\boldsymbol{P}_r^T + \alpha \cdot \mathrm{Skew}(P_r)) \tag{3.20}$$

$$\text{where} \qquad \alpha = \sqrt{4 - |\boldsymbol{P}_r|^2}$$

In (3.20), Skew ($\boldsymbol{P}_r$) is defined by

$$\mathrm{Skew}(\boldsymbol{P}_r) = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \tag{3.21}$$

Therefore, we can determine $\boldsymbol{R}$ readily once we can obtain $\boldsymbol{P}_r$.

3.3.5 Calculating $\boldsymbol{R}_{cg}$ and $\boldsymbol{T}_{cg}$

By representing a set of rotation matrices which are involved in (3.9) with their principal axes respectively, the principal axis for the rotation matrix from the camera coordinate frame to the robot head coordinate frame is estimated at first by using a simple geomtric relationship between those principal axes. Then, the rotation matrix $\boldsymbol{R}_{cg}$ is calculated from its principal axis. Finally, the translation vector $\boldsymbol{T}_{cg}$ is obtained on the basis of previous aquired $\boldsymbol{R}_{cg}$.

Some definitions of notation are lised as follows :

θ_R: Angle of rotation for $\boldsymbol{R}$.

$\boldsymbol{P}_{gij}$: Principal axis or rotation axis for $\boldsymbol{R}_{gij}$, which is the 3D rotation from coordinate frame G_i to G_j.

$\boldsymbol{P}_{cij}$: Rotation axis for $\boldsymbol{R}_{cij}$.

$\boldsymbol{P}_{cg}$: Rotation axis for $\boldsymbol{R}_{cg}$.

$$\boldsymbol{P}'_{cg} = \frac{1}{2\cos\left(\frac{\theta_{R_{cg}}}{2}\right)}\boldsymbol{P}_{cg} = \frac{1}{\sqrt{4 - |\boldsymbol{P}_{cg}|^2}}\boldsymbol{P}_{cg} \tag{3.22}$$

A skew-symmetric matrix generated by a 3D vector $\boldsymbol{V}$ is given by:

$$\text{Skew}(\boldsymbol{V}) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{3.23}$$

It is worthwhile to note that since $\boldsymbol{P}_{gij}$, $\boldsymbol{P}_{cij}$, and $\boldsymbol{P}_{cg}$ are rotation axes with a function of angle as their length, they completely represent $\boldsymbol{R}_{gij}$, $\boldsymbol{R}_{cij}$, and $\boldsymbol{R}_{cg}$.

By rearranging (3.9), gives

$$\boldsymbol{H}_{gij} = \boldsymbol{H}_{cg}\boldsymbol{H}_{cij}\boldsymbol{H}_{cg}^{-1} \tag{3.24}$$

According to (3.8) , the inverse of $\boldsymbol{H}_{cg}$ is given by

$$\begin{bmatrix} \boldsymbol{R}_{cg} & \boldsymbol{T}_{cg} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{R}_{cg}^T & -\boldsymbol{R}_{cg}^T\boldsymbol{T}_{cg} \\ 0 & 1 \end{bmatrix} \tag{3.25}$$

Expanding (3.24) gives:

$$\begin{aligned} \boldsymbol{H}_{gij} &= \begin{bmatrix} \boldsymbol{R}_{gij} & \boldsymbol{T}_{gij} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_{cg}\boldsymbol{R}_{cij}\boldsymbol{R}_{cg}^T & (\boldsymbol{I}-\boldsymbol{R}_{cg}\boldsymbol{R}_{cij}\boldsymbol{R}_{cg}^T)\boldsymbol{T}_{cg}+\boldsymbol{R}_{cg}\boldsymbol{T}_{cij} \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.26}$$

Compare the cooresponding entries of (3.26) yields:

$$\boldsymbol{R}_{gij} = \boldsymbol{R}_{cg}\boldsymbol{R}_{cij}\boldsymbol{R}_{cg}^T \tag{3.27}$$

By (3.17), representing $\boldsymbol{R}_{gij}$ with its principal axis $\boldsymbol{P}_{gij}$ gives

$$\left(\boldsymbol{R}_{cg}\boldsymbol{R}_{cij}\boldsymbol{R}_{cg}^T\right)\boldsymbol{P}_{gij} = \boldsymbol{P}_{gij} \tag{3.28}$$

Rearranging (3.28), we have,

$$\boldsymbol{R}_{cij}\left(\boldsymbol{R}_{cg}^T\boldsymbol{P}_{gij}\right) = \boldsymbol{R}_{cg}^T\boldsymbol{P}_{gij} \tag{3.29}$$

Thus, $\boldsymbol{R}_{cg}^{T}\boldsymbol{P}_{gij}$ is a real eigenvector of $\boldsymbol{R}_{cij}$ It had been shown by Y. C. Shiu and S. Ahmad in their Lemma 4 [3] that $\boldsymbol{R}_{cij}$ and $\boldsymbol{R}_{gij}$ must have the same angle of rotation, therefore, we obtain,

$$\boldsymbol{P}_{gij} = \boldsymbol{R}_{cg}\boldsymbol{P}_{cij} \tag{3.30}$$

The geometrical relationship of the above equation is depicted in Fig. 3.2. It can be shown that

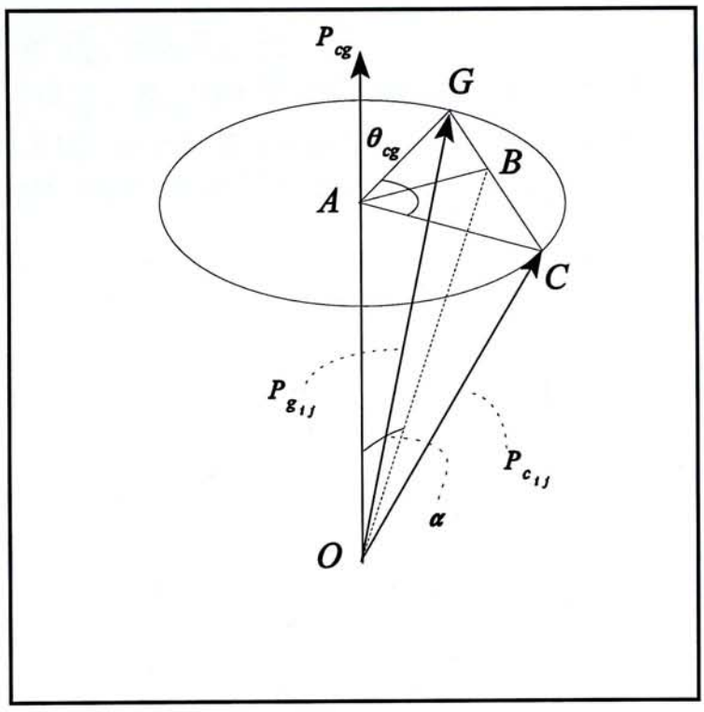


Figure 3.2 Geometrical relationship between P_{cg}, P_{cij}, P_{gij}. P_{cg} rotate P_{cij} into P_{gij}

$\boldsymbol{P}_{cg}$ is perpendicular to the vector $\boldsymbol{P}_{cij} - \boldsymbol{P}_{gij}$, and $\boldsymbol{P}_{cij} - \boldsymbol{P}_{gij}$ is perpendicular to $\boldsymbol{P}_{cij} + \boldsymbol{P}_{gij}$. After some scaling manipulation in (3.22) gives the following formula:

$$(\boldsymbol{P}_{gij} + \boldsymbol{P}_{cij}) \times \boldsymbol{P}_{cg}' = \boldsymbol{P}_{cij} - \boldsymbol{P}_{gij} \tag{3.31}$$

The outer product of any two 3D vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ can be represented in terms of matrix multiplication as follows:

$$\boldsymbol{a} \times \boldsymbol{b} = \text{Skew}(\boldsymbol{a})\boldsymbol{b} \tag{3.32}$$

Combine (3.31) and (3.32) gives (3.33).

$$\text{Skew}(\boldsymbol{P}_{gij} + \boldsymbol{P}_{cij})\boldsymbol{P}_{cg}' = \boldsymbol{P}_{cij} - \boldsymbol{P}_{gij} \tag{3.33}$$

$\boldsymbol{R}_{cg}$ and $\boldsymbol{T}_{cg}$ will be estimated in the following steps:

Step 1 Compute $\boldsymbol{P}_{gij}$ and $\boldsymbol{P}_{cij}$

By (3.10) and (3.11), $\boldsymbol{R}_{gij}$, $\boldsymbol{R}_{cij}$ can be calculated from $\boldsymbol{H}_{ci}$, $\boldsymbol{H}_{gi}$,where $\boldsymbol{H}_{ci}$ is obtained by camera calibration and $\boldsymbol{H}_{gi}$ is calcaulated using recorded robot joint variables. Then , the real eigenvector and rotation angle are calculated for $\boldsymbol{R}_{gij}$, $\boldsymbol{R}_{cij}$, respectively.

Let
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.34}$$

thus, the rotation angle is,

$$\theta = \arctan\frac{\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{12} - r_{21})^2}}{(r_{11} + r_{22} + r_{33} - 1)}$$

Finally, $\boldsymbol{P}_{gij}$ and $\boldsymbol{P}_{cij}$ are calculated from $\boldsymbol{R}_{gij}$, $\boldsymbol{R}_{cij}$ using (3.19), respectively.

Step 2 Compute $\boldsymbol{P}_{cg}'$

By using (3.33), set up a system of linear equations with $\boldsymbol{P}_{cg}'$ as the unkown. Since Skew($\boldsymbol{P}_{gij}$+$\boldsymbol{P}_{cij}$) is always singular, it takes at least two pairs of observing positions to solve for a unique solution for $\boldsymbol{P}_{cg}'$ using linear least squares technique.

Step 3 Compute $\boldsymbol{P}_{cg}$

By (3.22), $\boldsymbol{P}_{cg}$ can be calculated from $\boldsymbol{P}_{cg}'$ as follows:

$$\boldsymbol{P}_{cg} = \frac{2\boldsymbol{P}_{cg}'}{\sqrt{1+|\boldsymbol{P}_{cg}'|^2}} \tag{3.35}$$

Step 4 Computing $\boldsymbol{R}_{cg}$

By (3.20) , $\boldsymbol{R}_{cg}$ can be obtained from $\boldsymbol{P}_{cg}$

Step 5 Computing $\boldsymbol{T}_{cg}$

Given at least two pairs of observing positions i , j, set up a linear systems of three linear equations with $\boldsymbol{T}_{cg}$ as unknowns.

$$(\boldsymbol{R}_{gij}-\boldsymbol{I})\boldsymbol{T}_{cg} = \boldsymbol{R}_{cg}\boldsymbol{T}_{cij}-\boldsymbol{T}_{gij} \tag{3.36}$$

The above equation is derived by comparing the entries in (3.26). For at least two pairs of

observing positions, two sets of (3.36) are established and can be solved for the common unknown T_{cg} using linear least squares solutions.

3.4 Our local implementation of Tsai's Head Eye Calibration Algorithm

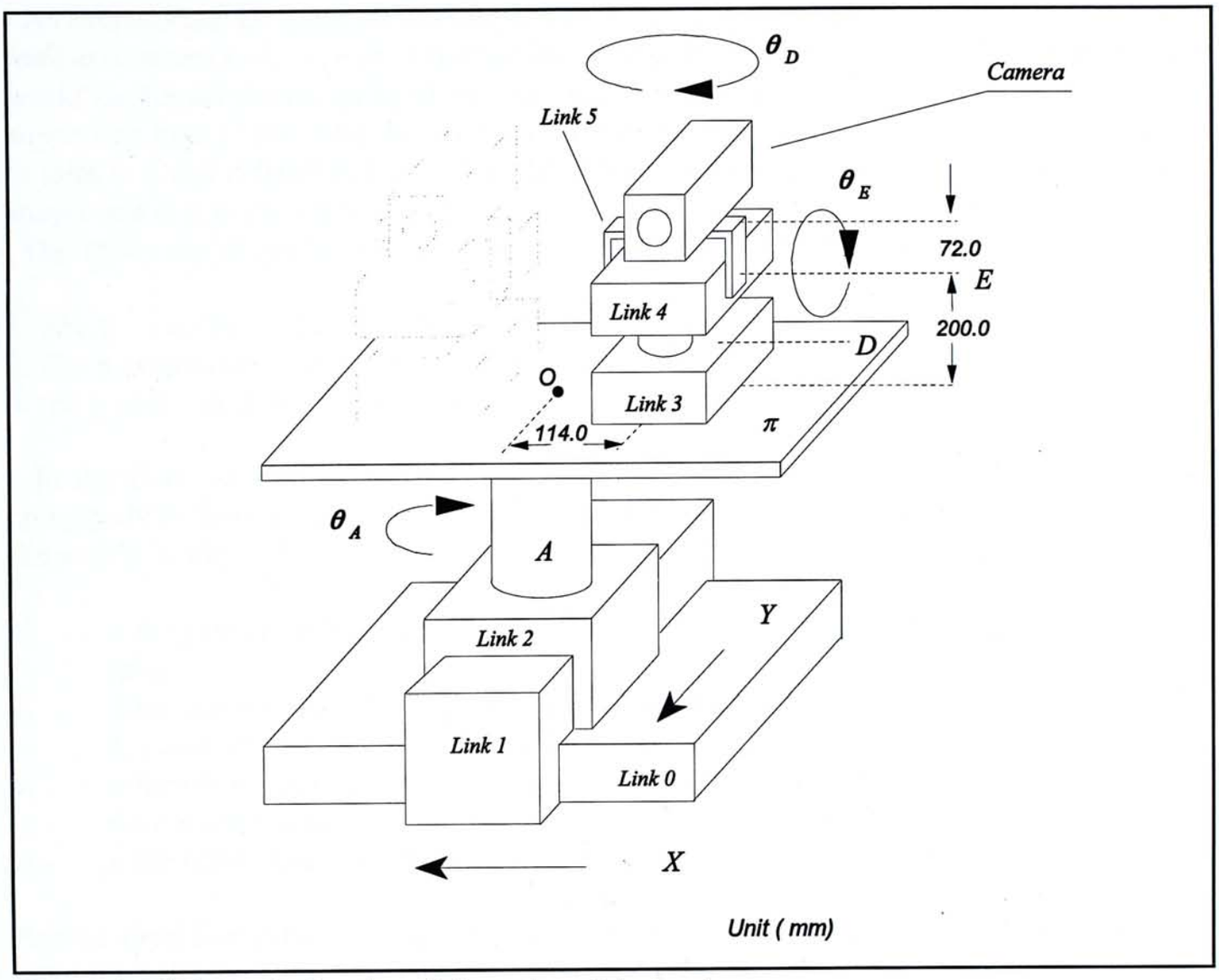


Figure 3.3 Schematic drawing of the active vision system

The schematic drawing of the active vision system is shown in Fig.3.3

Our active vision system is a 7 degree-of -freedom observer robot. This novel active vision system has two linear axes which can provide translational motion within a horizontal plane. It has one common rotary axis around which stereo cameras can pan from one side to the other, one independent vergence axis and one independent tilt axis for each camera. Apart from its flexibility, the active vision system has very high accuracy and repeatability.

Two CCD cameras are fixed rigidly on the last link of two manipulators, respectively. We consider only one camera in our project. The camera is mounted on the last link of the manipulator which consists of two prismatic joints X and Y, the common rotary joint A, one vergence joint D and one tilt joint E. Each joint provides one degree of freedom.

To describe the translational and rotational relationships between adjacent links, Denavit and

Hartengerg [8] proposed a method of establishing a body-attached coordinate frame to each link of a manipulator. A 4×4 homogeneous transformation matrix is used to representing each link's coordinate system at the joint with respect to the previous link's coordinate system. Thus, coordinate transformation from the last link coordinate frame to the robot world coordinate frame can obtained by sequential matrix multiplications.

An orthonormal cartesian coordinate frame ($\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{z}_i$) can be established for each link at its joint axis, where $i = 1, \ldots, n$ (n is the number of degrees of freedom of the manipulator). A robot world coordinate frame, denoted by ($\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$), is established on the link 0 which is the supporting base of the robot. Each ($\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{z}_i$) coordinate frame of a manipulator corresponds to joint $i + 1$ and is fixed in link i. Thus, for this five-axis manipulator, we have six coordinate frame, that is ,($\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$) , ($\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{z}_1$) , ... , ($\mathbf{x}_5$, $\mathbf{y}_5$, $\mathbf{z}_5$).

The following three rules should be obeyed when establishing the coordinate system for a link:

1. The $\mathbf{z}_{i-1}$ axis lies along the axis of motion of the ith joint.
2. The $\mathbf{x}_i$ is normal to the $\mathbf{z}_{i-1}$ axis, and pointing away from it.
3. the $\mathbf{y}_i$ axis completes the right-handed coordinate system as required.

In the D-H representation , the kinematic configuration of a rigid link can be described completely by four geometric parameters associated with each link. These four parameters are illustrated in. Fig. 3.4.

θ_i is the joint angle from the $\mathbf{x}_{i-1}$ axis to the $\mathbf{x}_i$ axis about the $\mathbf{z}_{i-1}$ axis (using the right-hand rule).

d_i is the distance from the origin of the (i - 1)th coordinate frame to the intersection of the $\mathbf{z}_{i-1}$ axis with the $\mathbf{x}_i$ axis along the $\mathbf{z}_{i-1}$ axis.

a_i is the offset distance from the intersection of the $\mathbf{z}_{i-1}$ axis with the $\mathbf{x}_i$ axis to the origin of the ith frame along the $\mathbf{x}_i$ axis (or the shortest distance between the $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ axes.

α_i is the offset angle from the $\mathbf{z}_{i-1}$ to the $\mathbf{z}_i$ axis about the $\mathbf{x}_i$ axis (using the right-hand rule).

Among these four parameters, d_i and θ_i is called the *distance* and the *angle* between the adjacent links, respectively. They determine the relative position of neighbouring links. a_i and α_i is called the *length* and the *joint angle* of the link i respectively. They determine the structure of the link i.

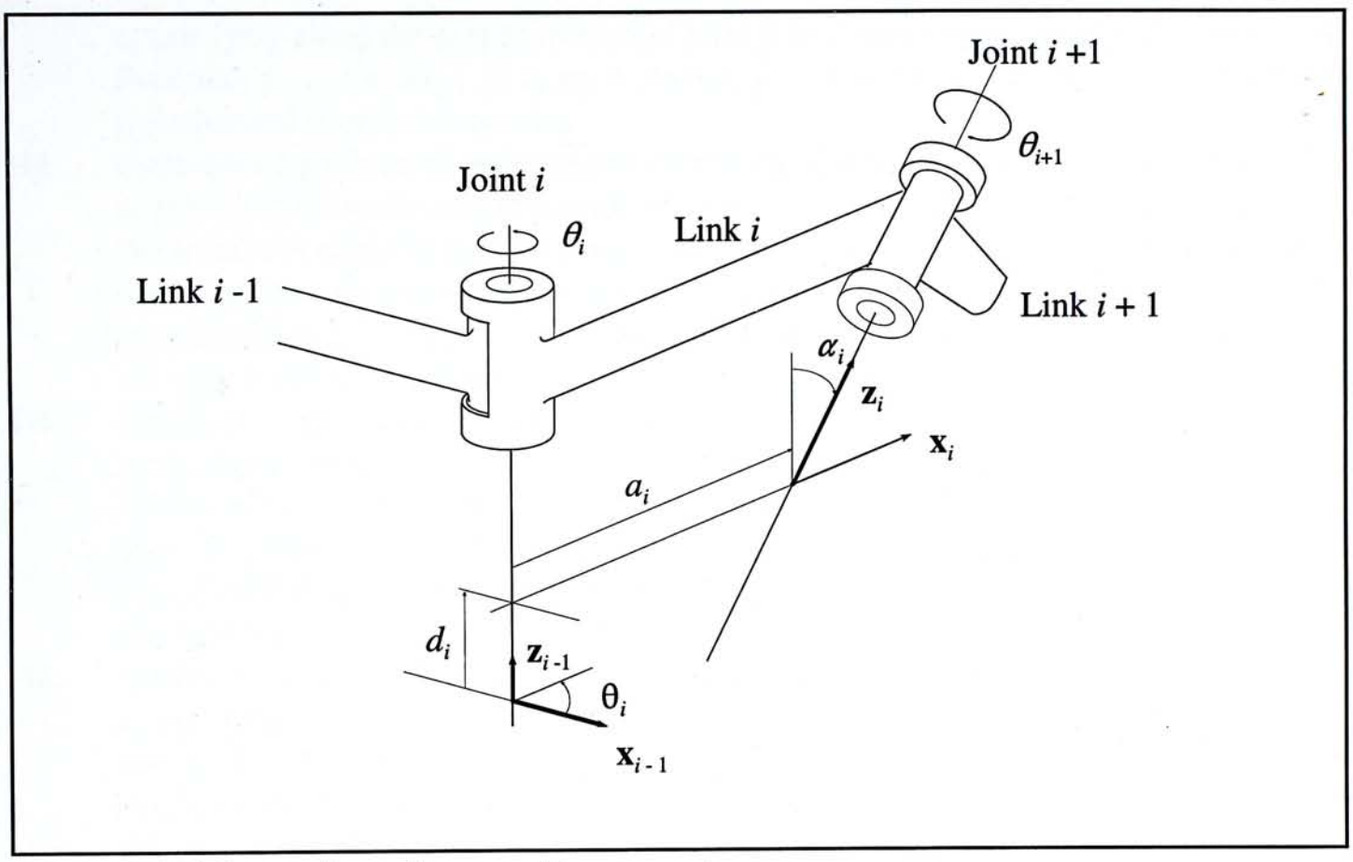


Figure 3.4 Link coordinate frame and its parameters.

3.4.1 Using Denavit - Hartternberg's Approach [8] to Establish a Body-Attached Coordinate Frame for Each Link of the Manipulator

In our research project, we are given a manipulator of 5 degrees of freedom, an orthonormal coordinate system is assigned to each link of manipulator according to Denavit-Hartenberg's approach. The labeling of the coordinate systems begins from the supporting base to the last link of the manipulator. The relations between adjacent links can be represented by a 4×4 homogeneous transformation matrix. In the following steps, six coordinate frames are estalished. The link parameters and joint variables of each link can be determined as well.

D1. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_0,\mathbf{y}_0,\mathbf{z}_0)$ at the supporting base with the $\mathbf{z}_0$ axis lying along the axis of motion of joint X and pointing toward the opposite direction of X. The $\mathbf{x}_0$ and $\mathbf{y}_0$ axes can be conveniently established and are normal to the $\mathbf{z}_0$ axis as shown. For ease of discussion, we locate the origin of the 0th coordinate system at the point O which is the intersection of the A axis with the plane π when X and Y axes are at zero position. The origins of coordinate frames $(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_1)$ and $(\mathbf{x}_2,\mathbf{y}_2,\mathbf{z}_2)$ are chosen to be at the point O as well. (see Fig 3.5). In Fig. 3.5, we use dash lines to show the coincidence of the orgins of the coordinate frames $(\mathbf{x}_0,\mathbf{y}_0,\mathbf{z}_0)$, $(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_1)$ and $(\mathbf{x}_2,\mathbf{y}_2,\mathbf{z}_2)$ when X and Y axes are at zero position.

D2. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_1)$ at the link 1 with the $\mathbf{z}_1$ axis lying along the axis of motion of joint Y and pointing toward the direction of Y. Establish $\mathbf{x}_1 = -(\mathbf{z}_0 \times \mathbf{z}_1) / \| \mathbf{z}_0 \times \mathbf{z}_1 \|$. Assign $\mathbf{y}_1 = +(\mathbf{z}_1 \times \mathbf{x}_1) / \| \mathbf{z}_1 \times \mathbf{x}_1 \|$ to complete the right-handed coordinate system.

D3. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_2,\mathbf{y}_2,\mathbf{z}_2)$ at the link 2 with the

$\mathbf{z}_2$ axis lying along the axis of motion of joint ***A*** and pointing toward the direction of ***A***. Establish $\mathbf{x}_2 = -(\mathbf{z}_1 \times \mathbf{z}_2) / \| \mathbf{z}_1 \times \mathbf{z}_2 \|$. Assign $\mathbf{y}_2 = +(\mathbf{z}_2 \times \mathbf{x}_2) / \| \mathbf{z}_2 \times \mathbf{x}_2 \|$ to complete the right-handed coordinate system.

D4. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_3,\mathbf{y}_3,\mathbf{z}_3)$ at the link 3 with the $\mathbf{z}_3$ axis lying along the axis of motion of joint ***D*** and pointing toward the direction of ***D***. We locate the origin of the coordinate frame $(\mathbf{x}_3,\mathbf{y}_3,\mathbf{z}_3)$ at the intersection of the $\mathbf{z}_3$ and $\mathbf{z}_2$ axes. The distance separating the origin of $(\mathbf{x}_3,\mathbf{y}_3,\mathbf{z}_3)$ and the origin of $(\mathbf{x}_2,\mathbf{y}_2,\mathbf{z}_2)$ is 114.0 mm. Establish $\mathbf{x}_3 = -(\mathbf{z}_2 \times \mathbf{z}_3) / \| \mathbf{z}_2 \times \mathbf{z}_3 \|$. Assign $\mathbf{y}_3 = +(\mathbf{z}_3 \times \mathbf{x}_3) / \| \mathbf{z}_3 \times \mathbf{x}_3 \|$ to complete the right-handed coordinate system.

D5. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_4,\mathbf{y}_4,\mathbf{z}_4)$ at the link 4 with the $\mathbf{z}_4$ axis lying along the axis of motion of joint ***E*** and pointing toward the direction of ***E***. We locate the origin of the coordinate frame $(\mathbf{x}_4,\mathbf{y}_4,\mathbf{z}_4)$ at the intersection of the $\mathbf{z}_4$ and $\mathbf{z}_3$ axes. The distance separating the origin of $(\mathbf{x}_4,\mathbf{y}_4,\mathbf{z}_4)$ and the origin of $(\mathbf{x}_3,\mathbf{y}_3,\mathbf{z}_3)$ is 200.0 mm. Establish $\mathbf{x}_4 = -(\mathbf{z}_3 \times \mathbf{z}_4) / \| \mathbf{z}_3 \times \mathbf{z}_4 \|$. Assign $\mathbf{y}_4 = +(\mathbf{z}_4 \times \mathbf{x}_4) / \| \mathbf{z}_4 \times \mathbf{x}_4 \|$ to complete the right-handed coordinate system.

D6. Establish a right-handed orthonormal coordinate system $(\mathbf{x}_5,\mathbf{y}_5,\mathbf{z}_5)$ at the link 5 with the $\mathbf{z}_5$ axis lying along the direction of $\mathbf{z}_4$ axis. Establish $\mathbf{x}_5$ such that it is normal to both $\mathbf{z}_4$ and $\mathbf{z}_5$ axes. Assign $\mathbf{y}_5$ to complete the right-handed coordinate system. We locate the origin of the coordinate frame $(\mathbf{x}_5,\mathbf{y}_5,\mathbf{z}_5)$ at the intersection of the $\mathbf{z}_5$ and $\mathbf{z}_4$ axes. The distance separating the origin of $(\mathbf{x}_5,\mathbf{y}_5,\mathbf{z}_5)$ and the origin of $(\mathbf{x}_4,\mathbf{y}_4,\mathbf{z}_4)$ is 72.0 mm. Establish $\mathbf{x}_5 = -(\mathbf{z}_4 \times \mathbf{z}_5) / \| \mathbf{z}_4 \times \mathbf{z}_5 \|$. Assign $\mathbf{y}_5 = +(\mathbf{z}_5 \times \mathbf{x}_5) / \| \mathbf{z}_5 \times \mathbf{x}_5 \|$ to complete the right-handed coordinate system.

D7. *Find joint and link parameters.* For each $i, i = 1, \ldots, 5$, perform steps **D8** to **D11**.

D8. *Find* d_i. d_i is the distance from the origin of the $(i - 1)$th coordinate system to the intersection of the $\mathbf{z}_{i-1}$ axis and the $\mathbf{x}_i$ axis along the $\mathbf{z}_{i-1}$ axis. It is the joint variable if joint i is prismatic.

D9. *Find* a_i. a_i is the distance from the intersection of the $\mathbf{z}_{i-1}$ axis and the $\mathbf{x}_i$ axis to the origin of the ith coordinate system along the $\mathbf{x}_i$ axis.

D10. *Find* θ_i. θ_i is the angle of rotation from the $\mathbf{x}_{i-1}$ axis to the $\mathbf{x}_i$ axis about the $\mathbf{z}_{i-1}$ axis. It is the joint variable if joint i is rotary.

D11. *Find* α_i. α_i is the angle of rotation from the $\mathbf{z}_{i-1}$ axis to the $\mathbf{z}_i$ axis.

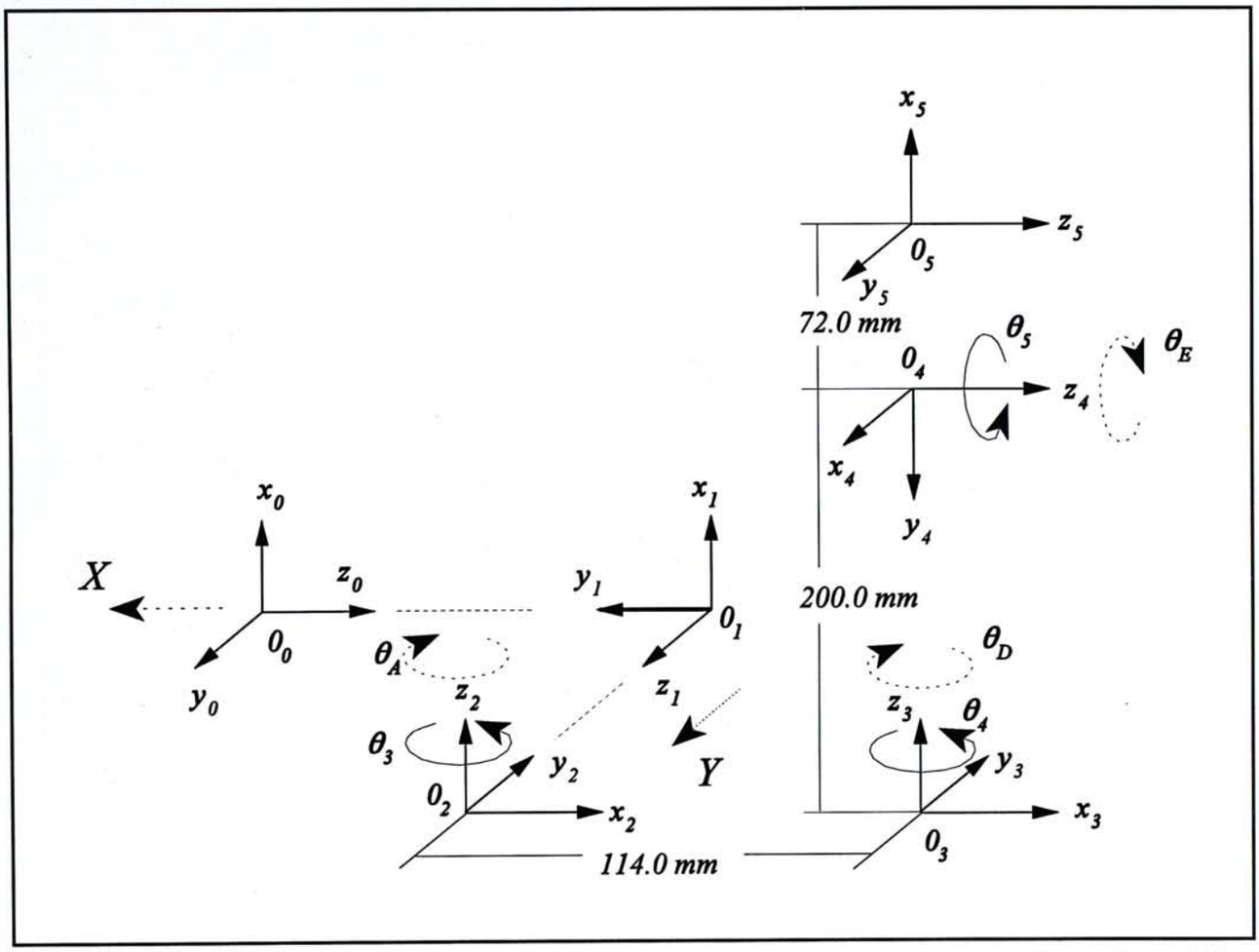


Figure 3.5 Link coordinate frames for each link of one manipulator of the active vision system

The coordinate frames for each link of one of two manipulators of the active vision system are shown in Fig.3.5. Six link coordinate frames ($\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$) , ($\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{z}_1$) , ... , ($\mathbf{x}_5$, $\mathbf{y}_5$, $\mathbf{z}_5$) are established for the manipulator. Among them, ($\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{z}_1$) , ... , ($\mathbf{x}_5$, $\mathbf{y}_5$, $\mathbf{z}_5$) represent the axes $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{A}$, $\boldsymbol{D}$, $\boldsymbol{E}$ of the active vision system , respectively. Link parameters and joint variables are tabulated in Table 3.1.

Axis	Joint	θ_i (°)	d_i (mm)	a_i (mm)	α_i (°)
X	1	0.0	$0.0 - \boldsymbol{X}$	0.0	-90.0
Y	2	-90.0	$0.0 + \boldsymbol{Y}$	0.0	-90.0
A	3	$0.0 - \theta_A$	0.0	114.0	0.0
D	4	$-90.0 - \theta_D$	200.0	0.0	-90.0
E	5	$-90.0 - \theta_E$	0.0	72.0	0.0

Table 3.1 Link parameters for links $i = 1, \ldots, 5$.

In Table 3.1, X , Y , θ_A , θ_D , θ_E are the joint variables of axes X , Y , A , D , E, respectively. Their positive directions are shown in Fig. 3.5 in dot line. Recalling that d_1 is the distance from the origin of the ($\mathbf{x}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$) coordinate system to the intersection of the $\mathbf{z}_0$ axis and the $\mathbf{x}_1$ axis along the $\mathbf{z}_0$ axis, thus, the positive direction of d_1 is anti-parallel to the positive direction of X. Hence, X takes a minus sign when d_1 is calculated. The signs of Y , θ_A , θ_D , θ_E are determined accordingly and are shown in Table 3.1.

Once the D-H coordinate system has been established for each link, the D-H transformation matrix for adjacent coordinate frames i and i - 1, denoted by , ${}^{i-1}\boldsymbol{A}_i$, relates the ith coordinate frame to the (i - 1)th coordinate frame. ${}^{i-1}\boldsymbol{A}_i$ can be calculated in the following equation,

$$
{}^{i-1}\boldsymbol{A}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\sin\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.37}
$$

Suppose that a point $\mathbf{p}$ is fixed on link i, $\mathbf{p}_{i-1} = (x_{i-1} , y_{i-1} , z_{i-1} , 1)^T$ and $\mathbf{p}_i = (x_i , y_i , z_i , 1)^T$ are the homogeneous coordinates of $\mathbf{p}$ in the ith coordinate frame and (i - 1)th frame, respectively. Thus,

$$
\boldsymbol{p}_{i-1} = {}^{i-1}\boldsymbol{A}_i\boldsymbol{p}_i \tag{3.38}
$$

Generally speaking, given a manipulator consists of n links, the homogeneous transformation matrix ${}^0\boldsymbol{T}_i$ which specifies the transformation from the ith coordinate frame to the base coordinate system is the chain product of successive coordinate transformation matrices of ${}^{i-1}\boldsymbol{A}_i$ and is given by

$$
\begin{aligned}
{}^0\boldsymbol{T}_i &= {}^0\boldsymbol{A}_1\,{}^1\boldsymbol{A}_2 \cdots {}^{i-1}\boldsymbol{A}_i = \prod_{j=1}^{i} {}^{j-1}\boldsymbol{A}_j \quad \text{for } i = 1,2,\cdots n \\
{}^0\boldsymbol{T}_i &= \begin{bmatrix} {}^0\boldsymbol{R}_i & {}^0\boldsymbol{p}_i \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.39}
$$

where

${}^0\boldsymbol{R}_i$ is the rotation matrix from the ith coordinate frame established at link i to the base coordinate system. It is the upper left 3×3 partitioned matrix of ${}^0\boldsymbol{T}_i$.

${}^0\boldsymbol{p}_i$ is the position vector which points from the origin of ith coordinate frame to the origin of the base coordinate system.

Thus, the resultant transformation matrix of robot head to robot world coordinate frame,denoted by $\boldsymbol{H}_{gi}$, is given by

$$\boldsymbol{H}_{gi} = {}^{0}\boldsymbol{T}_{5}\Big|_{q_i} = {}^{0}\boldsymbol{A}_{1}\,{}^{1}\boldsymbol{A}_{2}\,{}^{2}\boldsymbol{A}_{3}\,{}^{3}\boldsymbol{A}_{4}\,{}^{4}\boldsymbol{A}_{5}\Big|_{q_i} \tag{3.40}$$

where, subscript i denotes the ith observing position , q_i represents the set of joint variables recorded at the ith observing position.

From (3.6) , (3.10) and (3.40) , $\boldsymbol{R}_{gij}$ which is the rotation matrix from the robot head coordinate frame at ith observing position , denoted by G_i to the robot head coordinate frame at jth observing position , denoted by G_j , can be calculated using the following equation.

$$\boldsymbol{R}_{gij} = \boldsymbol{R}_{gj}^{-1}\boldsymbol{R}_{gi} \tag{3.41}$$

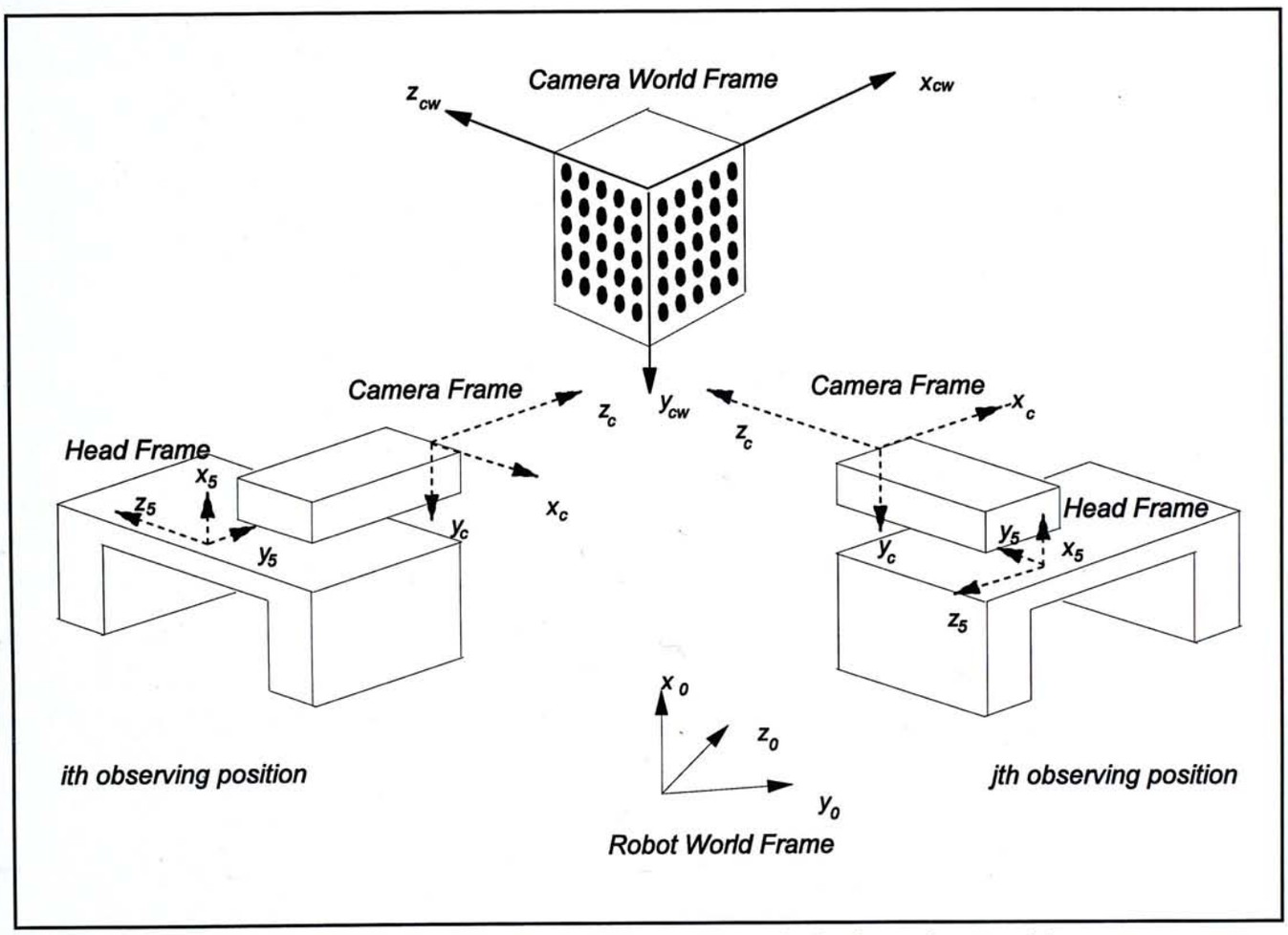


Figure 3.6 Head and camera coordinate frames at ith and jth observing position.

By performing the extrinsic calibration of camera using R. Y. Tsai's RAC based camera calibration method[5] at each observing position, the corresponding camera pose with respect to a camera coordinate frame which is attached on a calibration block can be calculated at the each observing station. The camera pose in ith observing positon is exactly the coordinate

transformation from CW to C_i, denoted by $\boldsymbol{H}_{ci}$.

From (3.5) , (3.7) and (3.11) , $\boldsymbol{R}_{cij}$ which is the rotation matrix from the camera coordinate frame at ith observing position , denoted by C_i to the camera coordinate frame at jth observing position , denoted by C_j , can be calculated using the following equation.

$$\boldsymbol{R}_{cij} = \boldsymbol{R}_{cj}\boldsymbol{R}_{ci}^{-1} \tag{3.42}$$

Robot head and camera coordinate frames at ith and jth observing position are shown in Fig. 3.6.

Following the steps described in **3.3.5** , $\boldsymbol{R}_{cg}$, $\boldsymbol{T}_{cg}$ can be estimated using at least two robot motions.

3.5 Function of Procedures and Formats of Data files

The flow chart of our local implementation of head-eye calibration for one of two manipulator of the active vision system is shown in Fig. 3.7. Procedures and data files are denoted by rectangles and shaded parallelograms, respectively.

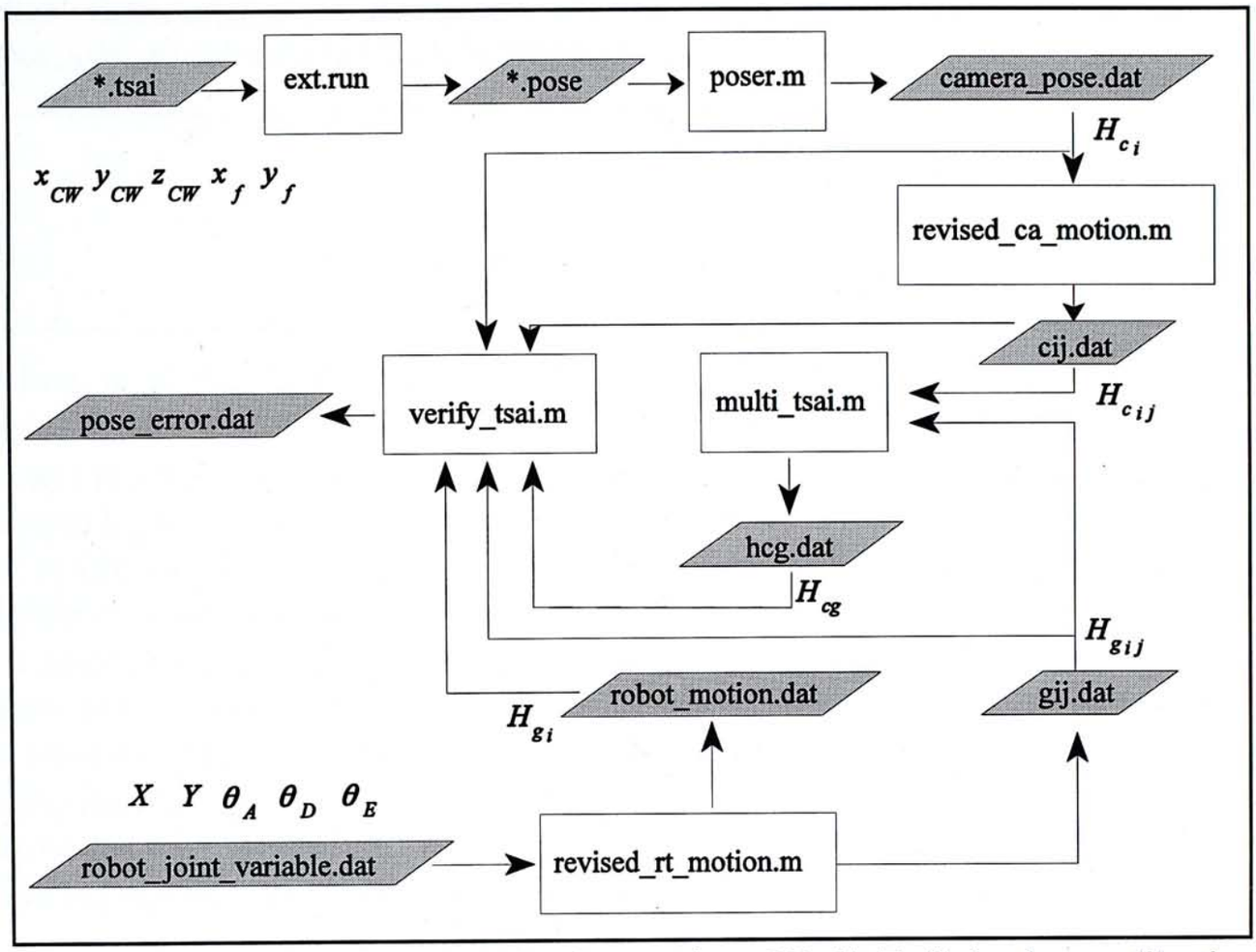


Figure 3.7 Flow chart of our local implementation of R. Y. Tsai's head-eye calibration approach.

The data file **.tsai* contains 3D world coordinates and corresponding image coordinates for a number of control points on the camera calibration block. In this experiment, fifty control points are utilized in the camera calibration. Referring to **2.4** for the aquisition and formats of the data file with extension **.tsai* which is suitable to be fed into the Tsai's camera calibration program. A batch command procedure *ext.run* was called to get the camera pose at each observing position. Actually , a procedure *n_cal_fo.exe* in the Reg Willson's implementation of R. Y. Tsai's RAC based camera calibration algorithm was called to perform the full scale camera calibration at the first observation station. Then, a procedure *ep_cal.exe* in the Reg Willson's implementation was called to perform extrinsic calibration at the remainning observing position using the set of intrinsic parameters which is dertermined in the first observing position. The resultant camera calibration result at each observing postion is stored in the data file **.pose*. A MATLAB program poser.m was developed to create the collective *camera_pose.dat* from all the data file **.pose*. *camera_pose.dat* contains all the $\boldsymbol{H}_{ci}$, where $i = 1, \ldots, n$, n is the number of

observing positions. Taking *camera_pose.dat* as input , a MATLAB program *revised_ca_motion.m* produce *cij.dat* which contains all the $\boldsymbol{H}_{cij}$, where $i = 1, \ldots, n$, $j = 2, \ldots, n$, n is the number of observing positions.

A MATLAB program *rt_joint_variable.m* is used to produce a collective *robot_joint_variable.dat* from the data files which contain the joint variables recorded by the active vision system at the each observing position, respectively. The data file *robot_joint_variable.dat* has the following format,

$$\text{number of observing positions}\left\{\begin{array}{c} X_0\ Y_0\ A_0\ B_0\ C_0\ D_0\ E_0 \\ \vdots \\ X_i\ Y_i\ A_i\ B_i\ C_i\ D_i\ E_i \\ \vdots \\ X_n\ Y_n\ A_n\ B_n\ C_n\ D_n\ E_n \end{array}\right.$$

where, $\boldsymbol{X}_i\ \boldsymbol{Y}_i\ \boldsymbol{A}_i\ B_i\ C_i\ \boldsymbol{D}_i\ \boldsymbol{E}_i$ are the joint variables recorded at the ith observing position. It should be noted that only $\boldsymbol{X}_i\ \boldsymbol{Y}_i\ \boldsymbol{A}_i\ \boldsymbol{D}_i\ \boldsymbol{E}_i$ are considered in our case.

A MATLAB program *revised_rt_motion.m* gives us two data files , *robot_motion.dat* which contains $\boldsymbol{H}_{gi}$ and *gij.dat* which contains $\boldsymbol{H}_{gij}$, from the data file *robot_joint_variable.dat*

A MATLAB program *multi_tsai.m* ,which implements the R.Y. Tsai's head-eye calibration algorithm, is used to get *hcg.dat* which contains $\boldsymbol{H}_{cg}$, from the data file *cij.dat* and *gij.dat.*

A MATLAB program *verify_tsai.m* are used to compare the difference between predicted camera pose and measured camera pose. Let $\boldsymbol{H}_{RC}$ represent the homogeneous transformation from the robot world frame RW to the camera calibration world frame CW. $\boldsymbol{H}_{RC}$ remains unchanged during the whole process of calibrating the head-eye relationship for the active vision system. The relationships between $\boldsymbol{H}_{RC}$, $\boldsymbol{H}_{ci}$, $\boldsymbol{H}_{cg}$ and $\boldsymbol{H}_{gi}$ are shown in Fig. 3.8. $\boldsymbol{H}_{RC}$ is approximated by an average $\overline{\boldsymbol{H}_{RC}}$ in the following equation,

$$\boldsymbol{H}_{RCi} = \boldsymbol{H}_{ci}^{-1}\boldsymbol{H}_{cg}^{-1}\boldsymbol{H}_{gi}^{-1}$$

$$\overline{\boldsymbol{H}_{RC}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{H}_{ci}^{-1}\boldsymbol{H}_{cg}^{-1}\boldsymbol{H}_{gi}^{-1}$$

where, N is the total number of observing positions.

For each observing positions, the predicted camera pose relative to the robot world frame is given by $\boldsymbol{H}_{cg}^{-1}\boldsymbol{H}_{gk}^{-1}$, the measured camera pose relative to the robot world frame is given by $\boldsymbol{H}_{ck}\overline{\boldsymbol{H}_{RC}}$,

$$\Delta\boldsymbol{H}_{ck} = \boldsymbol{H}_{ck}\overline{\boldsymbol{H}_{RC}} - \boldsymbol{H}_{cg}^{-1}\boldsymbol{H}_{gk}^{-1}$$

The resultant errors are stored in the data file *pose_error.dat.*

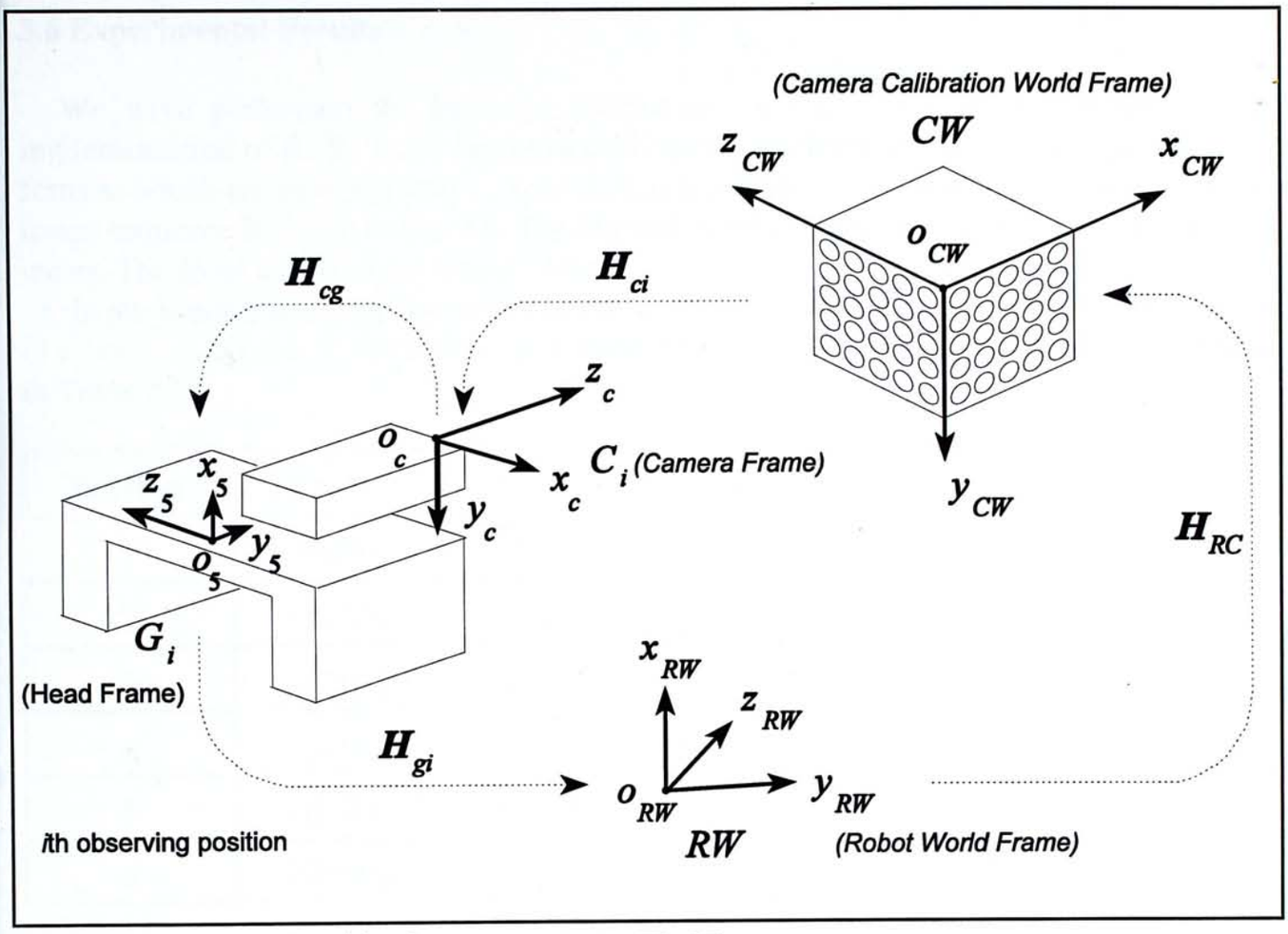


Figure 3.8 Relationships between H_{gi} H_{ci} H_{cg} H_{RC}

Procedures are listed in **Appendix I**.

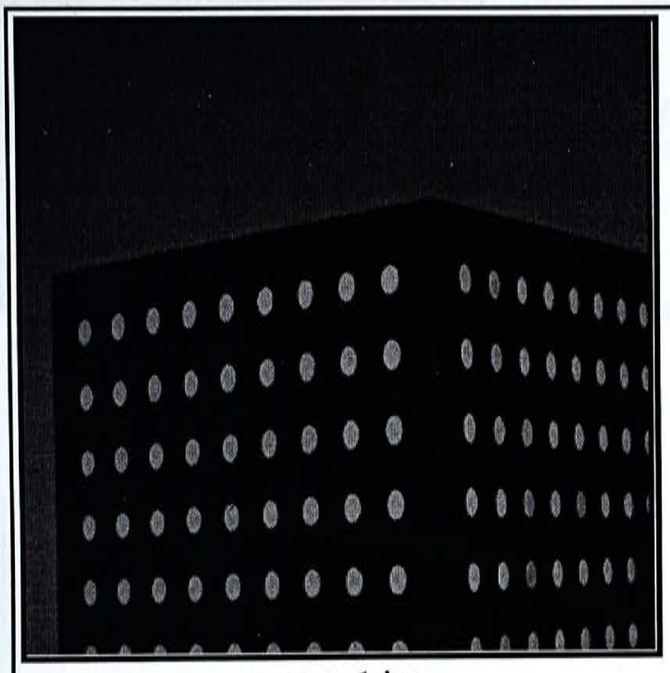
3.6 Experimental Results

We have performed the head-eye calibration for the active vision system with our implementation of R. Y. Tsai's head-eye calibration algorithm. A real image sequence of six frames, which are near_r_1.img .... near_r_6.img, is aquired by the active vision system. The image sequence is shown in Fig. 3.9. The distance separating the camera and object is about one meter. The focal length of the lens is 16mm.

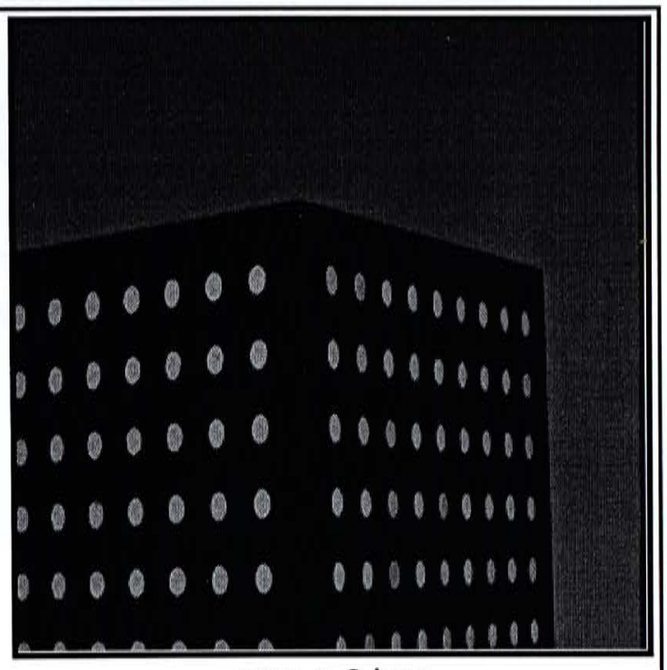
In our experiment, Only four joints of the active vision system are moved. The input values of robot joint variable to the active vision system and corresponding image frames are tabulated in Table 3.2:

Position	***X*** (mm)	***Y*** (mm)	***D*** (°)	***E*** (°)	Frame
1	0.000	98.000	0.000	6.000	near_r_1.img
2	-20.000	98.000	5.000	6.000	near_r_2.img
3	-20.000	78.000	5.000	3.000	near_r_3.img
4	-20.000	78.000	0.000	3.000	near_r_4.img
5	-20.000	98.000	0.000	6.000	near_r_5.img
6	-70.000	98.000	-4.000	6.000	near_r_6.img

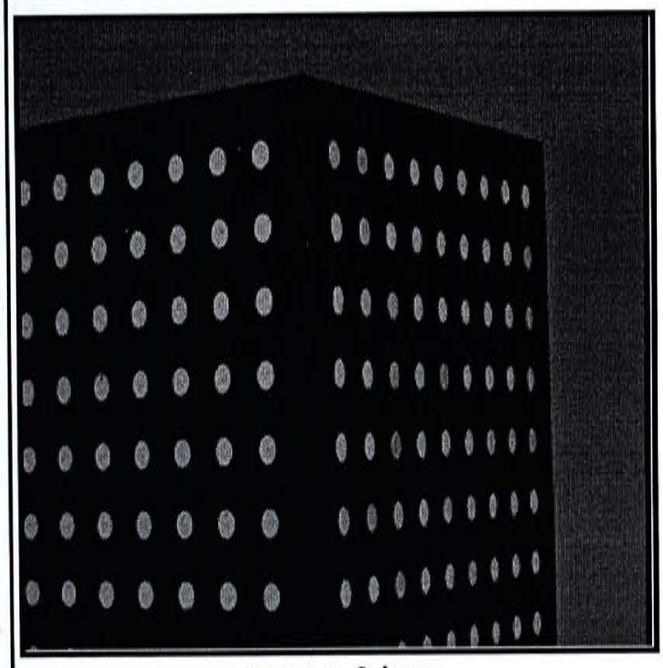
Table 3.2 Input values and corresponding image frames



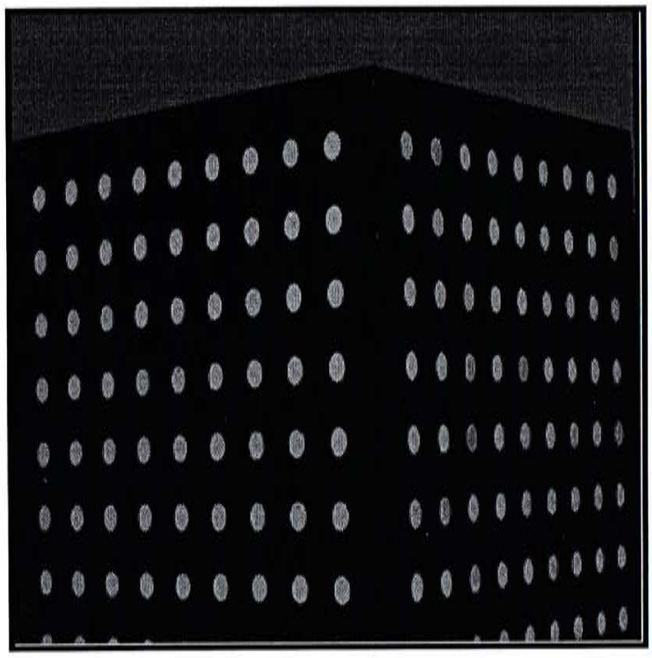
near_r_1.img



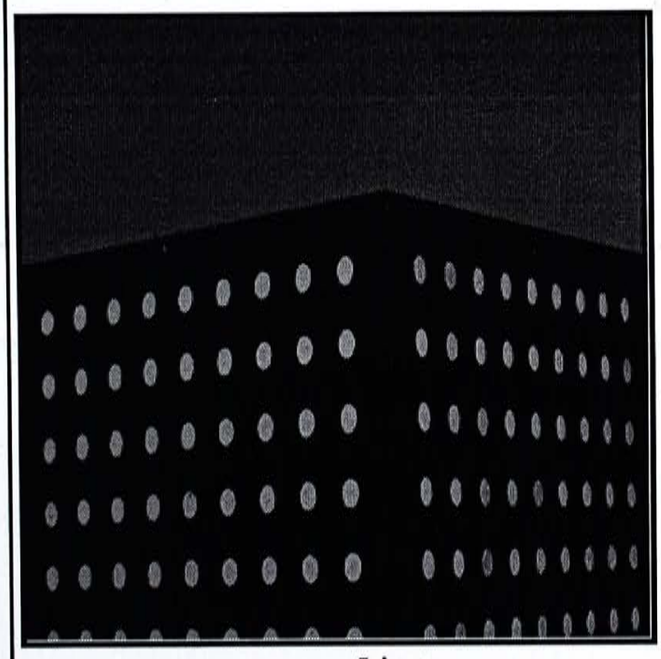
near_r_2.img



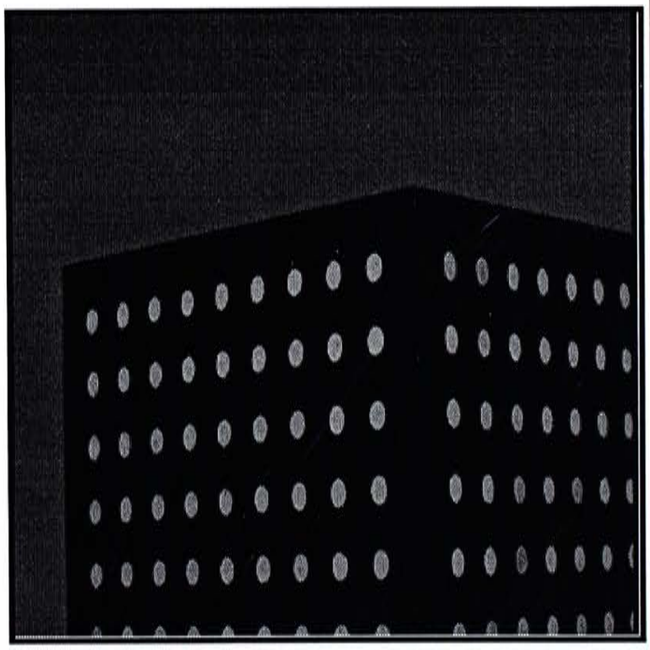
near_r_3.img



near_r_4.img



near_r_5.img



near_r_6.img

Figure 3.4 The images used for head-eye calibration of the active vision system.

A program named *sparse.exe* running on *PC* generates from image files *.img a set of data file *.get which include 3D world coordinates and corresponding image coordinates for a number of control points on the camera calibration block.

The following command line

```
>> source prepare.run
```

transforms the data files *.get to the data files *.tsai which are suitable to be fed into the Tsai's camera calibration program.

The content of *prepare.run* is as follows,

```
ws230b.ee.cuhk.hk> more prepare.run
new_prepare near_r_1.get near_r_1.tsai
new_prepare near_r_2.get near_r_2.tsai
new_prepare near_r_3.get near_r_3.tsai
new_prepare near_r_4.get near_r_4.tsai
new_prepare near_r_5.get near_r_5.tsai
new_prepare near_r_6.get near_r_6.tsai
exit
```

Fifty control points are utilized in each image of camera calibration block. The image coordinates of control points are obtained by detecting centroids of ellipse. Fig. 3.10 illustrates the detected centroids of ellipse labelled by plus signs. The content of near_r_1.tsai, ... , near_r_6.tsai are listed the Table 3.3 , ... , 3.8 , respectively.

near_r_1.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.582685e+02	2.082211e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.596092e+02	2.684481e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.609555e+02	3.289260e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.623147e+02	3.899844e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.636782e+02	4.512336e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.820886e+02	2.132005e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.834987e+02	2.719832e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.849149e+02	3.310202e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.863441e+02	3.906026e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.877777e+02	4.503652e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	4.048537e+02	2.179594e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	4.062887e+02	2.753555e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	4.077302e+02	3.330145e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	4.091846e+02	3.911901e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	4.106434e+02	4.495412e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	4.265673e+02	2.224985e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	4.280275e+02	2.785723e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	4.294949e+02	3.349169e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	4.309749e+02	3.917506e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	4.324594e+02	4.487550e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	4.472726e+02	2.268268e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	4.487561e+02	2.816396e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	4.502473e+02	3.367308e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	4.517509e+02	3.922850e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	4.532591e+02	4.480055e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.969536e+02	2.085106e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.619026e+02	2.143481e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.280186e+02	2.199913e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.956222e+02	2.253867e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.642173e+02	2.306169e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.981290e+02	2.690090e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.629550e+02	2.737460e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.290076e+02	2.783179e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.964746e+02	2.826993e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.650109e+02	2.869367e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.993140e+02	3.300047e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.640152e+02	3.335796e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.300030e+02	3.370242e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.973319e+02	3.403330e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.658082e+02	3.435256e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	3.005059e+02	3.913514e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.650809e+02	3.937235e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.310031e+02	3.960053e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.981927e+02	3.982023e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.666084e+02	4.003171e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	3.017024e+02	4.529396e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.661513e+02	4.541287e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.320080e+02	4.552707e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.990579e+02	4.563728e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.674132e+02	4.574312e+02

Table 3.3 The data file *near_r_1.tsai* is the input data file to camera calibration package at first observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

near_r_2.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	2.580241e+02	2.102634e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	2.590695e+02	2.701426e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	2.601235e+02	3.305145e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	2.611797e+02	3.910127e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	2.622454e+02	4.520550e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	2.810828e+02	2.155310e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	2.821979e+02	2.739299e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	2.833217e+02	3.327874e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	2.844484e+02	3.917967e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	2.855834e+02	4.512383e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.030518e+02	2.205497e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.041758e+02	2.775289e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.053084e+02	3.349417e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.064446e+02	3.925378e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.075874e+02	4.504684e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.239543e+02	2.253247e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.251224e+02	2.809589e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.262989e+02	3.369983e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.274799e+02	3.932466e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.286658e+02	4.497308e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.438592e+02	2.298719e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.450511e+02	2.842223e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.462513e+02	3.389532e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.474567e+02	3.939196e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.486653e+02	4.490310e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	1.973269e+02	2.097048e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	1.618349e+02	2.150798e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	1.276024e+02	2.202640e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	9.459345e+01	2.252630e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	6.257803e+01	2.301115e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	1.982148e+02	2.701065e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	1.626282e+02	2.744244e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	1.282914e+02	2.785906e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	9.516869e+01	2.826096e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	6.310904e+01	2.864995e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	1.991089e+02	3.309377e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	1.634272e+02	3.341943e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.289854e+02	3.373376e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	9.574810e+01	3.403711e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	6.364395e+01	3.433011e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.000070e+02	3.920386e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	1.642302e+02	3.942628e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.296832e+02	3.964105e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	9.633103e+01	3.984839e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	6.418244e+01	4.004825e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.009103e+02	4.534929e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	1.650364e+02	4.545753e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.303826e+02	4.556209e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	9.691430e+01	4.566307e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	6.472034e+01	4.576021e+02

Table 3.4 The data file *near_r_2.tsai* is the input data file to camera calibration package at 2th observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

near_r_3.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	2.545688e+02	1.095884e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	2.556194e+02	1.687726e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	2.566710e+02	2.280120e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	2.577303e+02	2.876879e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	2.587894e+02	3.473508e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	2.773963e+02	1.147906e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	2.784462e+02	1.724884e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	2.794990e+02	2.303516e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	2.805575e+02	2.885228e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	2.816168e+02	3.467403e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	2.991348e+02	1.197445e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.002128e+02	1.760316e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.012958e+02	2.325854e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.023824e+02	2.893210e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.034709e+02	3.461558e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.198909e+02	1.244746e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.209687e+02	1.794103e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.220536e+02	2.347128e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.231398e+02	2.900802e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.242290e+02	3.456007e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.397341e+02	1.289966e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.407729e+02	1.826340e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.418206e+02	2.367386e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.428676e+02	2.908018e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.439186e+02	3.450741e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	1.946803e+02	1.089808e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	1.598221e+02	1.143299e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	1.259898e+02	1.195215e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	9.349442e+01	1.245079e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	6.190529e+01	1.293553e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	1.955952e+02	1.686622e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	1.606272e+02	1.729577e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	1.267605e+02	1.771180e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	9.418669e+01	1.811194e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	6.255538e+01	1.850051e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	1.965109e+02	2.283940e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	1.614339e+02	2.316940e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.275336e+02	2.348833e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	9.488170e+01	2.379551e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	6.320875e+01	2.409349e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	1.974330e+02	2.885426e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	1.622451e+02	2.907655e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.283101e+02	2.929092e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	9.557898e+01	2.949770e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	6.386347e+01	2.969805e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	1.983569e+02	3.488092e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	1.630588e+02	3.500118e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.290898e+02	3.511691e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	9.627979e+01	3.522870e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	6.452219e+01	3.533690e+02

Table 3.5 The data file *near_r_3.tsai* is the input data file to camera calibration package at third observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

near_r_4.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.223910e+02	1.075918e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.235328e+02	1.669544e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.246821e+02	2.267100e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.258328e+02	2.865331e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.269904e+02	3.467193e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.467019e+02	1.125541e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.478626e+02	1.705562e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.490296e+02	2.288678e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.501991e+02	2.873080e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.513744e+02	3.460315e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.698477e+02	1.172786e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.710196e+02	1.739845e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.721962e+02	2.309210e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.733767e+02	2.880451e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.745615e+02	3.453775e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.920063e+02	1.218015e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.931944e+02	1.772673e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.943859e+02	2.328877e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.955825e+02	2.887513e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.967821e+02	3.447507e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	4.132023e+02	1.261280e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	4.143630e+02	1.804012e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	4.155256e+02	2.347612e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	4.166946e+02	2.894227e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	4.178652e+02	3.441560e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.617459e+02	1.077267e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.276566e+02	1.135430e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	1.949266e+02	1.191274e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.634683e+02	1.244948e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.329215e+02	1.297067e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.627898e+02	1.673818e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.286458e+02	1.720557e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	1.958246e+02	1.765485e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.642636e+02	1.808689e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.337129e+02	1.850509e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.638412e+02	2.274633e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.296424e+02	2.310090e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.967296e+02	2.344214e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.650654e+02	2.377044e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.345111e+02	2.408723e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.648948e+02	2.876735e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.306405e+02	2.900469e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.976353e+02	2.923338e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.658672e+02	2.945350e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.353087e+02	2.966524e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.659523e+02	3.481075e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.316426e+02	3.493269e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.985449e+02	3.505032e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.666728e+02	3.516360e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.361104e+02	3.527221e+02

Table 3.6 The data file *near_r_4.tsai* is the input data file to camera calibration package at 4th observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

near_r_5.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.242568e+02	2.078681e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.255365e+02	2.681918e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.268208e+02	3.287294e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.281109e+02	3.895432e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.294123e+02	4.508857e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.488340e+02	2.128179e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.501606e+02	2.717221e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.514914e+02	3.308173e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.528301e+02	3.902579e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.541765e+02	4.500456e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.722520e+02	2.175343e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.736128e+02	2.750844e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.749777e+02	3.328050e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.763523e+02	3.909380e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.777311e+02	4.492465e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.946396e+02	2.220432e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.960513e+02	2.783013e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.974668e+02	3.347083e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.988942e+02	3.915897e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	4.003218e+02	4.484801e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	4.160401e+02	2.263532e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	4.174508e+02	2.813693e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	4.188649e+02	3.365192e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	4.202929e+02	3.922085e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	4.217171e+02	4.477542e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.628211e+02	2.081245e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.282654e+02	2.139750e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	1.949930e+02	2.196082e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.629895e+02	2.250265e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.321591e+02	2.302463e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.639011e+02	2.686645e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.292510e+02	2.733798e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	1.959046e+02	2.779177e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.638149e+02	2.822846e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.329059e+02	2.864908e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.649885e+02	3.296227e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.302424e+02	3.331276e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.968203e+02	3.364988e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.646431e+02	3.397445e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.336544e+02	3.428703e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.660826e+02	3.909566e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.312405e+02	3.932862e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.977431e+02	3.955259e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.654782e+02	3.976832e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.344098e+02	3.997605e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.671812e+02	4.525383e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.322428e+02	4.536912e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.986697e+02	4.547991e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.663168e+02	4.558668e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.351684e+02	4.568946e+02

Table 3.7 The data file *near_r_5.tsai* is the input data file to camera calibration package at 5th observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

near_r_6.tsai

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.588054e+02	2.076266e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.601457e+02	2.680860e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.614955e+02	3.289771e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.628501e+02	3.900861e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.642166e+02	4.517265e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.852433e+02	2.122859e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.866383e+02	2.714302e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.880426e+02	3.309672e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.894507e+02	3.906666e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.908710e+02	4.508818e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	4.104449e+02	2.167272e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	4.118922e+02	2.746181e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	4.133484e+02	3.328642e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	4.148073e+02	3.912200e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	4.162787e+02	4.500766e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	4.346427e+02	2.209916e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	4.361177e+02	2.776762e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	4.376011e+02	3.346822e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	4.390861e+02	3.917498e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	4.405839e+02	4.493063e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	4.576964e+02	2.250544e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	4.592183e+02	2.805922e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	4.607480e+02	3.364174e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	4.622782e+02	3.922559e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	4.638213e+02	4.485699e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.967555e+02	2.087011e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.634610e+02	2.149008e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.316340e+02	2.208274e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	2.010831e+02	2.265163e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.716133e+02	2.320038e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.979032e+02	2.691519e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.645433e+02	2.741634e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.325993e+02	2.789622e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	2.019524e+02	2.835661e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.724312e+02	2.880009e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.990630e+02	3.302383e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.656357e+02	3.339748e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.335722e+02	3.375589e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	2.028275e+02	3.409955e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.732536e+02	3.443013e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	3.002298e+02	3.916942e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.667344e+02	3.941323e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.345505e+02	3.964749e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	2.037071e+02	3.987199e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.740800e+02	4.008764e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	3.014009e+02	4.533788e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.678374e+02	4.545265e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.355327e+02	4.556313e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	2.045905e+02	4.566895e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.749101e+02	4.577044e+02

Table 3.8 The data file *near_r_6.tsai* is the input data file to camera calibration package at 6th observing position. It contains the 3D coordinates of control points ,which are expressed with respect to the camera world coordinate system *CW*, and corresponding image coordinates.

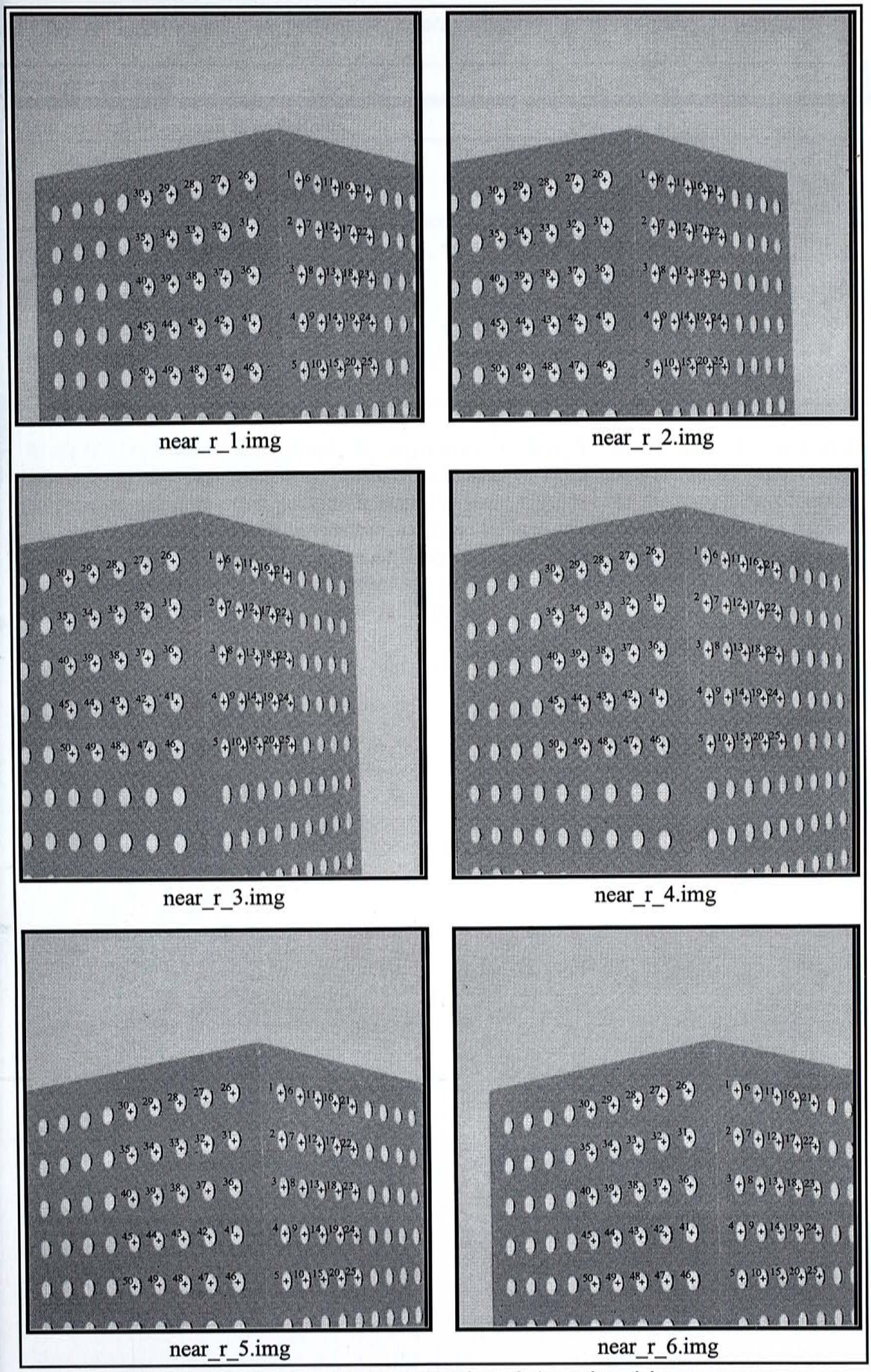


Figure 3.4 The images used for head-eye calibration of the active vision system.

The following command lines performed camera calibration at all observing positions.

```
>>source ext.run
```

The content of *ext.run* is as follows,

```
ws230b.ee.cuhk.hk> more ext.run
../../../camera/Tsai/n_cal_fo near_r_1.tsai > near_r_1.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_2.tsai > near_r_2.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_3.tsai > near_r_3.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_4.tsai > near_r_4.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_5.tsai > near_r_5.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_6.tsai > near_r_6.pose
```

In the batch command file *ext.run* , *n_cal_fo.exe* was called to perform the full scale camera calibration at the first observing position with *near_r_1.tsai* taken as input. The camera calibration result at the first position is stored in *near_r_1.pose*. At the remaining observing positions, *ep_cal.exe* was called to perform extrinsic calibration of the camera using the intrinsic camera parameters stored in *near_r_1.pose*. Thus, at the *i*th observing position, *ex_cal.exe* takes *near_r_1.pose* and *near_r_i.tsai* as inputs and stores the calibration result into *near_r_i.pose*. The camera calibration results are listed in Table 3.9.

	near_r_1.pose	*near_r_2.pose*	*near_r_3.pose*	*near_r_4.pose*	*near_r_5.pose*	*near_r_6.pose*
Ncx	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02	7.5200000000e+02
Nfx	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02	5.1200000000e+02
dx	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03	8.6000000000e-03
dy	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03
dpx	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02	1.2631250000e-02
dpy	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03	8.3000000000e-03
Cx	2.4864442949e+02	2.4864442949e+02	2.4864442949e+02	2.4864442949e+02	2.4864442949e+02	2.4864442949e+02
Cy	2.5929119238e+02	2.5929119238e+02	2.5929119238e+02	2.5929119238e+02	2.5929119238e+02	2.5929119238e+02
sx	1.1054204813e+00	1.1054204813e+00	1.1054204813e+00	1.1054204813e+00	1.1054204813e+00	1.1054204813e+00
f	1.6086029215e+01	1.6086029215e+01	1.6086029215e+01	1.6086029215e+01	1.6086029215e+01	1.6086029215e+01
κ_1	2.0046547932e-04	2.0046547932e-04	2.0046547932e-04	2.0046547932e-04	2.0046547932e-04	2.0046547932e-04
Tx	5.6318026317e+01	-9.9049894294e+00	-1.2205211366e+01	3.2636103538e+01	3.3200250506e+01	5.4790392699e+01
Ty	-2.7362830257e+01	-2.6572083047e+01	-7.6558352246e+01	-7.7100712069e+01	-2.7533605338e+01	-2.7414434984e+01
Tz	9.3716252103e+02	9.4243553800e+02	9.5423771981e+02	9.4945509466e+02	9.3727491279e+02	9.3432002590e+02
Rx	-1.6290723138e-01	-1.7994653085e-01	-8.7672075592e-02	-8.1092214446e-02	-1.6294213935e-01	-1.5213733225e-01
Ry	-9.0101718292e-01	-9.8572613431e-01	-9.9068667768e-01	-9.0761451954e-01	-9.0366714604e-01	-8.3221580709e-01
Rz	1.0462295130e-01	1.2730649592e-01	5.0134869383e-02	4.1073883840e-02	1.0479707847e-01	8.9905331538e-02
p1	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00
p2	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00	0.0000000000e+00

Table 3.9 Camera calibration results at the six observing positions. The unit of quantities are listed as follows: Ncx [sel] , Nfx [pix] , dx [mm/sel] , dy [mm/sel] , dpx [mm/pix] , dpy [mm/pix] , Cx [pix] , Cy [pix] , sx [] , f [mm] , kapp1 [1 / mm^2] , Tx [mm] , Ty [mm] , Tz [mm] , Rx [rad], Ry [rad], Rz [rad].

Only the extrinsic parameters of the camera are used in the calculation of $\boldsymbol{H}_{ci}$, A MATLAB program *poser.m* was developed to create the collective *camera_pose.dat* which is tabulated in Table 3.10.

camera_pose.dat				
$\boldsymbol{H}_{c1}$	0.61741826	0.02340367	-0.78628682	56.31802632
	0.06483285	0.99464273	0.08051422	-27.36283026
	0.78395880	-0.10068816	0.61259326	937.16252103
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c2}$	0.54778874	0.02308799	-0.83629806	-9.90498943
	0.07011626	0.99483532	0.07339207	-26.57208305
	0.83367332	-0.09884144	0.54334074	942.43553800
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c3}$	0.54742695	0.02322183	-0.83653122	-12.20521137
	0.02746820	0.99857770	0.04569540	-76.55835225
	0.83640255	-0.04799290	0.54601048	954.23771981
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c4}$	0.61510812	0.02285245	-0.78811152	32.63610354
	0.02527910	0.99849435	0.04868273	-77.10071207
	0.78803742	-0.04986789	0.61360429	949.45509466
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c5}$	0.61533873	0.02352289	-0.78791174	33.20025051
	0.06472281	0.99467185	0.08024250	-27.53360534
	0.78560117	-0.10037219	0.61053765	937.27491279
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c6}$	0.67051994	0.02286089	-0.74153920	54.79039270
	0.06044627	0.99451859	0.08531715	-27.41443498
	0.73942495	-0.10203012	0.66546269	934.32002590
	0.00000000	0.00000000	0.00000000	1.00000000

Table 3.10 The homogeneous transformations from the camera world coordinate frame *CW* to the camera coordinate frame C_i , i = 1, ..., 6. The unit of translational components of $\boldsymbol{H}_{ci}$ is mm.

A MATLAB program *rt_joint_variable.m* is used to produce a collective data file *robot_joint_variable.dat* from *near_r_1.dat,..., near_r_6.dat*, which are the data files containing the joint variables recorded by the active vision system at the each observing position, respectively. The content of *robot_joint_variable.dat* is tabulated in Table 3.11.

Observing Positions	*robot_joint_variable.dat*						
	X (mm)	***Y*** (mm)	***A*** (°)	***B*** (°)	***C*** (°)	***D*** (°)	***E*** (°)
1	-0.001	-97.999	0.001	0.000	0.000	-0.002	-6.000
2	-19.999	-97.999	0.001	0.000	0.000	4.999	-5.999
3	-20.001	-78.001	0.001	0.000	0.000	4.999	-3.001
4	19.999	-78.001	0.001	0.000	0.000	-0.001	-3.001
5	19.999	-97.999	0.001	0.000	0.000	-0.001	-5.999
6	69.999	-97.999	0.001	0.000	0.000	-4.001	-5.995

Table 3.11 Measured robot joint variables at six observing positions

Taking *robot_joint_variable.dat* as input, a MATLAB program *revised_rt_motion.m* gives us two data files *robot_motion.dat* and *gij.dat*. The data file *robot_motion.dat* contains $\boldsymbol{H}_{g1}, \ldots, \boldsymbol{H}_{g6}$, where $\boldsymbol{H}_{gi}$ is the homogeneous coordinate transformation matrix from the head coordinate frame at the *i*th observing positon G_i to the robot world coordinate frame RW, respectively. The data file *gij.dat* contains $\boldsymbol{H}_{g12}, \boldsymbol{H}_{g23}, \boldsymbol{H}_{g34}, \boldsymbol{H}_{g45}, \boldsymbol{H}_{g56}$, where $\boldsymbol{H}_{gij}$ is the homogeneous coordinate transformation matrix from G_i to G_j, respectively. The data file *robot_motion.dat* is shown in Table 3.12. The data file *gij.dat* is listed in Table 3.13. The link parameters employed in the calculation of $\boldsymbol{H}_{gi}$ are tabulated in Table 3.1.

	robot_motion.dat			
$\boldsymbol{H}_{g1}$	0.99452190	0.10452846	0.00000000	271.60557647
	-0.10452846	0.99452190	-0.00001745	90.47494032
	-0.00000182	0.00001736	1.00000000	114.00086863
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g2}$	0.99452372	0.10451111	0.00000000	271.60570781
	-0.10411341	0.99073926	0.08715574	90.50482421
	0.00910874	-0.08667845	0.99619470	134.65482948
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g3}$	0.99862862	0.05235339	0.00000000	271.90126072
	-0.05215417	0.99482854	0.08715574	74.24788978
	0.00456290	-0.08703622	0.99619470	134.32952865
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g4}$	0.99862862	0.05235339	0.00000000	271.90126072
	-0.05235339	0.99862862	0.00000000	74.23354591
	0.00000000	0.00000000	1.00000000	94.00099998
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g5}$	0.99452372	0.10451111	0.00000000	271.60570781
	-0.10451111	0.99452372	0.00000000	90.47619007
	0.00000000	0.00000000	1.00000000	94.00099998
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g6}$	0.99453101	0.10444167	0.00000000	271.60623297
	-0.10418726	0.99210839	-0.06975647	90.49950697
	-0.00728548	0.06937498	0.99756405	43.47644521
	0.00000000	0.00000000	0.00000000	1.00000000

Table 3.12 The homogeneous transformation matrices from the head coordinate frame at the ith observing positon G_i , $i = 1, ..., 6$, to the robot world coordinate frame RW. The unit of the tranlational components of $\boldsymbol{H}_{gi}$ is mm.

	gij.dat			
$\boldsymbol{H}_{g12}$	0.99995841	0.00041313	0.00911056	-0.18515093
	0.00037829	0.99623476	-0.08669575	1.76063252
	-0.00911207	0.08669559	0.99619318	-20.57797085
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g23}$	0.99863136	0.05230110	0.00000000	-1.14153013
	-0.05230110	0.99863136	0.00000000	16.12907615
	0.00000000	0.00000000	1.00000000	1.74094815
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g34}$	0.99998957	0.00019895	-0.00456290	-0.00075095
	0.00019895	0.99620513	0.08703622	0.01432420
	0.00456290	-0.08703622	0.99619470	40.32852867
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g45}$	0.99863136	-0.05230110	0.00000000	1.99147108
	0.05230110	0.99863136	0.00000000	-16.12280633
	0.00000000	0.00000000	1.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{g56}$	0.99997341	0.00032283	-0.00728548	-0.36618874
	0.00018338	0.99759064	0.06937498	3.48195206
	0.00729033	-0.06937447	0.99756405	50.40310600
	0.00000000	0.00000000	0.00000000	1.00000000

Table 3.13 Homogeneous coordinate transformation matrices from G_i, which is the robot head coordinate frame at the *i*th observing position, to G_j, which is the robot head coordinate frame at the *j*th observing position. The unit of the translational components of $\boldsymbol{H}_{gij}$ is mm.

Given *camera_pose.dat*, a MABLAB program *revised_ca_motion.m* calculates us *cij.dat* The data file *cij.dat* contains $\boldsymbol{H}_{c12}, \boldsymbol{H}_{c23}, \boldsymbol{H}_{c34}, \boldsymbol{H}_{c45}, \boldsymbol{H}_{c56}$, where $\boldsymbol{H}_{cij}$ is the homogeneous coordinate transformation matrix from C_i which is the camera coordinate frame at the *i*th observing position, to C_j, which is the camera coordinate frame at the *j*th observing position, respectively. The data file *cij.dat* is shown in Table 3.14.

	cij.dat			
$\boldsymbol{H}_{c12}$	0.99632525	-0.00885488	-0.08519144	13.57986638
	0.00886663	0.99996066	-0.00024040	0.51561042
	0.08519022	-0.00051585	0.99636457	3.86814974
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c23}$	0.99999990	0.00009067	-0.00044153	-1.88170113
	-0.00011305	0.99870001	-0.05097318	-1.98299464
	0.00043633	0.05097322	0.99869992	14.38620839
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c34}$	0.99653733	0.00370278	0.08306410	-34.18036177
	-0.00369930	0.99999315	-0.00019586	-0.40114304
	-0.08306425	-0.00011210	0.99654420	-2.50736628
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c45}$	0.99999972	0.00068499	0.00027088	0.35977889
	-0.00069783	0.99871677	0.05063895	1.41153478
	-0.00023585	-0.05063914	0.99871698	-14.85863146
	0.00000000	0.00000000	0.00000000	1.00000000
$\boldsymbol{H}_{c56}$	0.99740209	0.00663406	0.07172905	-45.37078753
	-0.00663350	0.99997797	-0.00024601	0.56937381
	-0.07172910	-0.00023043	0.99742412	1.83449809
	0.00000000	0.00000000	0.00000000	1.00000000

Table 3.14 Homogeneous coordinate transformation matrices from C_i, which is the camera coordinate frame at the *i*th observing position, to C_j, which is the camera coordinate frame at the *j*th observing position. The unit of the translational components of $\boldsymbol{H}_{cij}$ is mm.

Taking *cij.dat* and *gij.dat* as inputs, a MATLAB program *multi_tsai.m*, which implements R. Y. Tsai's head-eye calibration algorithm, is used to get *hcg.dat*. The data file *hcg.dat* contains $\boldsymbol{H}_{cg}$, where $\boldsymbol{H}_{cg}$ is the homogeneous coordinate transformation matrix from the camera coordinate frame C_i to the head coordinate frame G_i. Since the camera is mounted rigidly on the robot head (last link of the manipulator) of the active vision system, $\boldsymbol{H}_{cg}$ remains unchanged throughout all the observing positions. In *multi_tsai.m* , $\boldsymbol{H}_{cg}$ is estimated by applying pseudo-inverse method on a linear system involving all $\boldsymbol{H}_{gij}$s and corresponding $\boldsymbol{H}_{cij}$s. The data file *hcg.dat* is shown in Table 3.15,

hcg.dat			
0.00226311	-0.99987558	-0.01561078	27.18797903
0.00351489	-0.01560277	0.99987209	71.13340852
-0.99999126	-0.00231769	0.00347914	-26.79699583
0.00000000	0.00000000	0.00000000	1.00000000

Table 3.15 Homogeneous transformation matrix from the camera coordinate frame to the robot head coordinate frame.

Thus the homogeneous coordinate transformation from the camera coordinate frame to the robot head coordinate frame can be written in the following matrix form,

$$\boldsymbol{H}_{cg} = \begin{bmatrix} 0.00226311 & -0.99987558 & -0.01561078 & 27.18797903 \\ 0.00351489 & -0.01560277 & 0.99987209 & 71.13340852 \\ -0.99999126 & -0.00231769 & 0.00347914 & -26.79699583 \\ 0.00000000 & 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix} \quad (3.43)$$

where the unit of the translational components of $\boldsymbol{H}_{cg}$ is mm.

A MATLAB program *verify_tsai.m* was used to compare the difference between predicted camera pose and measured camera pose at each observing position.

	pose_error.dat			
ΔH_{c1}	-0.00050748	0.00058341	0.00000257	0.76181331
	-0.00004221	-0.00046658	0.00055555	-0.38890401
	-0.00046627	0.00003771	0.00053554	0.05103441
	0.00000000	0.00000000	0.00000000	0.00000000
ΔH_{c2}	0.00076347	0.00211394	-0.00017179	-0.83915288
	-0.00000404	-0.00011727	-0.00056396	-0.21146569
	-0.00007708	0.00018509	0.00217272	-0.08752664
	0.00000000	0.00000000	0.00000000	0.00000000
ΔH_{c3}	0.00080795	0.00155168	-0.00012527	-0.96786484
	0.00004972	0.00119133	-0.00085079	0.38104413
	0.00125285	0.00008325	0.00158839	-0.62217463
	0.00000000	0.00000000	0.00000000	0.00000000
ΔH_{c4}	0.00025389	-0.00246851	-0.00000298	-0.05538839
	0.00003709	0.00095829	-0.00034344	0.14175561
	0.00096339	-0.00003217	-0.00245453	-0.10710857
	0.00000000	0.00000000	0.00000000	0.00000000
ΔH_{c5}	-0.00059443	-0.00203952	-0.00000425	0.55581501
	-0.00006293	-0.00070325	0.00040969	-0.22415075
	-0.00069789	0.00006537	-0.00208524	0.25364531
	0.00000000	0.00000000	0.00000000	0.00000000
ΔH_{c6}	-0.00087576	0.00024368	0.00001769	0.62587904
	-0.00006991	-0.00083388	0.00083508	0.31917257
	-0.00077136	0.00005451	0.00016903	0.33978131
	0.00000000	0.00000000	0.00000000	0.00000000

Table 3.16 Error matrices between predicted camera pose and measured camera pose at each observing position. The unit of the translational components of the error matrices is mm.

3.7 Discussion

The Inherent Difficulties of Head Eye Calibration

1. The accuracy of the head eye calibration depends upon the accuracies of the estimation of the relative motion of camera coordinate frame which in turn depends on the accuracies of the estimation of the camera pose at the every image aquisition position. It is obviously that the error incurred in the process of estimating the pose of camera at any position will propagate and be magnified in the process of estimating the relative motion of the camera and of course the head eye geometrical relationship.

2. There exists a contradiction. When the acitve vision system is being used in an applicaiton, the typical range of the distance which separates the camera and the object of interest is usually within 1-2 meters. As an typical active vision system, although our active vision system has 5 DOF's for each camera, it suffers from rather short range of translation. The range of view angle of the active vision system is limited to be small. However, the intersation angle is required to be as large as possible since Tsai's investigation on this respect showed that larger interstation angles yield more accurate result. One alternative to deal with this contradiction is using wide angle lens. But it also introduces more error due to the lens distorsions.

The Analysis of the Experimental Results

We have performed the head-eye calibraiton method with a real image sequence which were taken by a camera fixed on the last link of the active system. The experimental results show that the estimated head-eye transformation can predict new camera pose quite satisfactorily. The performance of this method can act as a baseline with which other head-eye calibration method can be compared.

Improvements to Tsai's Method

Theoretically, the magnitude of $\boldsymbol{P}_{cij}$ should be equal to $\boldsymbol{P}_{gij}$. In the presence of noise, this relation is not accurately satisfied. We have taken this fact into account from two ways. At first, measure the difference between the length of $\boldsymbol{P}_{cij}$ and $\boldsymbol{P}_{gij}$. Those motion with small differences are chosen for being used in the estimation of the $\boldsymbol{R}_{cg}$. On the other hand, the length of $\boldsymbol{P}_{cij}$'s are replaced by $\boldsymbol{P}_{gij}$'s while $\boldsymbol{P}_{cij}$'s directions are kept intouched. With this improvement, the resultant estimation for Rcg appears more stable.

Our New Feature Correspondenced Based Head-Eye Calibration Technique

1 Estimation of $\boldsymbol{R}_{cg}$.

We are also using three non_coplanar translation of robot to estimate the $\boldsymbol{R}_{cg}$. The orthonormality of the $\boldsymbol{R}_{cg}$ are ensured by an adoption of SVD based absolute orientation technique mentioned by Haralick[9]

2 Estimation of $\boldsymbol{T}_{cg}$.

It is well known that the enssential matrix $\boldsymbol{E}$ can be estimated given eight feature point

correspondences across two images. Since $\boldsymbol{E} = \boldsymbol{T}_{cij} \times \boldsymbol{R}_{cij}$, and $\boldsymbol{R}_{cij}$ can be calculated as $\boldsymbol{R}_{cg}^{-1}\boldsymbol{R}_{gij}\boldsymbol{R}_{cg}$, we can get $\boldsymbol{T}_{cij}$ immediately. Finally, $\boldsymbol{T}_{cg}$ can be calculated with the following equation.

$$(\boldsymbol{R}_{gij}-\boldsymbol{I})\boldsymbol{T}_{cg} = \boldsymbol{R}_{cg}\boldsymbol{T}_{cij}-\boldsymbol{T}_{gij}$$

But please note that in this case, $\boldsymbol{T}_{cij}$ is determined only to an undetermined scale. Now the only problem remained is to determine this unknown scale.

3.8 Conclusion

We have performed head-eye calibraion for the active vision system in our lab successfully. Beside's we have suggested a new hand-eye calibration technique which is developed on the basis of feature correspondences.

References

[1] R. Y. Tsai, R. K. Lenz,"A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robotics Automa.*, vol. 5, no. 3, June 1989, pp. 345-358.

[2] Song De Ma and Guo-Qing Wei, "A self-calibration technique for active vision system," *in Proc. Asian Conf. Comput. Vision*, 1993, pp.628-631.

[3] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," *IEEE Trans. Robotics and Automat.*, vol. 5, no. 1, pp. 16-29, 1989.

[4] Y.Aloimonos,I.Weiss,and A. Bandopadhay, "Active vision", *Int. J. Comput. Vision* 2(1), 1988, 333-356

[5] R. Y. Tsai, "A versatile camera calibration techinque for high-accuracy 3D machine vision metrology using off-the-sehlf TV cameras and lenses," *IEEE Trans. Robotics Automn.* vol. 3, no. 4, pp. 323-344, August, 1987.

[6] T. Viéville,"Auto-calibration of visual sensor parameters on a robotic head," *Image and Vision Computing*, vol. 12, no. 4, pp. 227-237, May, 1994.

[7] G. Q. Wei, S. D. Ma, "A complete two-plane camera calibration method and experimental comparisons," *Proc. 4th Intern. Conf. Comput. Vision, Berlin,* May 11-14, 1993, pp. 133-138.

[8] R. Horaud, F. Dornaika, B. Boufama, and R.Mohr, "Self calibration of a stereo head mounted onto a robot arm," *Lecture Notes in Computer Science,* vol. 800. Jan-Olof Eklundh (Ed.), Computer Vision - ECCV'94, Springer-Verlag Berlin-Heidelberg 1994.

[9] J. Denavit, R.S. Hartenberg, "A kinematic notation for lower-pair machanisms based on matrices," *ASME Journal of Applied Mechanics*, June 1955, pp. 215-221.

[10] Robert M. Haralick, Hyonam Joo, "Pose Estimation from corresponding point data," *IEEE Trans. Systems, Man and Cybernetics*, vol 19, no. 6, November / December 1989. pp. 1426-1445.

Appendix I Procedures

Ext.run

```
../../../camera/Tsai/n_cal_fo near_r_1.tsai > near_r_1.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_2.tsai > near_r_2.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_3.tsai > near_r_3.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_4.tsai > near_r_4.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_5.tsai > near_r_5.pose
../../../camera/Tsai/ep_cal near_r_1.pose near_r_6.tsai > near_r_6.pose
exit
```

Poser.m

```
First_file = '?'
keyboard
head_end = max(findstr(First_file,'_'));
file_head = First_file(1:head_end);
tail_head = min(findstr(First_file,'.'));
file_tail = First_file(tail_head:length(First_file));
file_number = First_file(head_end+1:tail_head-1);

kk_0 = str2num(file_number);
kk = kk_0;
while 1
Inname = [file_head int2str(kk) file_tail];
 if ~exist(Inname), break, end
fid = fopen(Inname,'r');
[file_string,count] = fscanf(fid,'%s',[1,inf]);
fclose(fid);

if findstr(file_string,'Ext') == [],
number_string =  file_string;
else
number_string = file_string(30:length(file_string));
end
segment_index = findstr(number_string,'.');
for i = 1:length(segment_index)
if  (segment_index(i) == 2)
CPCC(i)  = str2num(number_string(segment_index(i)-1:segment_index(i)+14));
elseif  number_string(segment_index(i)-2) ~= '-',
CPCC(i)  =
 str2num(number_string(segment_index(i)-1:segment_index(i)+14));
else
CPCC(i)  = str2num(number_string(segment_index(i)-2:segment_index(i)+14));
end;
end;
Cx = CPCC(7);
Cy = CPCC(8);
Tci =[CPCC(12)
     CPCC(13)
     CPCC(14)];

Rx = CPCC(15);
Ry = CPCC(16);
Rz = CPCC(17);

f = CPCC(10);

Rci = [cos(Rz)*cos(Ry) cos(Rz)*sin(Ry)*sin(Rx)-sin(Rz)*cos(Rx) cos(Rz)*sin(Ry)*cos(Rx)+sin(Rz)*sin(Rx)
     sin(Rz)*cos(Ry)  sin(Rz)*sin(Ry)*sin(Rx)+cos(Rz)*cos(Rx)   sin(Rz)*sin(Ry)*cos(Rx)-cos(Rz)*sin(Rx)
     -sin(Ry)        cos(Ry)*sin(Rx)                          cos(Ry)*cos(Rx)];

Hci_sub = [ Rci,Tci
       zeros(1,3),1];
if kk == kk_0,
Hci = Hci_sub;
else
Hci = [Hci
      Hci_sub];
```

```
end
clear A;
kk = kk+1;
end;
fid = fopen('camera_pose.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',Hci');
fclose(fid);
```

revised_camera_motion.m

```
fid = fopen('camera_pose.dat','r');
[A,count] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
Extrinsic_Parameter = A';
number_of_positions = count/(4.*4);

fid = fopen('Cij.dat','w');
for position_index = 2:number_of_positions,


Ci = Extrinsic_Parameter((position_index-2)*4+1:(position_index-2)*4+4,:);

Cj = Extrinsic_Parameter((position_index-1)*4+1:(position_index-1)*4+4,:);
Relative_Camera_Motion = Cj*inv(Ci);
if(position_index == 2)
 Cij= Relative_Camera_Motion;
 else
 Cij = [Cij
     Relative_Camera_Motion];
 position_pairs = position_pairs +1;
end;
end;
fid = fopen('Cij.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',Cij');
fclose(fid);
```

revised_robot_motion.m

```
%in this version, we change the sign of joint varibles of Y(2) and E(7)
link_constant = [  0.0     0.0     0.0    -pi/2
              -pi/2     0.0     0.0    -pi/2
                0.0     0.0   114.0      0.0
              -pi/2   200.0     0.0    -pi/2
              -pi/2     0.0    72.0      0.0
             ];

fid = fopen('robot_joint_variable.dat','r');
[A,count] = fscanf(fid,'%f %f %f %f %f %f %f',[7,inf]);
fclose(fid);
JointVariable = A';
number_of_positions = count/7;
link_parameter = link_constant;
for position_index = 1:number_of_positions,
link_parameter(1,2) = link_constant(1,2) - JointVariable(position_index,1);
link_parameter(2,2) = link_constant(2,2) - JointVariable(position_index,2);
link_parameter(3,1) = link_constant(3,1) - JointVariable(position_index,3)/180*pi;
link_parameter(4,1) = link_constant(4,1) - JointVariable(position_index,6)/180*pi;
link_parameter(5,1) = link_constant(5,1) + JointVariable(position_index,7)/180*pi;

T5 = kinematic_equation(link_parameter)
if position_index == 1
T_output = T5;
else
T_output = [T_output
      T5
      ];
end;
end; %loop of position_index
postition_pairs = 0;
for position_index = 2:number_of_positions,
```

```
Gj = T_output((position_index-1)*4+1:(position_index-1)*4+4,:);
Gj_inv = inv(Gj);
Gi = T_output((position_index-2)*4+1:(position_index-2)*4+4,:);
Hand_Motion = Gj_inv*Gi;

if(position_index == 2)
 Gij= Hand_Motion;
 else
 Gij = [Gij
      Hand_Motion];
 position_pairs = position_pairs +1;
end;
end;
fid = fopen('Gij.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',Gij');
fclose(fid);
fid = fopen('robot_motion.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',T_output');
fclose(fid);
```

multi_tsai.m

```
fid = fopen('Gij.dat','r');
[Trans_Gij,count_G] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
fid = fopen('Cij.dat','r');
[Trans_Cij,count_C] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
Gij = Trans_Gij';
Cij = Trans_Cij';
count = count_G;
if count_C < count_G
 count = count_C;
end;
number_of_r_c_motion_pairs = count/16;
for pair_index = 1:number_of_r_c_motion_pairs-1,
Rgij = Gij((pair_index-1)*4+1:(pair_index-1)*4+3,1:3);

Rcij = Cij((pair_index-1)*4+1:(pair_index-1)*4+3,1:3);

Rgjk = Gij(pair_index*4+1:pair_index*4+3,1:3);

Rcjk = Cij(pair_index*4+1:pair_index*4+3,1:3);

[V1,D1] = eig(Rgij);
for I = 1:3,
 if norm(D1(I,I) -1.0) < 0.001
unity_index = I;
 break;
 end;
end;
Pgij = 2*sin(angle(Rgij)/2)*V1(:,unity_index);
Pgij = direction(Pgij,Rgij);
[V2,D2] = eig(Rgjk);
for I = 1:3,
 if norm(D2(I,I) -1.0) < 0.001
unity_index_gjk = I;
break;
 end;
end
Pgjk =  2*sin(angle(Rgjk)/2)*V2(:,unity_index_gjk);
Pgjk = direction(Pgjk,Rgjk);
clear V1
clear V2
clear D1
clear D2
[V1,D1] = eig(Rcij);
for I = 1:3,
 if norm(D1(I,I) -1.0) < 0.001
unity_index_cij = I;
```

```
break;
 end;
end;
Pcij =  2*sin(angle(Rcij)/2)*V1(:,unity_index_cij);
Pcij = direction(Pcij,Rcij);
[V2,D2] = eig(Rcjk);
for I = 1:3,
  if norm(D2(I,I) -1.0) < 0.001
unity_index_cjk = I;
break;
 end;
end
Pcjk = 2*sin(angle(Rcjk)/2)*V2(:,unity_index_cjk);
Pcjk = direction(Pcjk,Rcjk);
clear V1
clear V2
clear D1
clear D2
A1 = skew(Pgij+Pcij);
A2 = skew(Pgjk+Pcjk);
A_sub = [A1;
      A2];
if pair_index == 1,
A = A_sub;
else
A = [A
    A_sub];
end

B1 = Pcij-Pgij;
B2 = Pcjk-Pgjk;
B_sub =[ B1;
    B2];
if pair_index == 1,
B = B_sub;
else
B = [B
    B_sub];
end
end;

Pcg_1 = pinv(A) * B;

%thita_R_gc  = 2 * atan(norm(Pcg_1));

Pcg = (2/sqrt(1+ Pcg_1'*Pcg_1))*Pcg_1;

Rcg = (1- Pcg'*Pcg/2)*eye(3) + 0.5*(Pcg*Pcg'+sqrt(4 - Pcg'*Pcg) *skew(Pcg));

for pair_index = 1:number_of_r_c_motion_pairs-1,
Rgij = Gij((pair_index-1)*4+1:(pair_index-1)*4+3,1:3);
Tgij = Gij((pair_index-1)*4+1:(pair_index-1)*4+3,4);
Rcij = Cij((pair_index-1)*4+1:(pair_index-1)*4+3,1:3);
Tcij = Cij((pair_index-1)*4+1:(pair_index-1)*4+3,4);
Rgjk = Gij(pair_index*4+1:pair_index*4+3,1:3);
Tgjk = Gij(pair_index*4+1:pair_index*4+3,4);
Rcjk = Cij(pair_index*4+1:pair_index*4+3,1:3);
Tcjk = Cij(pair_index*4+1:pair_index*4+3,4);
D1 = Rgij - eye(3);
D2 = Rgjk - eye(3);

D_sub = [D1;
    D2];

if pair_index == 1,
D = D_sub;
else
D = [D
    D_sub];
end
```

```
E1 = Rcg*Tcij -Tgij;
E2 = Rcg*Tcjk -Tgjk;

E_sub = [ E1
    E2];

if pair_index == 1,
E = E_sub;
else
E = [E
    E_sub];
end

end;
Tcg = pinv(D)*E;
HandEye = [Rcg,Tcg
            zeros(1,3) , 1];
fid = fopen('Hcg.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',HandEye');
fclose(fid);
```

verify_tsai.m

```
fid = fopen('camera_pose.dat','r');
[A,count] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
Camera_Pose = A';
Number_of_Stations = count/16;
fid = fopen('robot_motion.dat','r');
[A,count] = fscanf(fid,'%f %f %f %f \n',[4,inf]);
fclose(fid);
Robot_Attitude = A'

fid = fopen('Hcg.dat','r');
[A,count] = fscanf(fid,'%f %f %f %f \n',[4,inf]);
fclose(fid);

HandEye = A';

fid = fopen('Gij.dat','r');
[Trans_Gij,count_G] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
fid = fopen('Cij.dat','r');
[Trans_Cij,count_C] = fscanf(fid,'%f %f %f %f ',[4,inf]);
fclose(fid);
Gij = Trans_Gij';
Cij = Trans_Cij';
count = count_G;
if count_C < count_G
 count = count_C;
end;
number_of_r_c_motion_pairs = count/16;

for station_index = 1:Number_of_Stations,
Ci = Camera_Pose((station_index-1)*4+1:(station_index-1)*4+4,:);
Gi = Robot_Attitude((station_index-1)*4+1:(station_index-1)*4+4,:);

HRC_sub = inv(Ci)*inv(HandEye)*inv(Gi);
if station_index == 1,
HRC = HRC_sub;
Acc_HRC_sub = HRC_sub;
else
HRC = [HRC;HRC_sub];
Acc_HRC_sub = Acc_HRC_sub + HRC_sub;
end
end;
AV_HRC = Acc_HRC_sub / Number_of_Stations;
%The following paragraph is to predictating the camera pose on the basis of the Hcg just obtained.
```

```
for station_index = 1:Number_of_Stations,
%Referred to the robot base world coordinate frame
Ci = Camera_Pose((station_index-1)*4+1:(station_index-1)*4+4,:);
Gi = Robot_Attitude((station_index-1)*4+1:(station_index-1)*4+4,:);


Predicted_Pose = inv(HandEye)*inv(Gi);
Measured_Pose = Ci*AV_HRC;
Pe_sub = Measured_Pose - Predicted_Pose;
if station_index == 1,
Pose_Error = Pe_sub;
else
Pose_Error = [Pose_Error;Pe_sub];
end

end;

fid = fopen('Pose_Error.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f \n',Pose_Error');
fclose(fid);
```

angle.m

```
function [thita] = angle(R_M)

thita = atan(sqrt((R_M(3,2)-R_M(2,3))^2
+(R_M(1,3)-R_M(3,1))^2+(R_M(2,1)-R_M(1,2))^2)/(R_M(1,1)+R_M(2,2)+R_M(3,3)-1));
if thita < 0
thita = pi+thita;
end;
```

direction.m

```
function [v] = direction(x,R_M)
[Maximum_value,Max_ID] =max(abs(x));
sign_eig = sign(x(Max_ID));
sign_flag = 0;
if Max_ID == 1
  if sign(R_M(3,2) - R_M(2,3)) ~= sign_eig
     sign_flag = 1;
  end
elseif Max_ID == 2
  if sign(R_M(1,3) - R_M(3,1)) ~= sign_eig
     sign_flag =1;
  end
else
  if sign(R_M(2,1) - R_M(1,2)) ~= sign_eig
     sign_flag =1;
  end;
end;

if sign_flag == 0
v = x;
else
v = -x;
end;
```

skew.m

```
function [v] = skew(x)

v(1, 1) = 0.0;
v(2,2) = 0.0;
v(3,3) = 0.0;
v(1,2) = - x(3);
v(1,3) = x(2);
v(2,1) = x(3);
v(2,3) = -x(1);
v(3,1) = -x(2);
v(3,2) = x(1);
```

kinematic_equation.m

```
function [v1] = kinematic_equation(x1)

A1 = homo_transformation(x1(1,:));

A2 = homo_transformation(x1(2,:));

A3 = homo_transformation(x1(3,:));

A4 = homo_transformation(x1(4,:));

A5 = homo_transformation(x1(5,:));

v1 = A1*A2*A3*A4*A5;
```

homo_transformation.m

```
function [v] = homo_transformation(x)
thita = x(1);
d = x(2);
a = x(3);
alpha = x(4);

v = [cos(thita)  -cos(alpha)*sin(thita)  sin(alpha)*sin(thita) a*cos(thita)
    sin(thita)   cos(alpha)*cos(thita) -sin(alpha)*cos(thita) a*sin(thita)
    0.0          sin(alpha)            cos(alpha)             d
    0.0          0.0                   0.0                    1    ];
```

rt_joint.m

```
First_file = '?'
keyboard
head_end = max(findstr(First_file,'_'));
file_head = First_file(1:head_end);
tail_head = min(findstr(First_file,'.'));
file_tail = First_file(tail_head:length(First_file));
file_number = First_file(head_end+1:tail_head-1);

kk_0 = str2num(file_number);
kk = kk_0;
while 1
Inname = [file_head int2str(kk) file_tail];
 if ~exist(Inname), break, end
fid = fopen(Inname,'r');
[A,count] = fscanf(fid,'%f %f %f %f %f %f %f',[1,inf]);
fclose(fid);
Robot_Joint_Sub = A
if kk == kk_0,
 Robot_Joint_Variable = Robot_Joint_Sub;
else
Robot_Joint_Variable = [Robot_Joint_Variable
Robot_Joint_Sub];
end;
kk = kk+1;
end;

fid = fopen('robot_joint_variable1.dat','w');
fprintf(fid,'%12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f\n',Robot_Joint_Variable');
fclose(fid);
```

Chapter 4

A New Tracking Method for Shape from Motion Using an Active Vision System

4.1 Introduction

Shape from motion is one of the fundamental problem of computer vision. Different approaches to the solution of this problem had been reported in the literature [2,3,4]. Particularly, Tomasi and Kanade [2] proposed a simple and robust factorization method by Singular Value Decomposition (SVD). Orthographic projection is assumed. Poelman and Kanade [3] extended this method to para-perspective model for wider range applications.

These two methods require the tracking of the R image points through F image frames. The feature tracking method used by Tomasi and Kanade [2] was developed on the basis of the similarity of image. This assumption requires an optimisation process and demands small inter-frame motions. However, the tracking of several hundred features through an image sequence is computationally expensive and time consuming. Also, the possibility of making error is high. A more satisfactory tracking method is required.

In this chapter, we propose a fast and reliable method for tracking feature points through a sequence of image frames. The details of our approach is described in section 2. In section 3, we describe the experiments of using our feature tracking method for shape recovery by the SVD factorization technique of [2]. Some discussions on the new method of feature tracking and our ideas on future research will be made in section 4.

4.2 A New Tracking Method

4.2.1 Our Approach

At first, a small set of salient feature points (projective basis) is extracted and tracked through an image sequence using conventional techniques. Then, the remaining feature points can be tracked efficiently using the concept of projective invariants. As a consequence, only simple algebraic computation is required. Conventional correspondence computation is unnecessary. Therefore, the efficiency and accuracy of the tracking stage can be much improved.

By introducing projective invariants into the tracking stage, our approach takes advantage of the robustness and simplicity of the method of Tomasi and Kanade [2]. As an advantage over the tracking method of [2], it also can accommodate the cases where larger motion occurs between adjacent frames.

In this chapter, we restrict the type of object under consideration to be polyhedra so that the theory of planar projective geometry can be applied. In contrast with the projective invariants to

the counterpart in 3D projective geometry, the planar projective invariants are rather simple to compute and less sensitive to noise.

4.2.2 Using an Active System to Track the Projective Basis Across Image Sequence

The image sequence under consideration is acquired using an active vision system in our laboratory.This active vision system has two linear axes which can provide translational motion within a horizontal plane. It has one common rotary axis around which stereo cameras can pan from side to side, one independent vergence axis and one independent tilt axis for each camera. Apart from its flexibility, the active vision system has very high accuracy and repeatability.The schematic configuration is shown in Fig.1.1.

Once an image sequence of a polyhedra is acquired, salient feature points corresponding to high curvature points are extracted from each frame of the sequences. In our experiments, the test object is a calibration block in the form of a cube (refer to section 3 for the details). For the first frame of the image sequence, four corner points of each face are chosen as projective basis. The question now is how to find the correspondences of these points across the image sequences.

We tackle this problem in the following way. Calibrating the head-eye geometry of our active vision system enables us to determine the corresponding positions in the subsequent frames in the image sequences. First, the new camera pose are predicted with the knowledge of head-eye geometry and relative motion of robot end effector with respect to the robot base co-ordinate system. Fundamental matrix and epipolar lines of each of the point of projective basis can be obtained readily. The best match of the projective basis can be found using the conventional stereo matching technique.

4.2.3 Using Projective Invariants to Track the Remaining Feature Points

Once the projective basis is defined for each planar surface, we can track the remaining feature points on the particular surface using the image transfer method developed by Mohr *et. al*[1]. The following is an abstract from [1].

It was pointed out by Efimov[5] that cross-ratio is the basic invariant in projective geometry and all other projective invariants can be derived from it. Consider four collinear points A, B, C, D, their cross ratio can be defined as:

$$[A, B, C, D] = \frac{\overline{CA}}{\overline{CB}} \times \frac{\overline{DB}}{\overline{DA}} \qquad (4.1)$$

where $\overline{CA}$ is the algebraic measure of CA.

It is noted that any homography preserves the cross-ratio. A homographic transform is a linear transformation in homogeneous coordinates. We define the cross-ratio of a pencil of 4 lines l_1,

l_2, l_3, l_4 going through a point O as the cross-ratio[A, B, C, D] of the points of intersection of the line l_k with any line l not going through O. Figure 4.1 illustrates the cross-ratio of 4 lines.

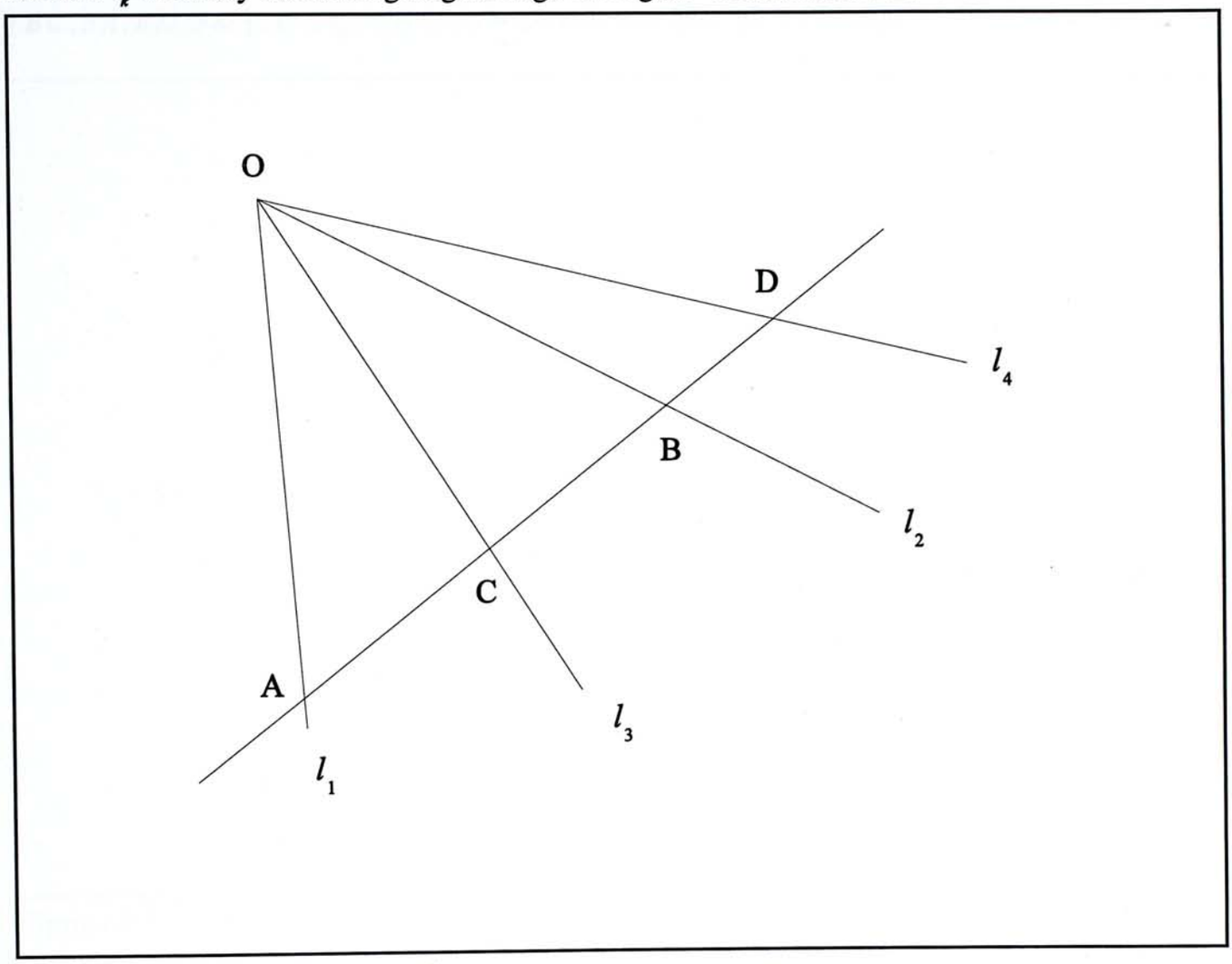


Figure 4.1 The cross-ratio of 4 lines

According to the concepts of planar project geometry, any four points A, B ,C, D in projective plane define a projective co-ordinate system if no three of them are collinear (see Figure 4.2). Given a point M of projective plane P, the projective co-ordinates (x_1, x_2, x_3) of M in the projective basis (A, B, C, D) are a triple of real numbers. They can be determined in term of cross ratios.

$$k_1 = \frac{x_1}{x_2} = [CA, CB, CD, CM]$$

$$k_2 = \frac{x_2}{x_3} = [AB, AC, AD, AM] \qquad (4.2)$$

$$k_3 = \frac{x_3}{x_1} = [BC, BA, BD, BM]$$

It should be observed from Figure 4.2 that $[CA,CB,CD,CM]$, $[AB,AC,AD,AM]$, and $[BC,BA,BD,BM]$ corresponds to three pencils of 4 lines going through C, A, and B respectively.

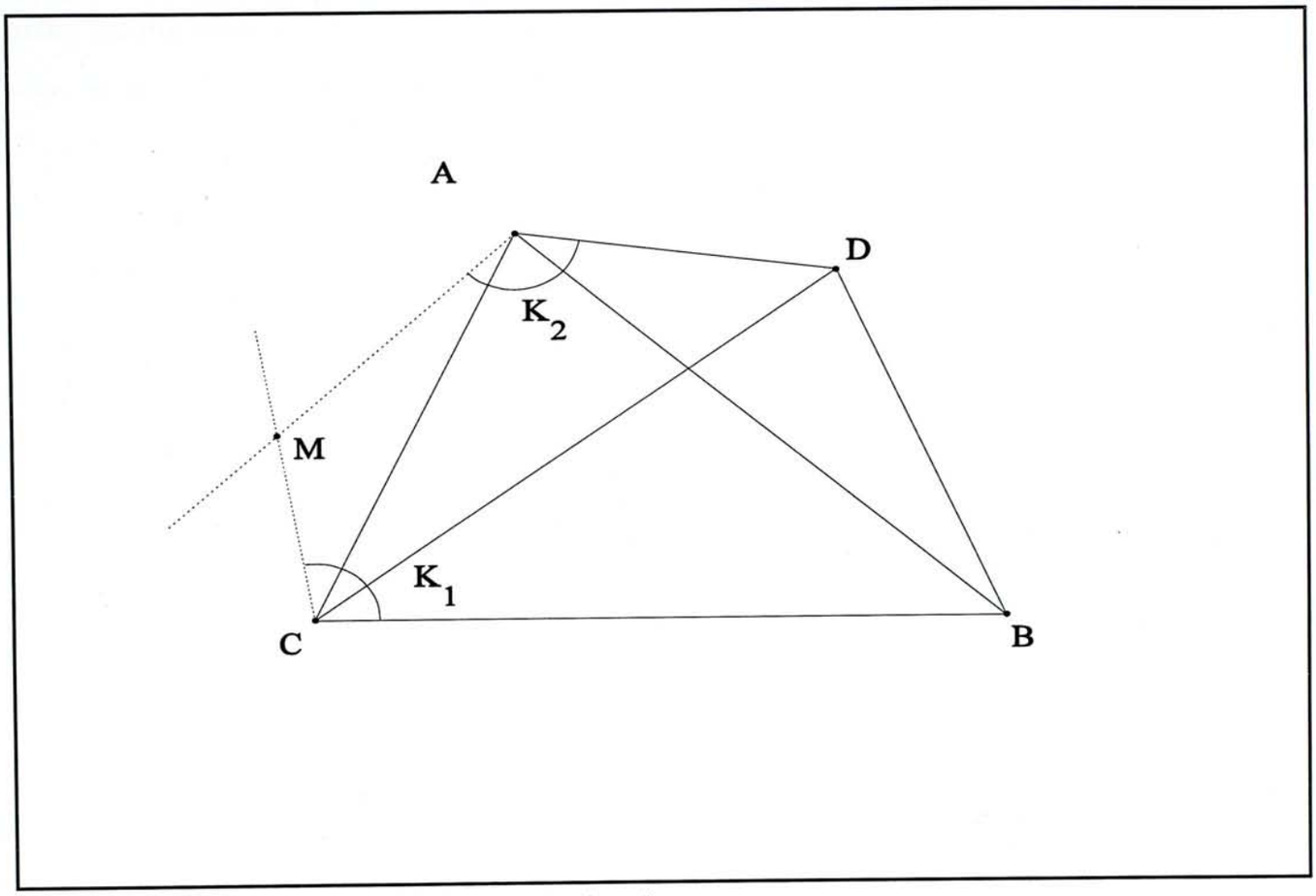


Figure 4.2 Projective coordinates in the plane

If the point M is not lying on AC, only two cross-ratio k_1 and k_2 are necessary to uniquely define a point. Given the projective basis (4 points) on a projective plane, say the reference projective plane, the projective co-ordinates of each point on the reference projective plane can be computed from the Cartesian co-ordinates. The projective co-ordinates will remain unchanged after any projective transformations. Assume that there is a target projective plane related to the reference projective plane by a projective transformation. One can always determine the Cartesian position of any point designated by a particular projective co-ordinates on the target projective plane if the Cartesian co-ordinates of the projective basis set on the target projective plane are given. In other words, one can map any point in a reference projective plane to a target projective plane if the Cartesian coordinates of the projective basis are known.

Generally speaking, any two frames of an image sequence are related by a particular projective transformation since both rigid motion and perspective projection can be subsumed into the framework of projective projection.

In our implementation, first frame of the image sequence was selected as the reference frame.

Any subsequent frame was treated as a target frame. A small set of salient features, forming a projective basis, was tracked through the image sequence as projective basis, We then computed the projective co-ordinates for the remaining feature points on the reference frame(see Figure 4.3). The image co-ordinates of these feature points on a particular target frame can be computed using the the method of [1]. The method is summerized as follows:

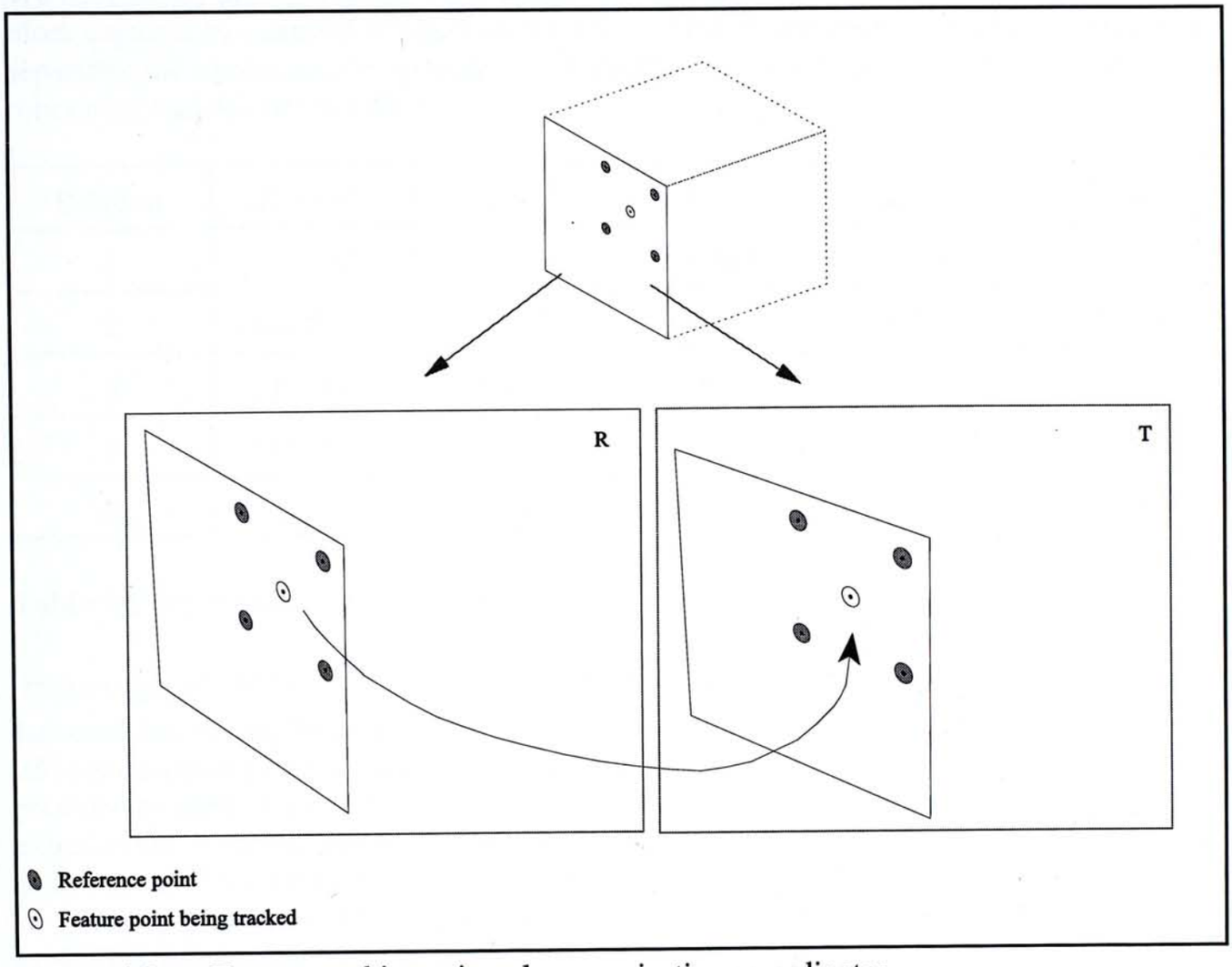


Figure 4.3 Feature tracking using planar projective co-ordinates.

1. Four arbitrary points are chosen as the projective basis in the reference frame. Then, the correspondences between the projective basis points across the image sequence are determined by conventional method.

2. For each particular feature point to be tracked, calculate the its projective coordinate.

3. With the knowledge of the projective coordinate for each particular feature points, we can always find the image position of this particular feature point on any frame within the image sequence as long as the projective basis has been determined previously.

4. Several degenerate case are discussed, when the feature points fall on the several particular straight lines.

5. Once the position of a feature point in the new frame is determined, the best match of the image of the feature point can be obtained by searching around a small neighborhood which is centered at the transferred position of the feature point.

We have tested this method on a real image sequence (ar_1.img .... ar_5.img) of a calibration block captured by a camera mounted on the active vision system mentioned above. The distance separating the camera and object is about 1.5 metres. The focal length of the lens is 16mm. The robot joint variable are as follows:

Position	*X* (mm)	*Y* (mm)	*D* (°)	*E* (°)	Frame
1	0.000	0.000	0.000	0.000	ar_1.img
2	-140.000	-98.000	10.000	0.000	ar_2.img
3	-140.000	-98.000	10.000	6.500	ar_3.img
4	140.000	-98.000	-8.000	5.000	ar_4.img
5	0.000	-98.000	0.000	7.000	ar_5.img

Table 4.1 Input values and corresponding image frames

Five images of a planar 5x5 grid lying on the surface π_1 of the calibration block (see Figure 4.9)are used for testing the performance of the algorithm. Four corner points of the grid (1, 5, 21, 25) are chosen as projective basis. And the correspondences of the projective basis points set are provided by hand. The first frame of the image sequence was taken as the reference image. The experimental results of tracking errors on the other four frames are shown in Table 4.2. The distributions of tracking errors across all the feature points on four target frames are illustrated in Chart 4.4 - Chart 4.7 respectively. From these results, we can conclude that this projective_invairants based tracker can yields good performance.

	ar_2.err (pixel)	ar_3.err (pixel)	ar_4.err (pixel)	ar_5.err (pixel)
1	0.000000	0.000000	0.000000	0.000000
2	0.142013	0.187018	0.051808	0.191136
3	0.11351	0.187718	0.004501	0.124423
4	0.272023	0.153914	0.14392	0.197137
5	0.000000	0.000000	0.000000	0.000000
6	0.059691	0.135189	0.12525	0.145593
7	0.126007	0.146353	0.05188	0.131061
8	0.145749	0.240188	0.034058	0.099644
9	0.200685	0.159888	0.154645	0.140845
10	0.095905	0.095300	0.170202	0.014900
11	0.038599	0.070723	0.102378	0.010735
12	0.123796	0.100990	0.089280	0.013706
13	0.173837	0.292396	0.090420	0.053888
14	0.109604	0.156947	0.112645	0.102861
15	0.005100	0.093800	0.050501	0.087901
16	0.067676	0.013787	0.079196	0.018430
17	0.121773	0.049889	0.060409	0.098133
18	0.203492	0.341198	0.061001	0.078802
19	0.035963	0.155811	0.081639	0.129678
20	0.001900	0.066300	0.097001	0.176702
21	0.000000	0.000000	0.000000	0.000000
22	0.089709	0.003600	0.086017	0.176048
23	0.221625	0.391961	0.078816	0.035709
24	0.044506	0.149023	0.039910	0.019804
25	0.000000	0.000000	0.000000	0.000000

Table 4.2 Tracking errors of 25 feature points on four target frames ar2.img ...ar5.img.

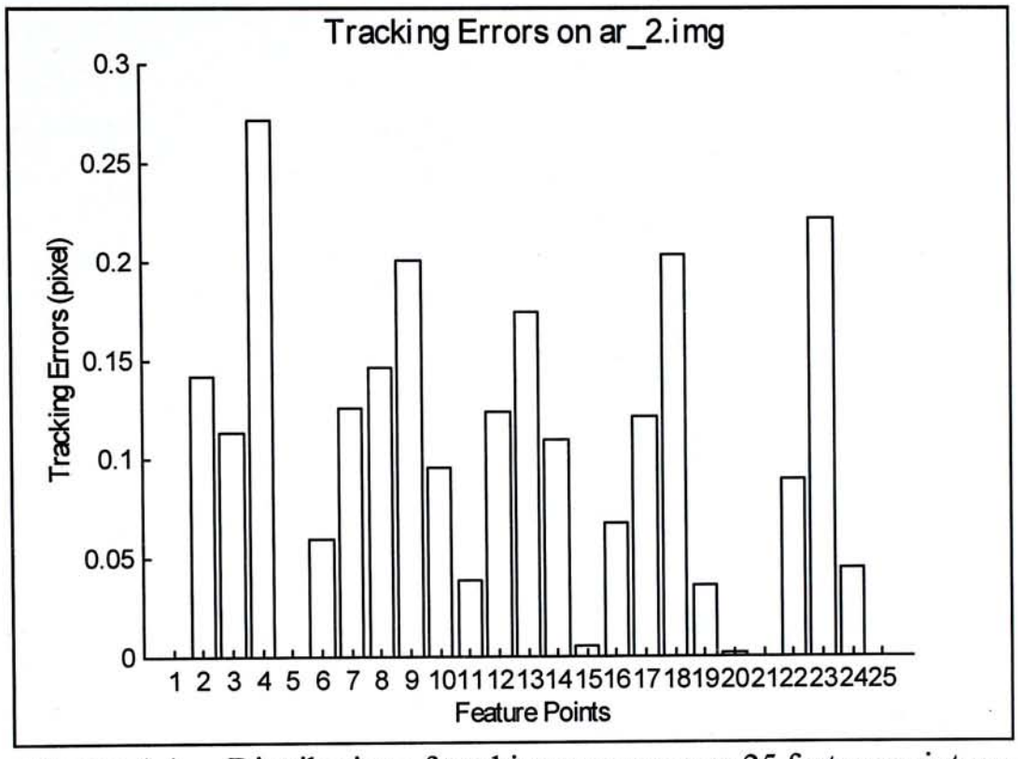


Figure 4.4 Distribution of tracking errors across 25 feature points on the targe frame ar_2.img.

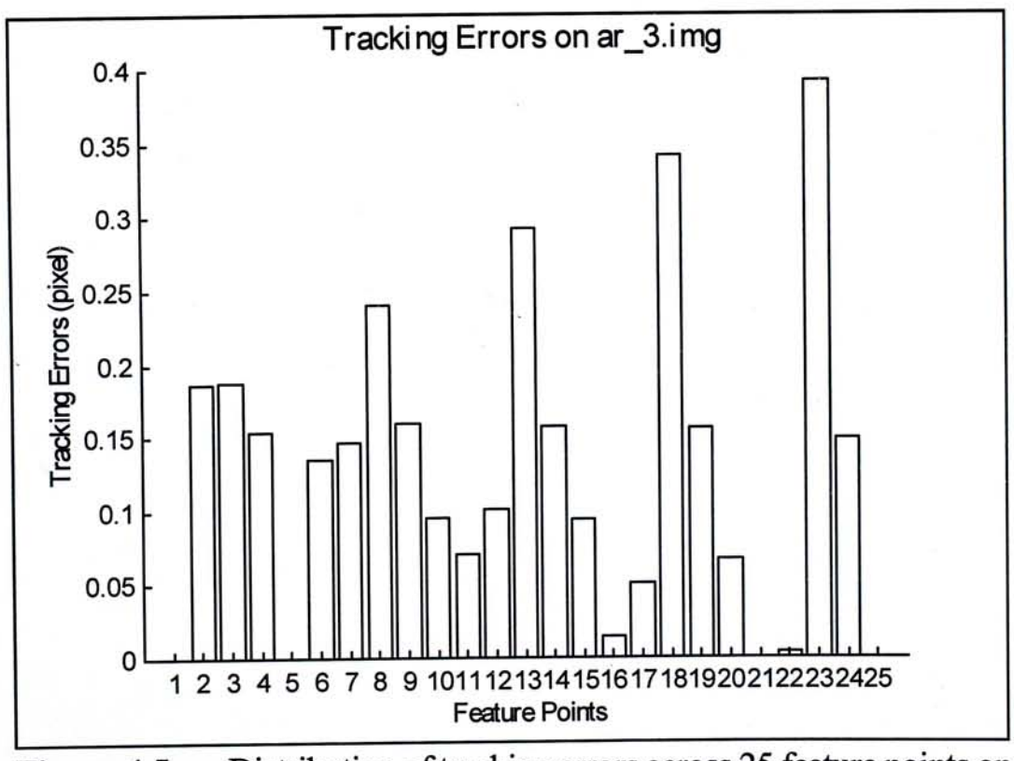


Figure 4.5 Distribution of tracking errors across 25 feature points on the targe frame ar_3.img.

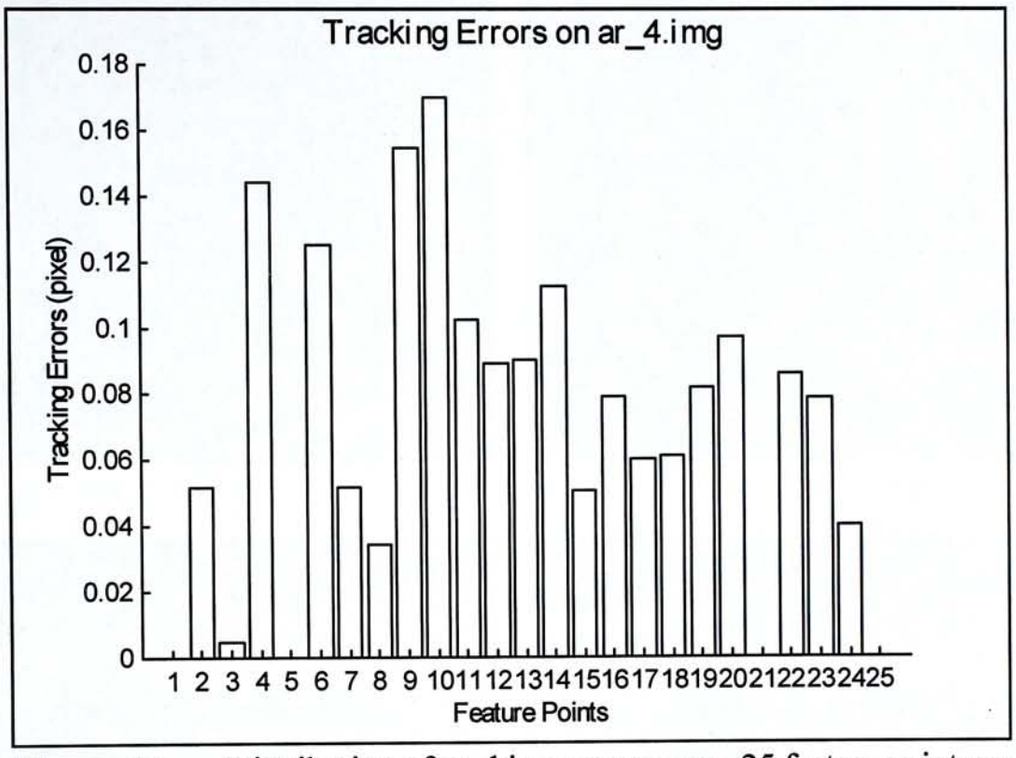


Figure 4.6 Distribution of tracking errors across 25 feature points on the targe frame ar_4.img.

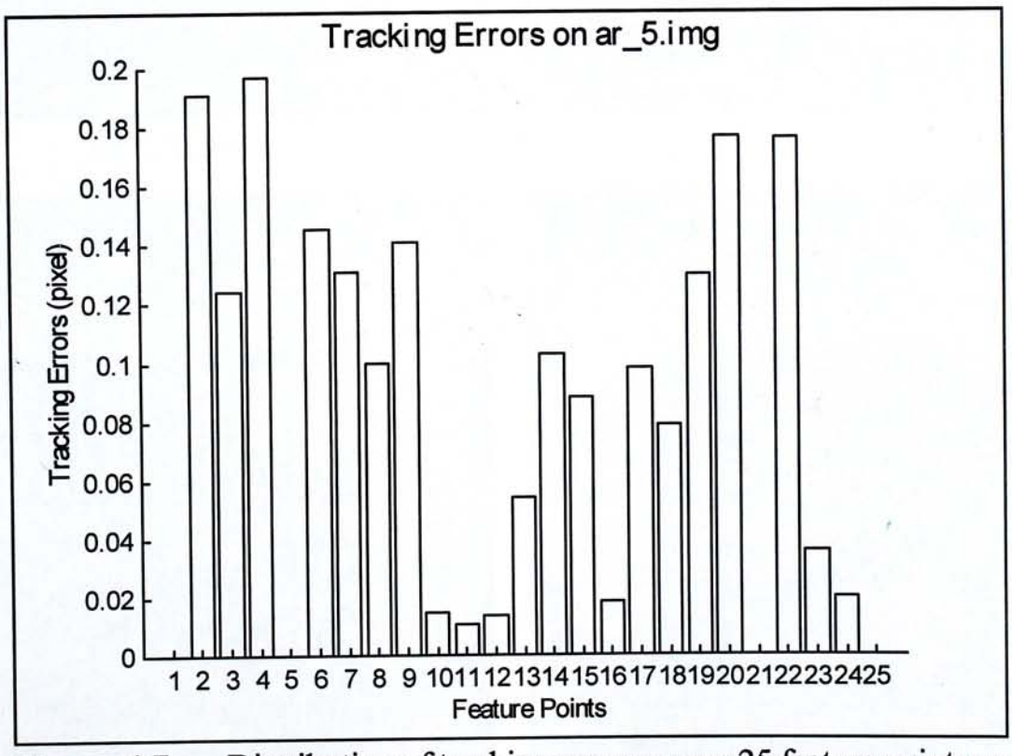


Figure 4.7 Distribution of tracking errors across 25 feature points on the targe frame ar_5.img.

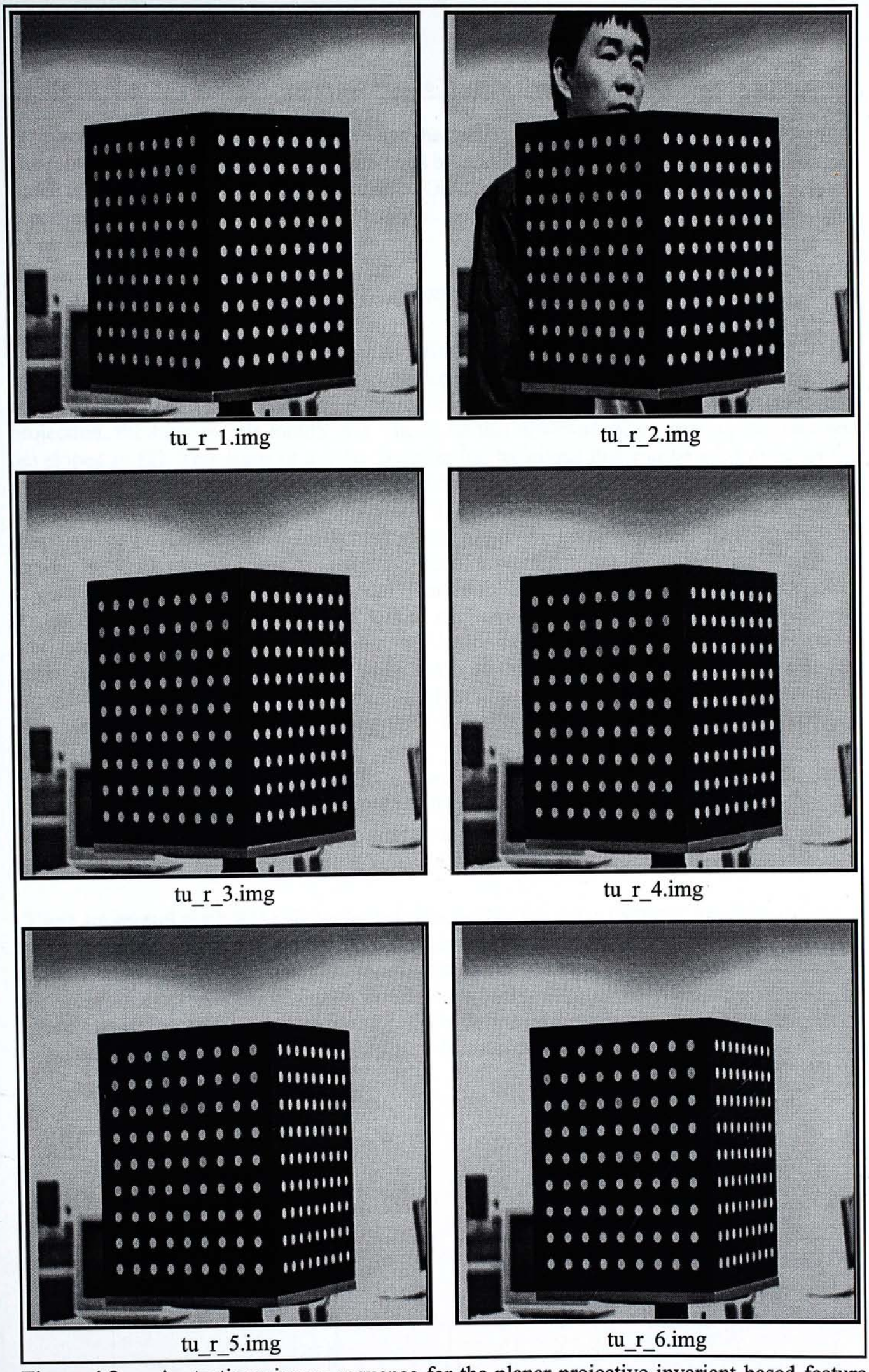


Figure 4.8 An testinng image sequence for the planar projective invariant based feature tracking algorithm.

The experimental result obtained shows that the maximum tracking error is less than 0.4 pixel. The robustness of this method can be improved by centering a searching window in the position which is predicted using projective invariants. Only pure transferred positions were used in our experiments. We shall introduce the random sample consensus technique to improve the robustness even further.

4.3 Using Factorisation Method to Recover Shape from Motion

The orthographic method of Tomasi and Kanade[2] was used for shape recovery in our experiments. In this method, an image sequence can be represented by a $2F \times R$ measurement matrix of the image coordinates of R points tracked through F frames. Under orthographic projection, the rank of this matrix is 3. Based on this observation, an ranking theorem was developed in [2]. This leads to a SVD factorization technique that can be used to factor the measurement matrix into a shape matrix and a motion matrix. The theorem makes no assumption on the kind of motion and the type of surface. Thus. it can be applied on an object of arbitrary shape. Using the tracking results yielded by this projective invariants based tracker, we try to recover the shape of two planar surface of a calibration block from an image sequence employing Tomasi and Kanade's factorisation method[2] under orthography. The image sequence (tu_1.img ,..., tu_6.img) is shown in Figure 4.8. The calibration reference block was an accurately fabricated black aluminium cube with a 9x9 lattice of white discs on each of the mutually orthogonal surfaces. The diameter of a white disc is 10.0mm. The Spacing separating the centres of the adjacent discs is 30.0mm. The positional accuracy is 0.1mm. The centroids of the white discs were chosen as feature points.(see Table 4.3 - 4.8). Figure 4.9 depicts the numbering of control points on the calibration block.
The image sequence consists of seven frames and only 25 points are considered for each of two planar surfaces π_1 and π_2. The reconstructed shape is shown in Figure 4.10. The reconstructed angle between two orthogonal surfaces is 87.03 degrees. There is an error of less than 3 degrees from the true value of 90 degrees.

There are several reasons to choose factorisation method for shape recovery. The first one is that the factorisation method can exploit the redundancy of the information in the image sequence to provide robustness of the estimation. A full solution can still be obtained if some of the features were missing in some of the images in the sequence due to occlusions and tracking failures. The second one is that the factorisation method allows the use of a small inter-frame camera motion to facilitate the tracking of the projective basis in an image sequence.

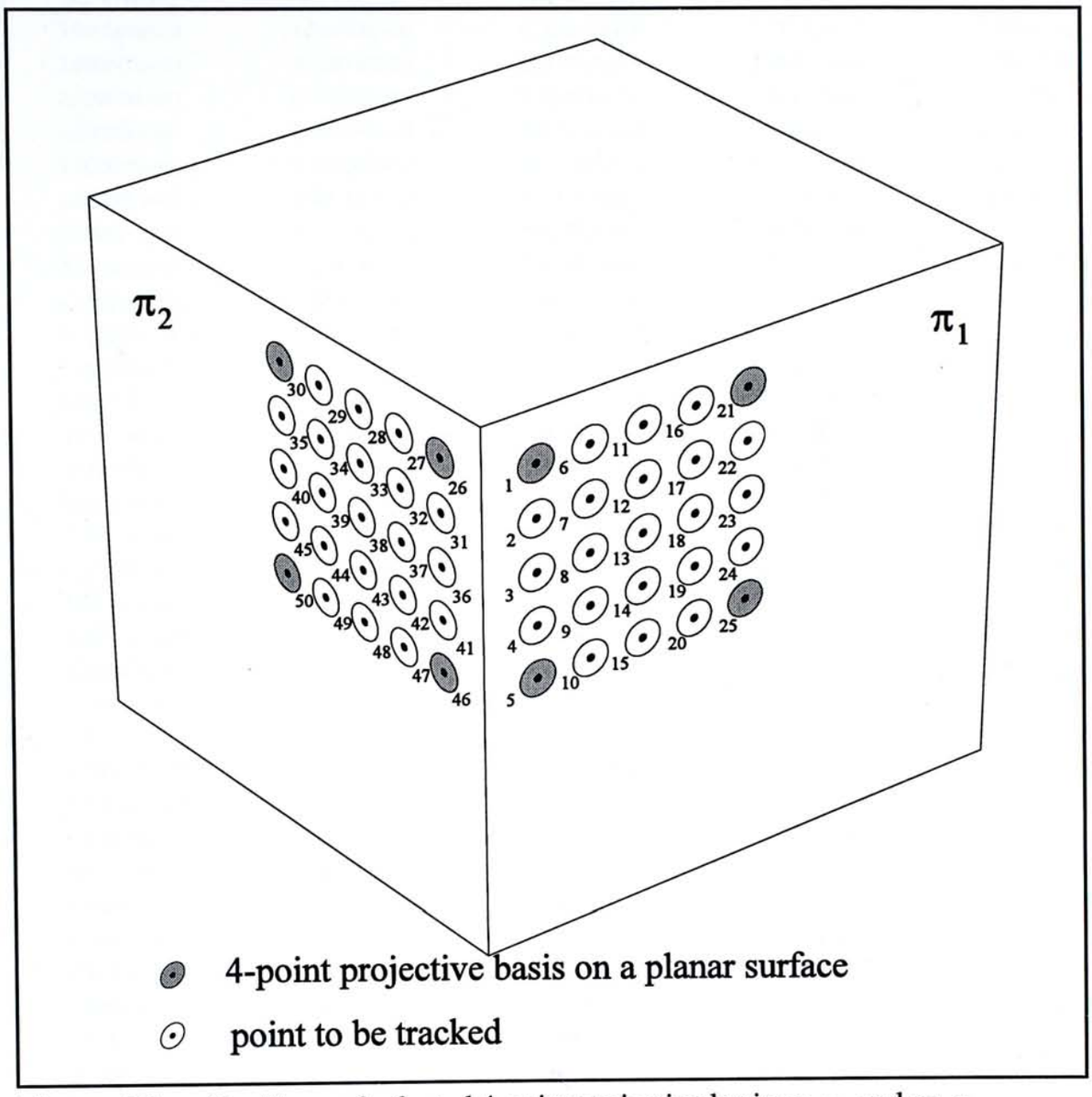


Figure 4.9 Positions of selected 4-point projective basis on π_1 and on π_2 on the image of a calibration block.

tu_1.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	2.617340e+02	1.584633e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	2.621039e+02	1.940170e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	2.624736e+02	2.295399e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	2.628431e+02	2.650520e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	2.632108e+02	3.003919e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	2.811683e+02	1.589831e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	2.815406e+02	1.942391e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	2.819129e+02	2.294954e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	2.822850e+02	2.647291e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	2.826558e+02	2.998474e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.002842e+02	1.594945e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.006669e+02	1.944576e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.010500e+02	2.294516e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.014326e+02	2.644110e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.018146e+02	2.993108e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.191393e+02	1.599988e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.195140e+02	1.946729e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.198894e+02	2.294085e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.202643e+02	2.640982e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.206392e+02	2.987836e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.377275e+02	1.604960e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.381078e+02	1.948853e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.384891e+02	2.293659e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.388699e+02	2.637891e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.392511e+02	2.982624e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.250616e+02	1.592765e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.086461e+02	1.607999e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	1.924951e+02	1.622987e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.766988e+02	1.637646e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.612466e+02	1.651986e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.254317e+02	1.946477e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.090285e+02	1.958202e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	1.928816e+02	1.969743e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.770760e+02	1.981040e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.615840e+02	1.992113e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.258034e+02	2.301619e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.094129e+02	2.310338e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	1.932707e+02	2.318925e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.774565e+02	2.327338e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.619247e+02	2.335600e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.261750e+02	2.656747e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.097971e+02	2.662147e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	1.936592e+02	2.667468e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.778359e+02	2.672685e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.622642e+02	2.677819e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.265473e+02	3.012497e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.101817e+02	3.014432e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	1.940480e+02	3.016340e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.782155e+02	3.018212e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.626037e+02	3.020058e+02

Table 4.3 Control points on the calibration block and detected image coordinates in the 1st frame

tu_2.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	2.796109e+02	1.581713e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	2.799862e+02	1.936103e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	2.803634e+02	2.292275e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	2.807406e+02	2.648392e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	2.811154e+02	3.002331e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	2.967865e+02	1.588115e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	2.972256e+02	1.939781e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	2.976662e+02	2.292606e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	2.981067e+02	2.645375e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	2.985451e+02	2.996412e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.138181e+02	1.594463e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.142533e+02	1.943415e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.146893e+02	2.292933e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.151252e+02	2.642419e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.155595e+02	2.990635e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.306081e+02	1.600722e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.310192e+02	1.946992e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.314303e+02	2.293254e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.318414e+02	2.639515e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.322515e+02	2.984967e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.470239e+02	1.606840e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.474592e+02	1.950500e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.478938e+02	2.293570e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.483283e+02	2.636652e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.487625e+02	2.979361e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.427645e+02	1.587453e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.240165e+02	1.601568e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.057103e+02	1.615351e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.877499e+02	1.628873e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.698827e+02	1.642325e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.431519e+02	1.942309e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.244617e+02	1.953848e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.061153e+02	1.965175e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.881228e+02	1.976283e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.702875e+02	1.987294e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.435397e+02	2.297637e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.249069e+02	2.306152e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.065198e+02	2.314555e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	1.884948e+02	2.322793e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.706906e+02	2.330930e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.439268e+02	2.652288e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.253520e+02	2.658348e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.069247e+02	2.664361e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	1.888677e+02	2.670252e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.710955e+02	2.676051e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.443155e+02	3.008353e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.257976e+02	3.010987e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.073291e+02	3.013615e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	1.892390e+02	3.016188e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.714975e+02	3.018712e+02

Table 4.4 Control points on the calibration block and detected image coordinates in the 2nd frame

tu_3.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	2.969454e+02	1.575098e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	2.973797e+02	1.930144e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	2.978140e+02	2.285200e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	2.982492e+02	2.641030e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	2.986808e+02	2.993896e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.118068e+02	1.582626e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.122604e+02	1.934665e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.127131e+02	2.286011e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.131671e+02	2.638390e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.136184e+02	2.988693e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.265558e+02	1.590097e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.269563e+02	1.939129e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.273552e+02	2.286809e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.277557e+02	2.635809e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.281548e+02	2.983629e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.409110e+02	1.597369e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.413101e+02	1.943490e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.417069e+02	2.287590e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.421055e+02	2.633270e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.425038e+02	2.978631e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.550372e+02	1.604525e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.554583e+02	1.947789e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.558761e+02	2.288362e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.562962e+02	2.630759e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.567169e+02	2.973680e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.605464e+02	1.578848e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.398918e+02	1.591952e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.195471e+02	1.604860e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	1.995199e+02	1.617567e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.796017e+02	1.630204e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.609630e+02	1.933457e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.403267e+02	1.944408e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.199497e+02	1.955223e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	1.999067e+02	1.965859e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.800113e+02	1.976418e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.613823e+02	2.290333e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.407636e+02	2.298503e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.203535e+02	2.306590e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	2.002938e+02	2.314538e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.804205e+02	2.322412e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.617993e+02	2.645251e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.411988e+02	2.651197e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.207563e+02	2.657097e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	2.006806e+02	2.662892e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.808301e+02	2.668621e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.622190e+02	3.002429e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.416357e+02	3.005291e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.211598e+02	3.008139e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	2.010671e+02	3.010933e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.812382e+02	3.013691e+02

Table 4.5 Control points on the calibration block and detected image coordinates in the 3rd frame

tu_4.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.100787e+02	1.574398e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.105833e+02	1.930087e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.110844e+02	2.283354e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.115880e+02	2.638398e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.120906e+02	2.992653e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.230130e+02	1.583175e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.235491e+02	1.935000e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.240822e+02	2.284876e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.246182e+02	2.636632e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.251525e+02	2.987267e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.359006e+02	1.591921e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.364003e+02	1.939869e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.368979e+02	2.286377e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.373984e+02	2.634899e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.378968e+02	2.982013e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.484943e+02	1.600467e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.489534e+02	1.944625e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.494112e+02	2.287842e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.498720e+02	2.633209e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.503304e+02	2.976886e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.608172e+02	1.608830e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.612626e+02	1.949289e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.617075e+02	2.289282e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.621553e+02	2.631544e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.626005e+02	2.971827e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.742928e+02	1.575808e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.524071e+02	1.588443e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.307567e+02	1.600942e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	2.092980e+02	1.613330e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.881698e+02	1.625527e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.748193e+02	1.930113e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.529075e+02	1.941015e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.312365e+02	1.951797e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	2.098064e+02	1.962459e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.886541e+02	1.972983e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.753516e+02	2.288368e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.534120e+02	2.296465e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.317188e+02	2.304471e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	2.103159e+02	2.312370e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.891381e+02	2.320185e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.758791e+02	2.643389e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.539131e+02	2.649561e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.321991e+02	2.655662e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	2.108246e+02	2.661667e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.896224e+02	2.667625e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.764081e+02	2.999323e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.544155e+02	3.003549e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.326806e+02	3.007724e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	2.113345e+02	3.011826e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	1.901078e+02	3.015904e+02

Table 4.6 Control points on the calibration block and detected image coordinates in the 4th frame

tu_5.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.238784e+02	1.574639e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.244090e+02	1.931179e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.249323e+02	2.282862e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.254639e+02	2.640090e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.259877e+02	2.992157e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.348106e+02	1.585192e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.352947e+02	1.937670e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.357730e+02	2.285892e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.362574e+02	2.638543e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.367364e+02	2.987304e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.454864e+02	1.595497e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.459743e+02	1.944037e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.464569e+02	2.288879e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.469442e+02	2.637013e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.474278e+02	2.982477e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.559247e+02	1.605573e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.563939e+02	1.950250e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.568588e+02	2.291786e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.573268e+02	2.635525e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.577927e+02	2.977797e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.660693e+02	1.615366e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.665568e+02	1.956309e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.670405e+02	2.294632e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.675258e+02	2.634064e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.680106e+02	2.973184e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	2.891457e+02	1.575581e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.661129e+02	1.587617e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.431990e+02	1.599591e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	2.205812e+02	1.611411e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	1.981221e+02	1.623147e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	2.896788e+02	1.931375e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.666169e+02	1.941433e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.437000e+02	1.951427e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	2.210782e+02	1.961293e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	1.986120e+02	1.971091e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	2.902117e+02	2.286949e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.671207e+02	2.295112e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.442009e+02	2.303216e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	2.215753e+02	2.311215e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	1.991022e+02	2.319160e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	2.907465e+02	2.643833e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.676267e+02	2.650336e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.447043e+02	2.656784e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	2.220753e+02	2.663149e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	1.995955e+02	2.669472e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	2.912799e+02	2.999803e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.681307e+02	3.004112e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.452049e+02	3.008379e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	2.225718e+02	3.012592e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	2.000845e+02	3.016777e+02

Table 4.7 Control points on the calibration block and detected image coordinates in the 5th frame

tu_6.tsa

	x_w (mm)	y_w (mm)	z_w (mm)	x_f (pixel)	y_f (pixel)
1	3.000000e+01	0.000000e+00	0.000000e+00	3.361370e+02	1.573601e+02
2	3.000000e+01	3.000000e+01	0.000000e+00	3.366598e+02	1.926505e+02
3	3.000000e+01	6.000000e+01	0.000000e+00	3.371834e+02	2.279897e+02
4	3.000000e+01	9.000000e+01	0.000000e+00	3.377074e+02	2.633642e+02
5	3.000000e+01	1.200000e+02	0.000000e+00	3.382307e+02	2.986835e+02
6	6.000000e+01	0.000000e+00	0.000000e+00	3.449808e+02	1.583844e+02
7	6.000000e+01	3.000000e+01	0.000000e+00	3.454814e+02	1.933760e+02
8	6.000000e+01	6.000000e+01	0.000000e+00	3.459804e+02	2.282597e+02
9	6.000000e+01	9.000000e+01	0.000000e+00	3.464810e+02	2.632464e+02
10	6.000000e+01	1.200000e+02	0.000000e+00	3.469814e+02	2.982290e+02
11	9.000000e+01	0.000000e+00	0.000000e+00	3.537150e+02	1.593960e+02
12	9.000000e+01	3.000000e+01	0.000000e+00	3.542191e+02	1.940947e+02
13	9.000000e+01	6.000000e+01	0.000000e+00	3.547192e+02	2.285279e+02
14	9.000000e+01	9.000000e+01	0.000000e+00	3.552219e+02	2.631290e+02
15	9.000000e+01	1.200000e+02	0.000000e+00	3.557252e+02	2.977749e+02
16	1.200000e+02	0.000000e+00	0.000000e+00	3.621567e+02	1.603738e+02
17	1.200000e+02	3.000000e+01	0.000000e+00	3.626479e+02	1.947880e+02
18	1.200000e+02	6.000000e+01	0.000000e+00	3.631333e+02	2.287860e+02
19	1.200000e+02	9.000000e+01	0.000000e+00	3.636220e+02	2.630162e+02
20	1.200000e+02	1.200000e+02	0.000000e+00	3.641119e+02	2.973394e+02
21	1.500000e+02	0.000000e+00	0.000000e+00	3.704751e+02	1.613372e+02
22	1.500000e+02	3.000000e+01	0.000000e+00	3.709858e+02	1.954738e+02
23	1.500000e+02	6.000000e+01	0.000000e+00	3.714881e+02	2.290424e+02
24	1.500000e+02	9.000000e+01	0.000000e+00	3.719948e+02	2.629038e+02
25	1.500000e+02	1.200000e+02	0.000000e+00	3.725035e+02	2.969036e+02
26	0.000000e+00	0.000000e+00	3.000000e+01	3.026666e+02	1.571070e+02
27	0.000000e+00	0.000000e+00	6.000000e+01	2.785392e+02	1.581892e+02
28	0.000000e+00	0.000000e+00	9.000000e+01	2.548029e+02	1.592537e+02
29	0.000000e+00	0.000000e+00	1.200000e+02	2.312777e+02	1.603088e+02
30	0.000000e+00	0.000000e+00	1.500000e+02	2.078890e+02	1.613578e+02
31	0.000000e+00	3.000000e+01	3.000000e+01	3.031780e+02	1.925249e+02
32	0.000000e+00	3.000000e+01	6.000000e+01	2.790910e+02	1.934842e+02
33	0.000000e+00	3.000000e+01	9.000000e+01	2.553355e+02	1.944303e+02
34	0.000000e+00	3.000000e+01	1.200000e+02	2.317722e+02	1.953687e+02
35	0.000000e+00	3.000000e+01	1.500000e+02	2.083653e+02	1.963010e+02
36	0.000000e+00	6.000000e+01	3.000000e+01	3.036922e+02	2.281399e+02
37	0.000000e+00	6.000000e+01	6.000000e+01	2.796454e+02	2.289482e+02
38	0.000000e+00	6.000000e+01	9.000000e+01	2.558702e+02	2.297474e+02
39	0.000000e+00	6.000000e+01	1.200000e+02	2.322684e+02	2.305408e+02
40	0.000000e+00	6.000000e+01	1.500000e+02	2.088428e+02	2.313283e+02
41	0.000000e+00	9.000000e+01	3.000000e+01	3.042053e+02	2.636732e+02
42	0.000000e+00	9.000000e+01	6.000000e+01	2.801986e+02	2.643276e+02
43	0.000000e+00	9.000000e+01	9.000000e+01	2.564036e+02	2.649761e+02
44	0.000000e+00	9.000000e+01	1.200000e+02	2.327632e+02	2.656205e+02
45	0.000000e+00	9.000000e+01	1.500000e+02	2.093190e+02	2.662595e+02
46	0.000000e+00	1.200000e+02	3.000000e+01	3.047207e+02	2.993650e+02
47	0.000000e+00	1.200000e+02	6.000000e+01	2.807534e+02	2.998188e+02
48	0.000000e+00	1.200000e+02	9.000000e+01	2.569379e+02	3.002698e+02
49	0.000000e+00	1.200000e+02	1.200000e+02	2.332584e+02	3.007182e+02
50	0.000000e+00	1.200000e+02	1.500000e+02	2.097947e+02	3.011625e+02

Table 4.8 Control points on the calibration block and detected image coordinates in the 6th frame

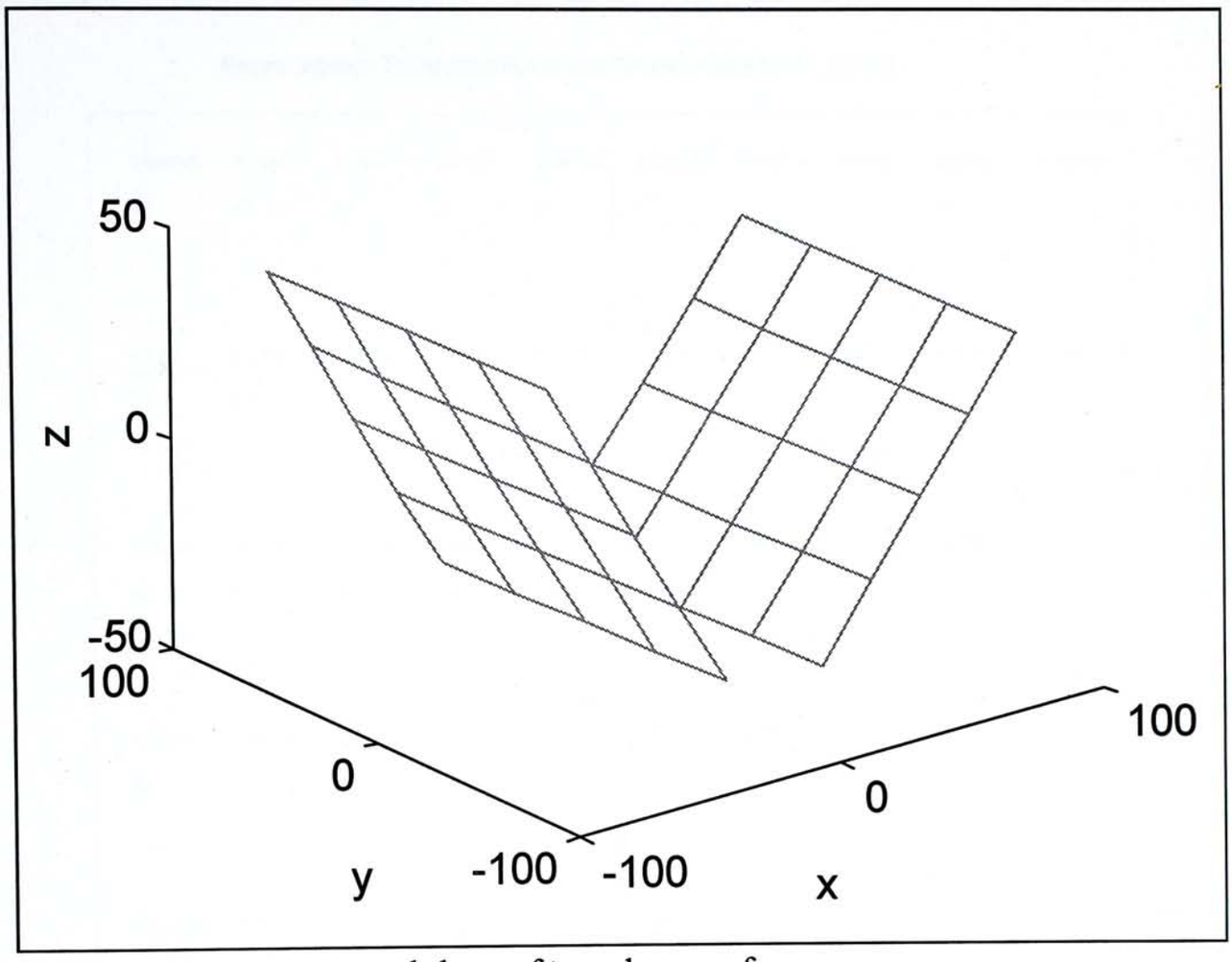


Figure 4.10 Reconstructed shape of two planar surface.

Errors between Transferred Points and Original Points in TU_2.IMG

0.0e+000	0.1496	0.0437	0.0735	0.0e+000	0.0e+000	0.1239	0.0682	0.0160	0.0e+000
30	29	28	27	26	1	6	11	16	21
0.2078	0.1773	0.1112	0.0769	0.0108	0.0642	0.0754	0.0755	0.1233	0.1831
35	34	33	32	31	2	7	12	17	22
0.0096	0.0246	0.0226	0.0249	0.0249	0.0247	0.0658	0.0983	0.1303	0.1644
40	39	38	37	36	3	8	13	18	23
0.0959	0.0704	0.0462	0.0366	0.0769	0.0628	0.0881	0.1081	0.1303	0.1519
45	44	43	42	41	4	9	14	19	24
0.0e+000	0.1090	0.0740	0.0449	0.0e+000	0.0e+000	0.0699	0.0212	0.0126	0.0e+000
50	49	48	47	46	5	10	15	20	25

Transferring the features in frame tu_1.img onto frame tu_2.img using the invariants computed in tu_1.img

The maximum error occurs at the 35th feature point. Maximum Error = 0.2078 pixel

Figure 4.11 Tracking error of the second frame. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

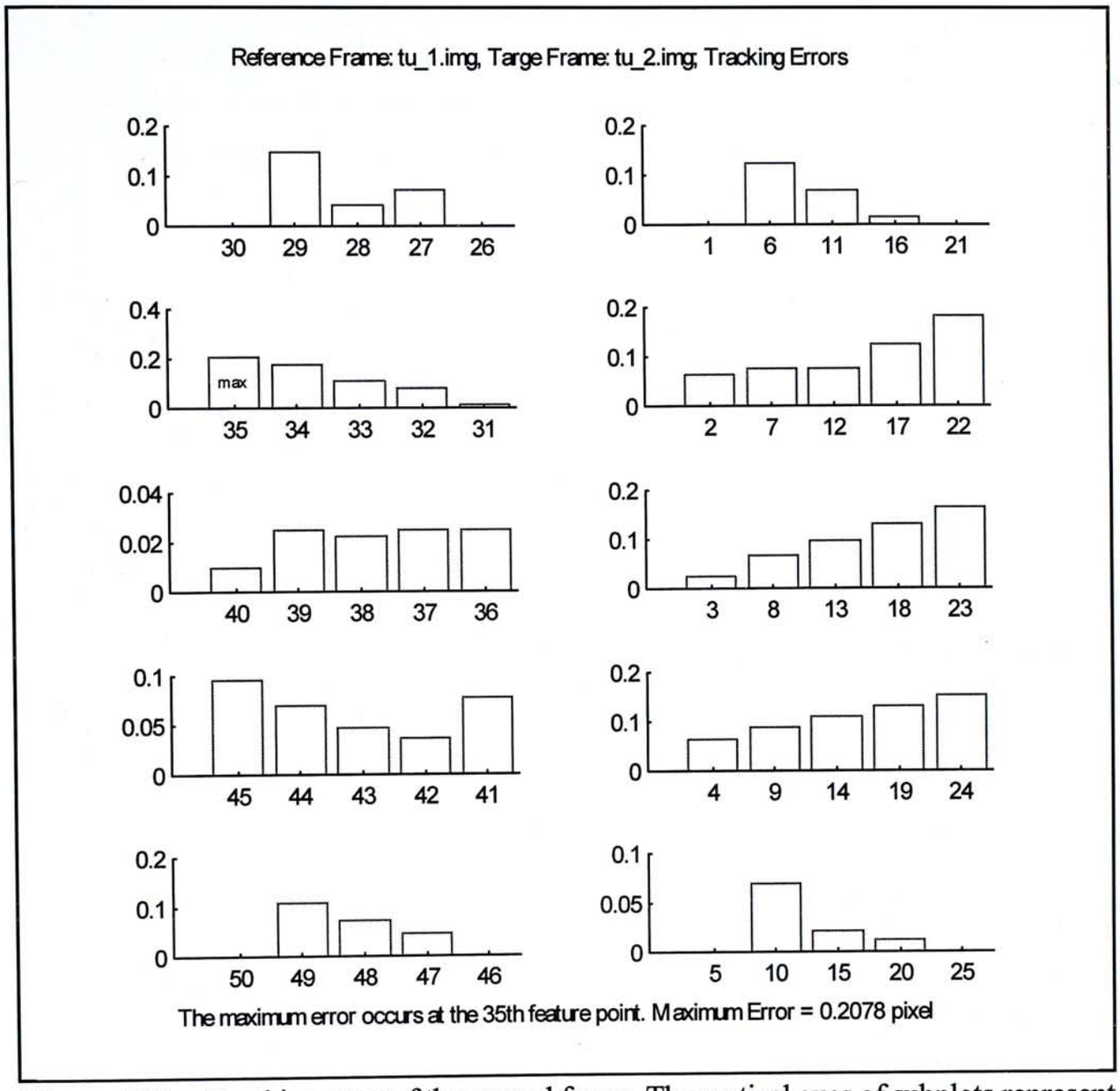


Figure 4.12 Tracking error of the second frame. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points in TU_3.IMG

0.0e+000	0.1920	0.1372	0.0361	0.0e+000	0.0e+000	0.0599	0.0563	0.0203	0.0e+000
30	29	28	27	26	1	6	11	16	21
0.1918	0.1809	0.1116	0.0340	0.0421	0.0670	0.0420	0.0379	0.0753	0.1213
35	34	33	32	31	2	7	12	17	22
0.0441	0.0711	0.0455	0.0364	0.0229	0.0821	0.0877	0.0910	0.0983	0.0984
40	39	38	37	36	3	8	13	18	23
0.0687	0.0321	0.0219	0.0475	0.0824	0.0111	0.0117	0.0485	0.0639	0.0599
45	44	43	42	41	4	9	14	19	24
0.0e+000	0.0706	0.0739	0.0111	0.0e+000	0.0e+000	0.0321	0.0742	0.0705	0.0e+000
50	49	48	47	46	5	10	15	20	25

Transferring the features in frame tu_1.img onto frame tu_3.img using the invariants computed in tu_1.img

The maximum error occurs at the 29th feature point. Maximum Error = 0.1920 pixel

Figure 4.13 Tracking error of the third frame.The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

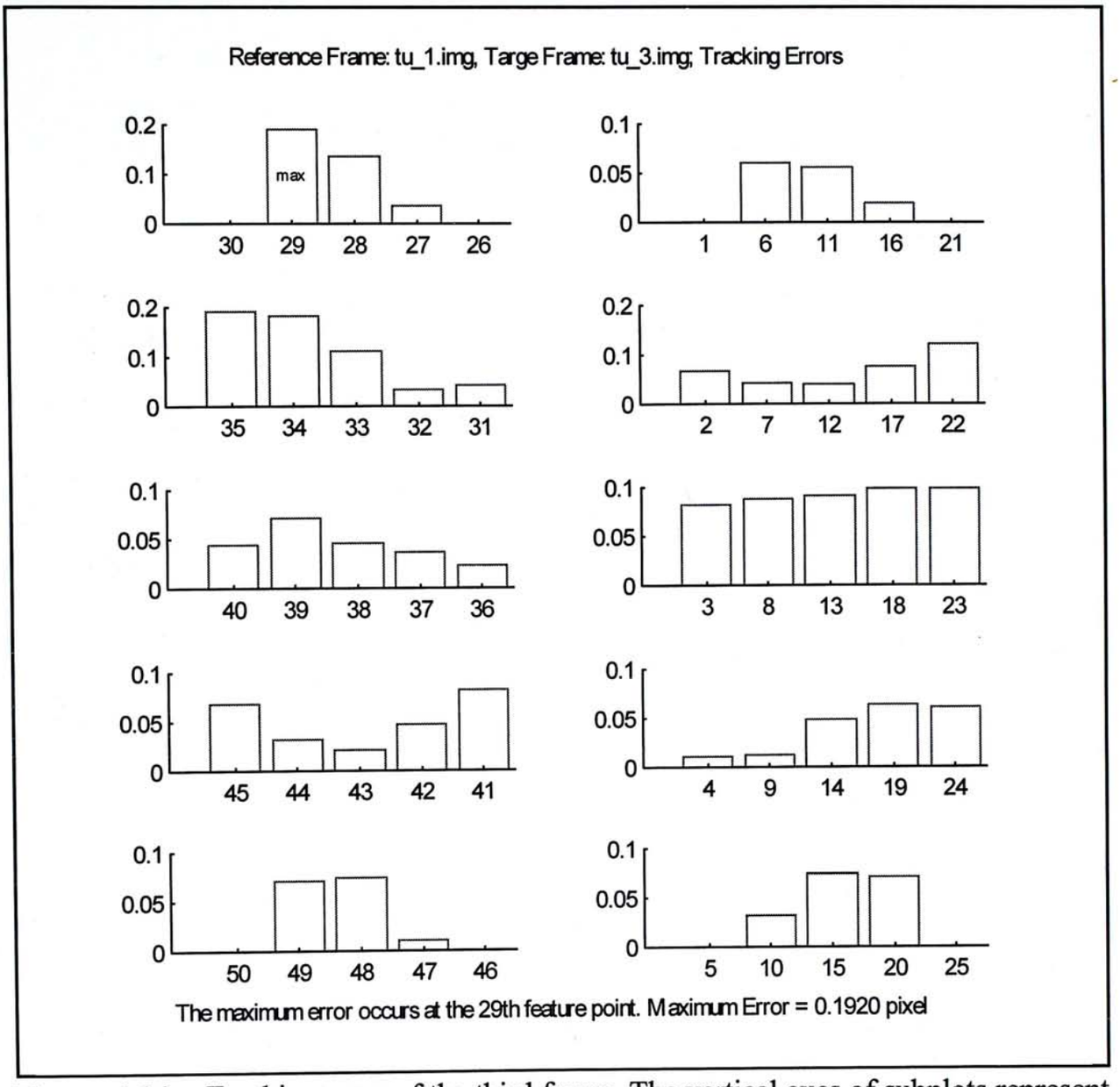


Figure 4.14 Tracking error of the third frame. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points in TU_4.IMG

0.0e+000	0.0603	0.1244	0.0469	0.0e+000	0.0e+000	0.1389	0.0443	0.0060	0.0e+000
30	29	28	27	26	1	6	11	16	21
0.1866	0.1367	0.0961	0.0276	0.0266	0.0738	0.1220	0.0833	0.0815	0.0834
35	34	33	32	31	2	7	12	17	22
0.0009	0.0480	0.0974	0.1503	0.1927	0.2821	0.2742	0.2572	0.2451	0.2330
40	39	38	37	36	3	8	13	18	23
0.0377	0.0303	0.0898	0.0995	0.0847	0.2255	0.1747	0.1262	0.0761	0.0281
45	44	43	42	41	4	9	14	19	24
0.0e+000	0.0594	0.1568	0.1263	0.0e+000	0.0e+000	0.0503	0.0388	0.0008	0.0e+000
50	49	48	47	46	5	10	15	20	25

Transferring the features in frame tu_1.img onto frame tu_4.img using the invariants computed in tu_1.img

The maximum error occurs at the 3th feature point. Maximum Error = 0.2821 pixel

Figure 4.15 Tracking error of the fourth frame. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

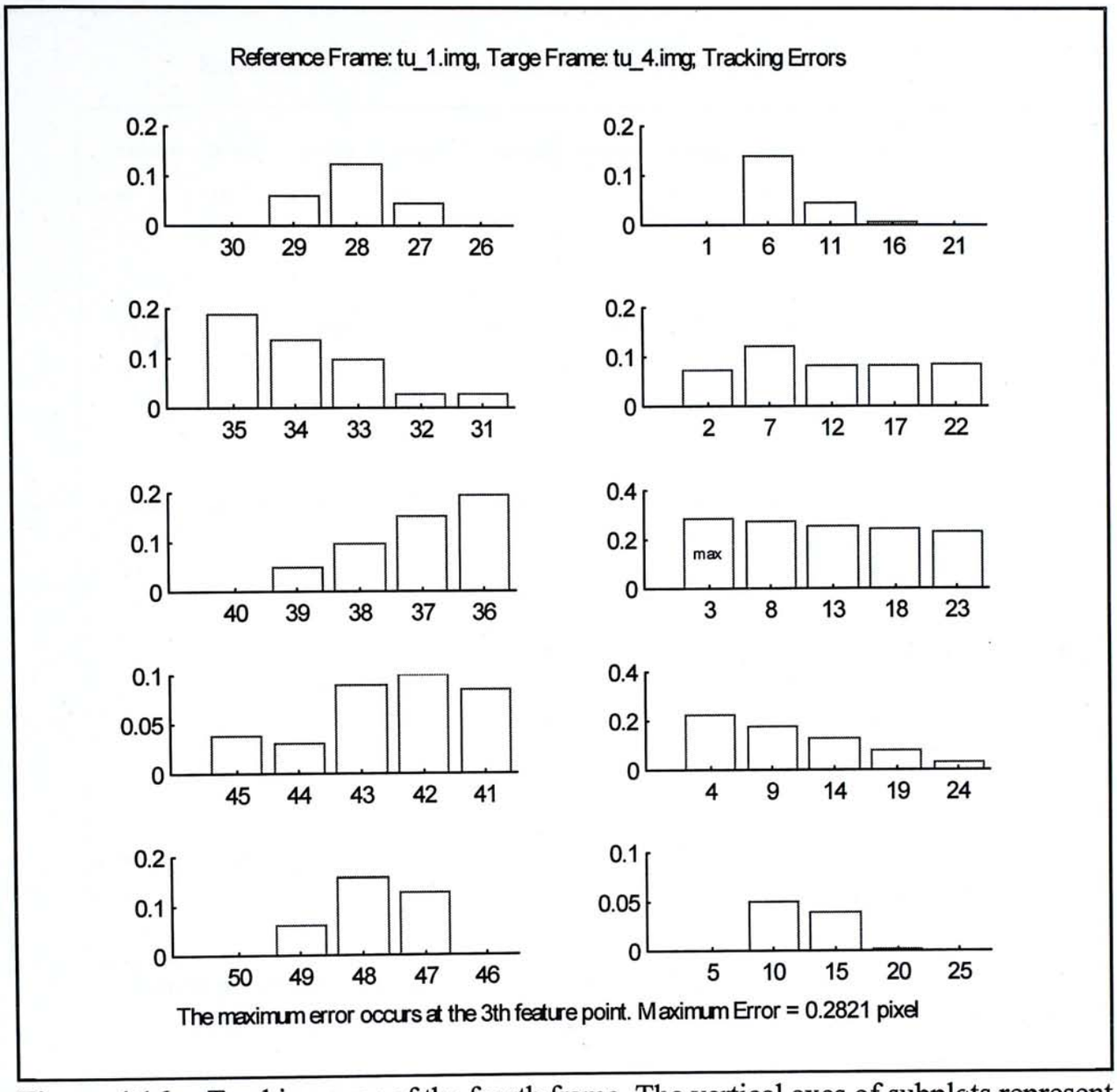


Figure 4.16 Tracking error of the fourth frame. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points in TU_5.IMG

0.0e+000	0.1615	0.1563	0.1161	0.0e+000	0.0e+000	0.0354	0.0653	0.0723	0.0e+000
30	29	28	27	26	1	6	11	16	21
0.1519	0.1764	0.1585	0.1316	0.1018	0.0598	0.0760	0.1030	0.1194	0.1224
35	34	33	32	31	2	7	12	17	22
0.0297	0.0639	0.0408	0.0273	0.0344	0.2786	0.2250	0.1646	0.1102	0.0553
40	39	38	37	36	3	8	13	18	23
0.1384	0.1284	0.1188	0.1104	0.0959	0.0048	0.0688	0.0093	0.0119	0.0134
45	44	43	42	41	4	9	14	19	24
0.0e+000	0.0374	0.0750	0.0745	0.0e+000	0.0e+000	0.1033	0.0324	0.0343	0.0e+000
50	49	48	47	46	5	10	15	20	25

Transferring the features in frame tu_1.img onto frame tu_5.img using the invariants computed in tu_1.img

The maximum error occurs at the 3th feature point. Maximum Error = 0.2786 pixel

Figure 4.17 Tracking error of the fifth frame. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

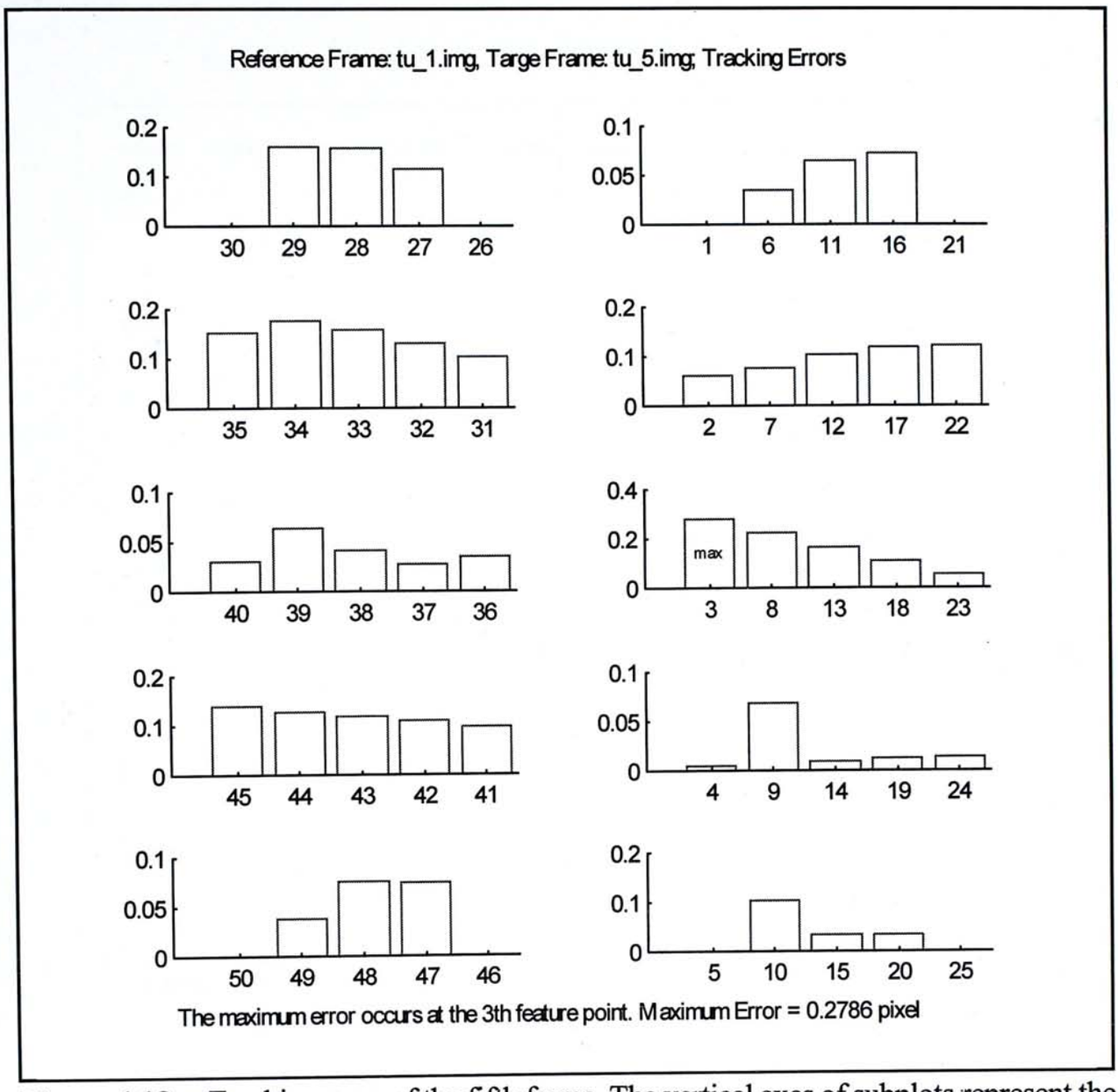


Figure 4.18 Tracking error of the fifth frame. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points in TU_6.IMG

0.0e+000	0.0956	0.0381	0.1016	0.0e+000	0.0e+000	0.0154	0.0577	0.0059	0.0e+000
30	29	28	27	26	1	6	11	16	21
0.1859	0.1444	0.0857	0.0885	0.0241	0.1280	0.0429	0.0943	0.1811	0.2812
35	34	33	32	31	2	7	12	17	22
0.1121	0.0873	0.0613	0.0721	0.0053	0.1656	0.1566	0.1260	0.1123	0.0853
40	39	38	37	36	3	8	13	18	23
0.0714	0.0474	0.0149	0.0477	0.0454	0.1463	0.1508	0.1192	0.1198	0.0923
45	44	43	42	41	4	9	14	19	24
0.0e+000	0.0590	0.0038	0.0262	0.0e+000	0.0e+000	0.0910	0.0062	0.0777	0.0e+000
50	49	48	47	46	5	10	15	20	25

Transferring the features in frame tu_1.img onto frame tu_6.img using the invariants computed in tu_1.img

The maximum error occurs at the 22th feature point. Maximum Error = 0.2812 pixel

Figure 4.19 Tracking error of the sixth frame. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

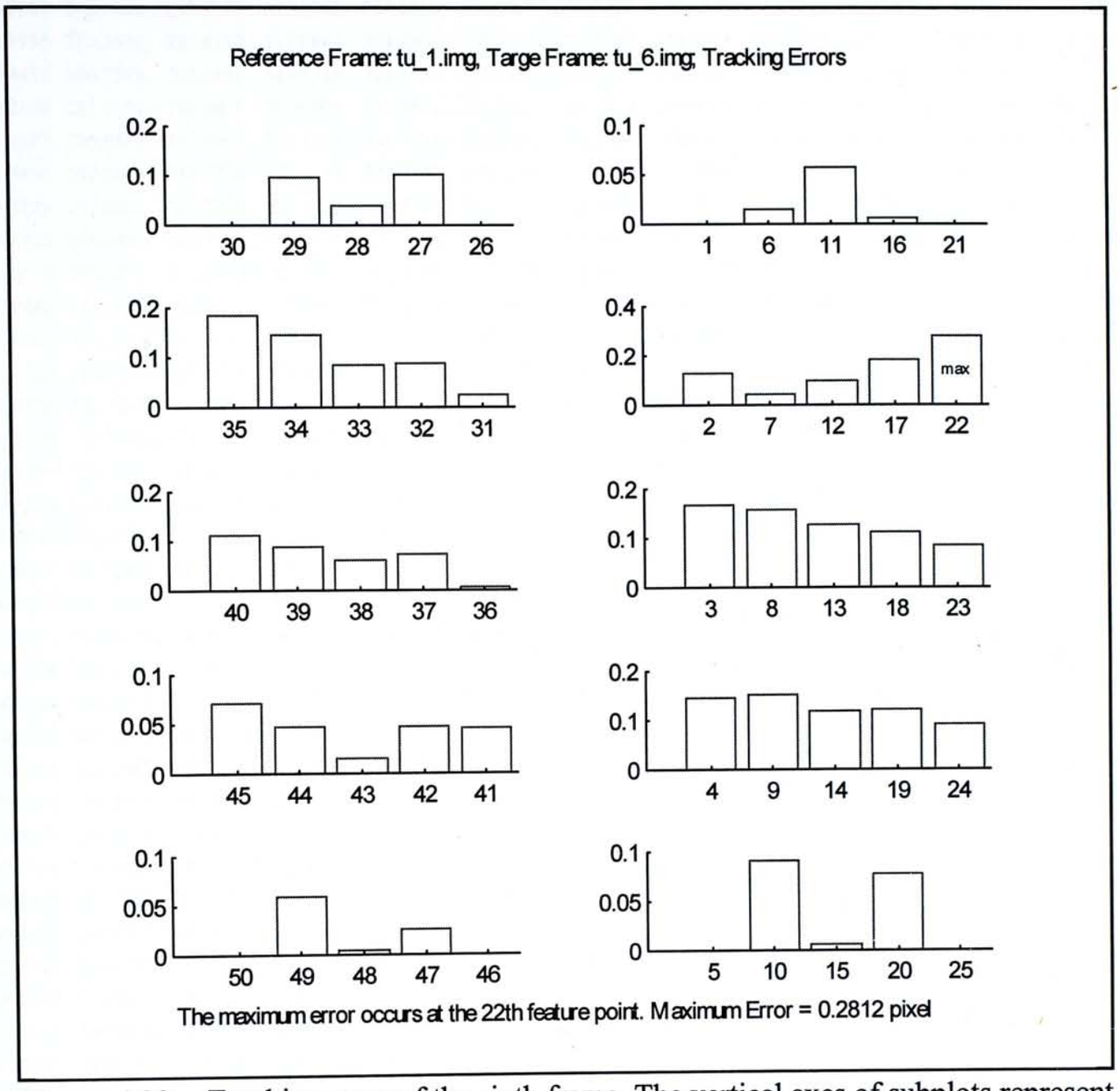


Figure 4.20 Tracking error of the sixth frame. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

u_1	u_2	u_3	u_4	u_5	u_6	v_1	v_2	v_3	v_4	v_5	v_6
261.7340	279.6109	296.9454	310.0787	323.8784	336.1370	158.4633	158.1713	157.5098	157.4398	157.4639	157.3601
262.1039	279.9869	297.3805	310.5843	324.4080	336.6617	194.0170	193.6745	193.0814	193.0825	193.0581	192.7785
262.4736	280.3632	297.8150	311.0884	324.9364	337.1858	229.5399	229.2028	228.6021	228.6175	228.5648	228.1553
262.8431	280.7399	298.2491	311.5913	325.4637	337.7096	265.0520	264.7764	264.0919	264.0653	264.0042	263.5105
263.2108	281.1154	298.6808	312.0906	325.9877	338.2307	300.3919	300.2331	299.3896	299.2653	299.2157	298.6835
281.1683	296.9103	311.8666	323.1516	334.7754	344.9961	158.9831	158.8161	158.2656	158.3269	158.5158	158.3862
281.5406	297.3010	312.2974	323.6414	335.2932	345.5160	194.2391	193.9805	193.4863	193.5798	193.6910	193.4013
281.9129	297.6924	312.7279	324.1301	335.8103	346.0357	229.4954	229.2002	228.6875	228.7576	228.8111	228.4062
282.2850	298.0842	313.1580	324.6175	336.3262	346.5549	264.7291	264.4523	263.8464	263.8379	263.8533	263.3778
282.6558	298.4752	313.5863	325.1022	336.8396	347.0723	299.8474	299.6437	298.8704	298.7289	298.7258	298.2244
300.2842	313.8863	326.4996	335.9448	345.4214	353.6577	159.4945	159.4489	159.0069	159.1951	159.5435	159.3894
300.6669	314.2984	326.9324	336.4247	345.9325	354.1765	194.4576	194.2809	193.8835	194.0665	194.3096	194.0106
301.0500	314.7117	327.3654	336.9041	346.4432	354.6956	229.4516	229.1976	228.7713	228.8948	229.0519	228.6517
301.4326	315.1250	327.7977	337.3820	346.9525	355.2141	264.4110	264.1338	263.6052	263.6150	263.7056	263.2479
301.8146	315.5383	328.2290	337.8580	347.4602	355.7314	299.3108	299.0642	298.3603	298.2029	298.2462	297.7746
319.1393	330.5921	340.8907	348.5003	355.8527	362.1508	159.9988	160.0715	159.7359	160.0472	160.5504	160.3731
319.5140	331.0088	341.3116	348.9588	356.3471	362.6603	194.6729	194.5763	194.2737	194.5438	194.9151	194.6073
319.8894	331.4270	341.7331	349.4171	356.8416	363.1707	229.4085	229.1951	228.8537	229.0292	229.2875	228.8919
320.2643	331.8452	342.1538	349.8738	357.3347	363.6802	264.0982	263.8213	263.3688	263.3970	263.5614	263.1209
320.6392	332.2641	342.5743	350.3296	357.8270	364.1895	298.7836	298.4963	297.8606	297.6886	297.7782	297.3354
337.7275	347.0239	355.0372	360.8172	366.0693	370.4751	160.4960	160.6840	160.4525	160.8830	161.5366	161.3372
338.1078	347.4569	355.4568	361.2637	366.5550	370.9816	194.8853	194.8669	194.6576	195.0123	195.5085	195.1926
338.4891	347.8917	355.8773	361.7105	367.0412	371.4894	229.3659	229.1926	228.9346	229.1612	229.5185	229.1277
338.8699	348.3265	356.2970	362.1557	367.5260	371.9962	263.7891	263.5133	263.1358	263.1825	263.4198	262.9961
339.2511	348.7625	356.7169	362.6005	368.0106	372.5035	298.2624	297.9361	297.3680	297.1827	297.3184	296.9036
225.0616	242.7645	260.5464	274.2928	289.1457	302.6666	159.2765	158.7453	157.8848	157.5808	157.5581	157.1070
208.6461	224.0898	239.8558	252.3603	265.9970	278.6407	160.7999	160.1513	159.1976	158.8470	158.7678	158.1846
192.4951	205.6667	219.4102	230.6325	243.0429	254.7648	162.2987	161.5384	160.4947	160.1013	159.9673	159.2554
176.6988	187.6007	199.3283	209.2378	220.4199	231.1822	163.7646	162.8985	161.7688	161.3364	161.1495	160.3131
161.2466	169.8827	179.6017	188.1698	198.1221	207.8890	165.1986	164.2325	163.0204	162.5527	162.3147	161.3578
225.4317	243.1517	260.9635	274.8197	289.6773	303.1784	194.6477	194.2201	193.3878	193.0379	193.0357	192.5490
209.0285	224.5047	240.2967	252.9034	266.5468	279.1734	195.8202	195.3210	194.4249	194.0742	194.0319	193.4520
192.8816	206.0998	219.8645	231.1812	243.5996	255.3066	196.9743	196.4074	195.4502	195.1011	195.0200	194.3496
177.0760	188.0361	199.7787	209.7739	220.9643	231.7132	198.1040	197.4737	196.4580	196.1131	195.9946	195.2369
161.5840	170.2850	180.0090	188.6515	198.6099	208.3628	199.2113	198.5216	197.4500	197.1117	196.9572	196.1151
225.8034	243.5400	261.3821	275.3488	290.2112	303.6922	230.1619	229.7886	229.0104	228.6441	228.6605	228.1346
209.4129	224.9213	240.7396	253.4495	267.0996	279.7088	231.0338	230.6355	229.8229	229.5010	229.4939	228.9138
193.2707	206.5352	220.3215	231.7336	244.1601	255.8521	231.8925	231.4720	230.6266	230.3508	230.3210	229.6888
177.4565	188.4748	200.2328	210.3147	221.5135	232.2489	232.7338	232.2936	231.4173	231.1890	231.1376	230.4557
161.9247	170.6907	180.4199	189.1380	199.1025	208.8412	233.5600	233.1026	232.1971	232.0176	231.9457	231.2162
226.1750	243.9277	261.8003	275.8779	290.7450	304.2060	265.6747	265.3057	264.6075	264.2542	264.2874	263.7186
209.7971	225.3371	241.1820	253.9953	267.6521	280.2440	266.2147	265.8682	265.1641	264.9001	264.9262	264.3422
193.6592	206.9693	220.7776	232.2853	244.7198	256.3967	266.7468	266.4239	265.7150	265.5410	265.5606	264.9629
177.8359	188.9116	200.6853	210.8541	222.0612	232.7830	267.2685	266.9702	266.2574	266.1735	266.1873	265.5774
162.2642	171.0944	180.8292	189.6229	199.5935	209.3180	267.7819	267.5092	266.7934	266.8002	266.8088	266.1881
226.5473	244.3155	262.2190	276.4081	291.2799	304.7207	301.2497	300.8353	300.2429	299.9323	299.9803	299.3650
210.1817	225.7527	241.6246	254.5418	268.2052	280.7796	301.4432	301.0994	300.5293	300.3524	300.4098	299.8183
194.0480	207.4031	221.2337	232.8374	245.2799	256.9417	301.6340	301.3604	300.8129	300.7694	300.8365	300.2697
178.2155	189.3480	201.1377	211.3939	222.6092	233.3174	301.8212	301.6173	301.0924	301.1814	301.2585	300.7171
162.6037	171.4975	181.2382	190.1078	200.0845	209.7947	302.0058	301.8712	301.3691	301.5904	301.6777	301.1625

Table 4.9 The transpose of the mesurement matrix which is contructed on the basis of tracked feature points.

4.4 Discussion and Future Research

We have proposed a new tracking method for tracking feature points for polyhedral objects over an image sequence. This method can be used to improve the speed and reliability of those shape from motion techniques requiring feature tracking. To verify this, real experiments on shape from motion using the factorization method of Tomasi and Kanade[2] were described in this chapter. The shape of a calibration block was recovered with a slight error of about 3 degree on the angle on the angle between two orthogonal surfaces. It should be noted that much of the errors may have been introduced by the orthographic projection assumption of the factorization method. In our method, we have made use of the accurate active vision system for the tracking of the projective basis. This means the tracking has been reduced to conventional stereo matching along epipoloar lines. Since the basis features are usually easily detected, the tracking is highly accurate and fast. The tracking by geometric invariance is even more simple and also very fast. Thus, the overall speed of the tracking can be quite fast even if hundreds of feature points over a dozon image frames are to be tracked.

To improve on our existing method in the future, we shall use extended Kalman filter[4] to predict the position of the projective basis in all the frames except for the first. Research will also be conducted on how to select the projective basis and how to change the reference frame dynamically. This shall make the tracking more accurate. We shall also investigate the use of geometric invariance for feature tracking of non-polyhedral objects.

References

[1] Roger Mohr, Luce Morin, and Enrico Grosso, " Relative positioning with uncalibrated cameras ", In J.Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pp. 440-460. The MIT press, 1992.

[2] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method," *Technical Report CMU-CS-91-172*, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[3] C.J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery, *Technical Report CMU-CS-93-219*, Carnegie Mellon University, Pittsburgh, PA, December 1993.

[4] T. H. Wu, R. Chellppa and Q Zheng, "Experiments on Estimating Egomotion and Structure Parameters Using Long Monocular Image Sequences", *International Journal of Computer Vision*, 15, 77-103,1995.

[5] N. V. Efimov, Advanced Geometry, Mir, Moscow, 1978.

Chapter 5
Experiments on Feature Tracking with 3D Projective Invariants

In this chapter, we present the experimental results on tracking of 3D feature point using projective invariants.

Generally, the geometric relation between objects in the world and their images, acquired by a pinhole camera, can be modelled as a perspective projection from a 3-Dimensional object space to a 2-Dimensional image plane. In the framework of analytical geometry, we can consider object space as the 3-D projective space P^3 , and image space as the 2-D projective space P^2 (a projective plane) [1]. Therefore, the perspective projection from a 3-Dimensional object space to a 2-Dimensional image plane can be described in terms of a more general projective transformation. The relationship between any two views of the same scene can be represented as a planar projective transformation.

As pointed out by Felix Klein in1872, the basis for geometry is the set of properties which remain invariant under a given transformation group. Being defined as a set of geometric properties which remained unchanged under projective transformation, projective invariants had been derived and applied in many vision tasks[3].

The derivation of 8 -points projective invariant is given in the following section[3].

5.1 8-points Projective Invariant

Let $\psi_1, \psi_2 \in P^2$ represent two distinct images of the same set of object points, there exists a 3×3 projective transformation matrix ,denoted by $\boldsymbol{A}$, which can transform ψ_1 to ψ_2. Assuming that camera centers for ψ_1, ψ_2 are located at $O, O' \in P^3$, respectively. The epipoles υ and υ' are defined to be at the intersection of the line $\overline{OO'}$ with both image planes. The symbol $\cong$ denotes the equality up to a scale. Let $\boldsymbol{p} \in \psi_1$, $\boldsymbol{p}' \in \psi_2$ be two corresponding points, and let $\boldsymbol{l}$, $\boldsymbol{l}'$ be their corresponding epipolar lines, υ and υ' be the epipoles of ψ_1, ψ_2, respectively, we have $\boldsymbol{l} \cong \upsilon \times \boldsymbol{p}$ $\boldsymbol{l}' \cong \upsilon' \times \boldsymbol{p}'$ (see [4], ***Proposition 2.3***, A line going through two points $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, is represented by the cross-product $\boldsymbol{m}_1 \times \boldsymbol{m}_2$.), since the epipolar line of an image point passes through both the image point and the epipole of the image plane. As lines are projective invariants, any point $\boldsymbol{p}$ along $\boldsymbol{l}$ is mapped by $\boldsymbol{A}$ to some point $\boldsymbol{A}\boldsymbol{p}$ along $\boldsymbol{l}'$, thus, $\boldsymbol{l}' \cong \upsilon' \times \boldsymbol{A}\boldsymbol{p}$, and because $\boldsymbol{p}'$ is incident with $\boldsymbol{l}'$, we have $\boldsymbol{p}'^T(\upsilon' \times \boldsymbol{A}\boldsymbol{p}) = 0$ ([4] , the *point equation* of the line), or , equivalently the following,

$$\boldsymbol{p}'^T[\upsilon']\boldsymbol{A}\boldsymbol{p} = 0 \tag{5.1}$$

where $[\upsilon']$ is the skew-symmetric matrix of υ' (i.e., $[\upsilon']\boldsymbol{x} = \upsilon' \times \boldsymbol{x}$ for any vector x). Let $\boldsymbol{F} = [\upsilon']\boldsymbol{A}$, we have

$$\boldsymbol{p}'^T\boldsymbol{F}\boldsymbol{p} = 0 \tag{5.2}$$

where the matrix $\boldsymbol{F}$ is called the fundamental matrix. The matrix $\boldsymbol{F}$ is a 3×3 matrix of rank 2 (because of $[\upsilon']$).

Let

$$\boldsymbol{p} = (x_p, y_p, 1)^T \quad \boldsymbol{p}' = (x_{p'}, y_{p'}, 1)^T$$

$$\boldsymbol{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3]$$

where $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3$ are three column vectors of $\boldsymbol{F}$, Rewriting (5.2) as

$$\boldsymbol{p}'^T [\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3] \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = 0 \tag{5.3}$$

Rearranging (5.3) gives

$$\boldsymbol{p}'^T [x_p, y_p, 1] \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix} = 0 \tag{5.4}$$

Thus, we have

$$[x_p \boldsymbol{p}'^T, y_p \boldsymbol{p}'^T, \boldsymbol{p}'^T] \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix} = 0 \tag{5.5}$$

Let

$$\boldsymbol{b} = [x_p \boldsymbol{p}'^T, y_p \boldsymbol{p}'^T, \boldsymbol{p}'^T]^T$$

$$= [x_p x_{p'}, x_p y_{p'}, x_p, y_p x_{p'}, y_p y_{p'}, y_p, x_{p'}, y_{p'}, 1]^T$$

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix} \tag{5.6}$$

$$= [f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33}]^T$$

Substituting (5.6) into (5.5) yields

$$\boldsymbol{b}^T \boldsymbol{f} = 0 \tag{5.7}$$

Supposing that $\boldsymbol{p}_i \in \psi_1$, $\boldsymbol{p}'_i \in \psi_2$ $i = 1, ..., 9$ are nine image point correspondences across two views ψ_1 and ψ_2 , we can obtain the following linear system of equations

$$\boldsymbol{b}_i^T \boldsymbol{f} = 0 \quad i = 1, ..., 9 \tag{5.8}$$

where $\boldsymbol{b}_i^T$ can be obtained by applying (5.6) to the *nine* point correspondences $\boldsymbol{p}_i = (x_{p_i}, y_{p_i}, 1)^T \in \psi_1$, $\boldsymbol{p}'_i = (x_{p'_i}, y_{p'_i}, 1)^T \in \psi_2$ $i = 1, ..., 9$

$$\boldsymbol{b}_i = [x_{p_i} x_{p'_i}, x_{p_i} y_{p'_i}, x_{p_i}, y_{p_i} x_{p'_i}, y_{p_i} y_{p'_i}, y_{p_i}, x_{p'_i}, y_{p'_i}, 1]^T \tag{5.9}$$

We can rewrite (5.8) as following

$$\boldsymbol{B}\boldsymbol{f} = 0 \tag{5.10}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1^T \\ \boldsymbol{b}_2^T \\ \cdots \\ \boldsymbol{b}_9^T \end{bmatrix} \tag{5.11}$$

Since $\boldsymbol{b}_i^T$, $i = 1, ..., 9$ are 1×9 vectors, $\boldsymbol{B}$ is a 9×9 matrix. In order to obtain a non-trival solution for $\boldsymbol{f}$, the determinant of $\boldsymbol{B}$ must be identically zero. Thus , the condition $|\boldsymbol{B}| = 0$ is a projective invariant for two views ψ_1 and ψ_2 since it holds for any arbitrary two views and any arbitrarily selected nine point correspondences.

5.2 Projective Invaraint Based Tranfer between Distinct Views of a 3-D Scene

This projective invariant can be used for developing a tranfer between distinct views of a 3-D scene[2]. Given two views as reference views, it is possible to transform any feature point in each of the reference view to a third target image on the condition that eight point correspondences are known across all three frames. Since the transfer is developed on the basis of projective invariant, there is no need for any external and internal camera parameter, only image coordinates of the feature points are involved in computation. The transfer approach is depicted in Fig. 5.1.

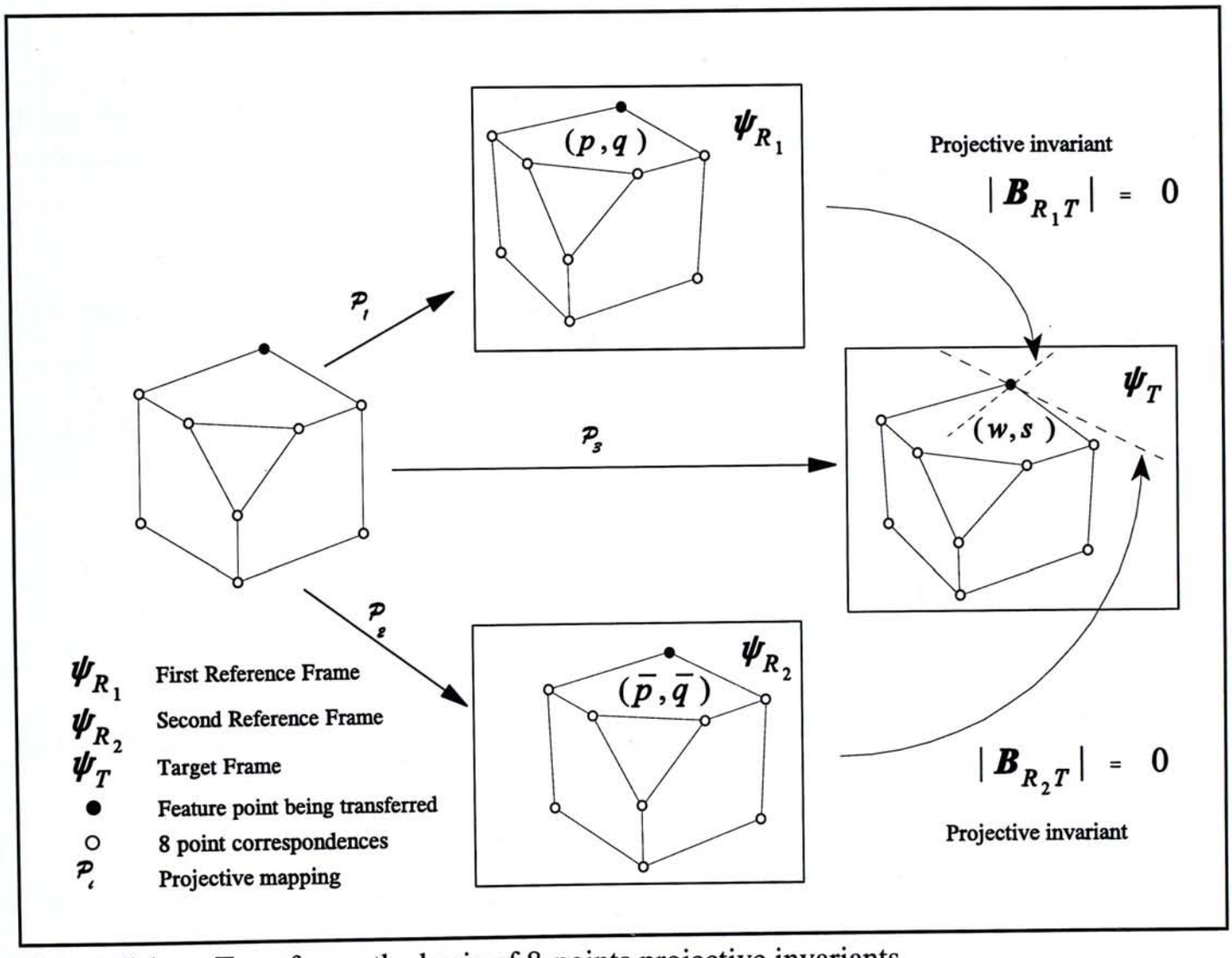


Figure 5.1 Transfer on the basis of 8-points projective invariants

Let ψ_{R_1} and ψ_{R_2} be two different reference image of a 3D scene, ψ_T be a target image of the same scene. It is assumed that eight corresponding points can be determined between ψ_{R_1} and ψ_{R_2}, ψ_{R_1} and ψ_T, as well as ψ_{R_2} and ψ_T., denoted by $(p_j,q_j) \in \psi_{R_1}$, $(\bar{p}_j,\bar{q}_j) \in \psi_{R_2}$ and $(w_j,s_j) \in \psi_T$ $j = 1,\ldots,8$, respectively. We wish to transfer any other corresponding reference image point pairs: $(p,q) \in \psi_{R_1}$, $(\bar{p},\bar{q}) \in \psi_{R_2}$, to a target image point $(w,s) \in \psi_T$. using the projective invariant which we rederived in the previous section. First, considering projective invariant which for first

reference view ψ_{R_1} and target view ψ_T, $|\boldsymbol{B}_{R_1T}| = 0$ holds for the nine point correspondences.That is, the point set $(p_j,q_j) \in \psi_{R_1}$, $j = 1...8$ and$(p,q) \in \psi_{R_1}$ corresponds to the point set $(w_j,s_j) \in \psi_T$ $j = 1...8, (w,s) \in \psi_T$. The invariant can be written as

$$|\boldsymbol{B}_{R_1T}| = \begin{vmatrix} p_1w_1 & p_1s_1 & p_1 & q_1w_1 & q_1s_1 & q_1 & w_1 & s_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_8w_8 & p_8s_8 & p_8 & q_8w_8 & q_8s_8 & q_8 & w_8 & s_8 & 1 \\ pw & ps & p & qw & qs & q & w & s & 1 \end{vmatrix} = 0 \tag{5.12}$$

It can be expanded as following form by absorbing all but the w and s dependencies in coefficients:

$$A^{(1)}w + B^{(1)}s + C^{(1)} = 0 \tag{5.13}$$

This means that the (w,s)solution lies on a line in the target view ψ_T . Similiarly, for second reference view ψ_{R_2} and target view ψ_T, $|\boldsymbol{B}_{R_2T}| = 0$ holds for $(\bar{p}_j,\bar{q}_j) \in \psi_{R_2}$, $(w_j,s_j) \in \psi_T$ $j = 1...8, (\bar{p},\bar{q}) \in \psi_{R_2}$ $(w,s) \in \psi_T$. The invariant can be written as

$$|\boldsymbol{B}_{R_2T}| = \begin{vmatrix} \bar{p}_1w_1 & \bar{p}_1s_1 & \bar{p}_1 & \bar{q}_1w_1 & \bar{q}_1s_1 & \bar{q}_1 & w_1 & s_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{p}_8w_8 & \bar{p}_8s_8 & \bar{p}_8 & \bar{q}_8w_8 & \bar{q}_8s_8 & \bar{q}_8 & w_8 & s_8 & 1 \\ \bar{p}w & \bar{p}s & \bar{p} & \bar{q}w & \bar{q}s & \bar{q} & w & s & 1 \end{vmatrix} = 0 \tag{5.14}$$

(5.14) defines a second line:

$$A^{(2)}w + B^{(2)}s + C^{(2)} = 0 \tag{5.15}$$

in the target view ψ_T . The intersection of these two lines determines $(w,s) \in \psi_T$ for any corresponding point pairs:$(p,q) \in \psi_{R_1}$, $(\bar{p},\bar{q}) \in \psi_{R_2}$ once eight point correspondence are available.

5.3 Transfer Experiments on the Image Sequence of an Calibration Block

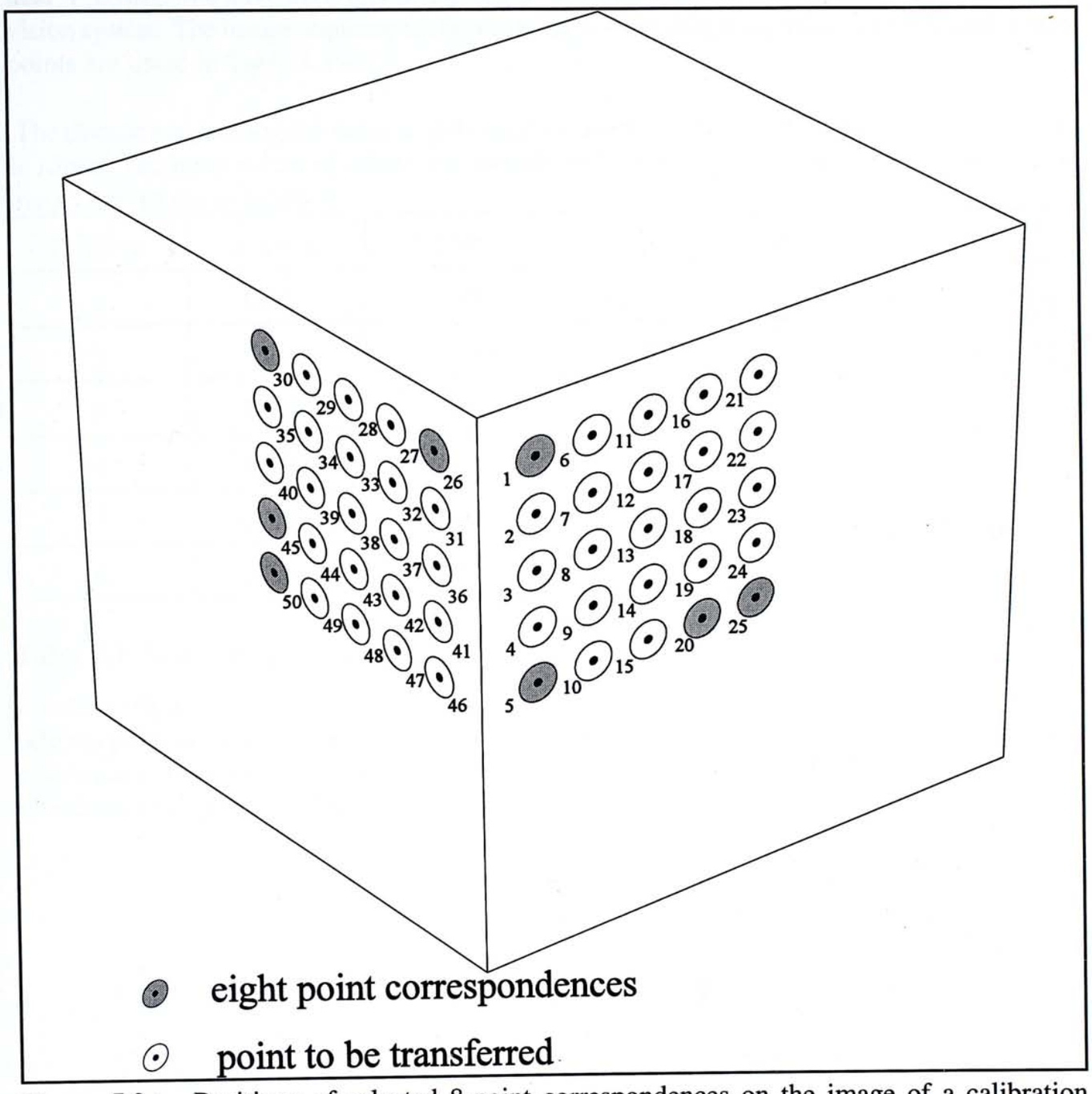


Figure 5.2 Positions of selected 8-point correspondences on the image of a calibration block.

We have developed a MATLAB transfer procudure on the basis of eight point correspondences projective invariant. The program, named *tracker3D.m* has been tested on several image sequences of a camera calibration block. In the program, feature points with indices 1, 5, 20, 25, 26, 30, 45, 50 are chosen as 8-points correspondences set. (see Fig 2.7 for the definition of control point index.)

5.3.1 Experiment 1. Real Image Sequence 1 of a Camera Calibration Block

In this experiment , we test the transfer algorithm on the image sequence near_r_1.img .... near_r_6.img. We have used this image sequence to do the head eye calibration for the active vision system. The image sequence is shown in Fig 3.4, the image coordinates of detected control points are listed in Table 3.3 - 3.8.

The distance separating the camera and object is about one meter. The focal length of the lens is 16mm. The input values of robot joint variable to the active vision system and corresponding image frames are as follows:

Position	***X*** (mm)	***Y*** (mm)	***D*** (°)	***E*** (°)	Frame
1	0.000	98.000	0.000	6.000	near_r_1.img
2	-20.000	98.000	5.000	6.000	near_r_2.img
3	-20.000	78.000	5.000	3.000	near_r_3.img
4	-20.000	78.000	0.000	3.000	near_r_4.img
5	-20.000	98.000	0.000	6.000	near_r_5.img
6	-70.000	98.000	-4.000	6.000	near_r_6.img

Table 5.1 Input values and corresponding image frames

Fourteen triple of image frames are tested. The maximum error between transferred point and original point in the target frame is reported in Table 5.2 for each triple of image frames. Three examples of distributions of the tracking errors across feature points on the target frames are illustrated in Figure 5.3 - Figure 5.8, in both graphical and digital form.

	First_Ref	Second_Ref	Target_Ref	Maximum_error(pix)
1	near_r_1.img	near_r_2.img	near_r_3.img	0.6497
2	near_r_1.img	near_r_2.img	near_r_4.img	5.1895
3	near_r_1.img	near_r_2.img	near_r_5.img	61.9914
4	near_r_1.img	near_r_2.img	near_r_6.img	41.6875
5	near_r_2.img	near_r_3.img	near_r_4.img	4.1364
6	near_r_2.img	near_r_3.img	near_r_5.img	4.0042
7	near_r_2.img	near_r_3.img	near_r_6.img	8.5364
8	near_r_3.img	near_r_4.img	near_r_5.img	0.3139
9	near_r_3.img	near_r_4.img	near_r_6.img	10.7559
10	near_r_4.img	near_r_5.img	near_r_6.img	3.3508
11	near_r_1.img	near_r_6.img	near_r_2.img	12.6503
12	near_r_1.img	near_r_6.img	near_r_3.img	1.3977
13	near_r_1.img	near_r_6.img	near_r_4.img	1.0111
14	near_r_1.img	near_r_6.img	near_r_5.img	6.1842

Table 5.2 Maximum errors in pixels between the tracked feature points in the target frame and the original feature points in the target frame

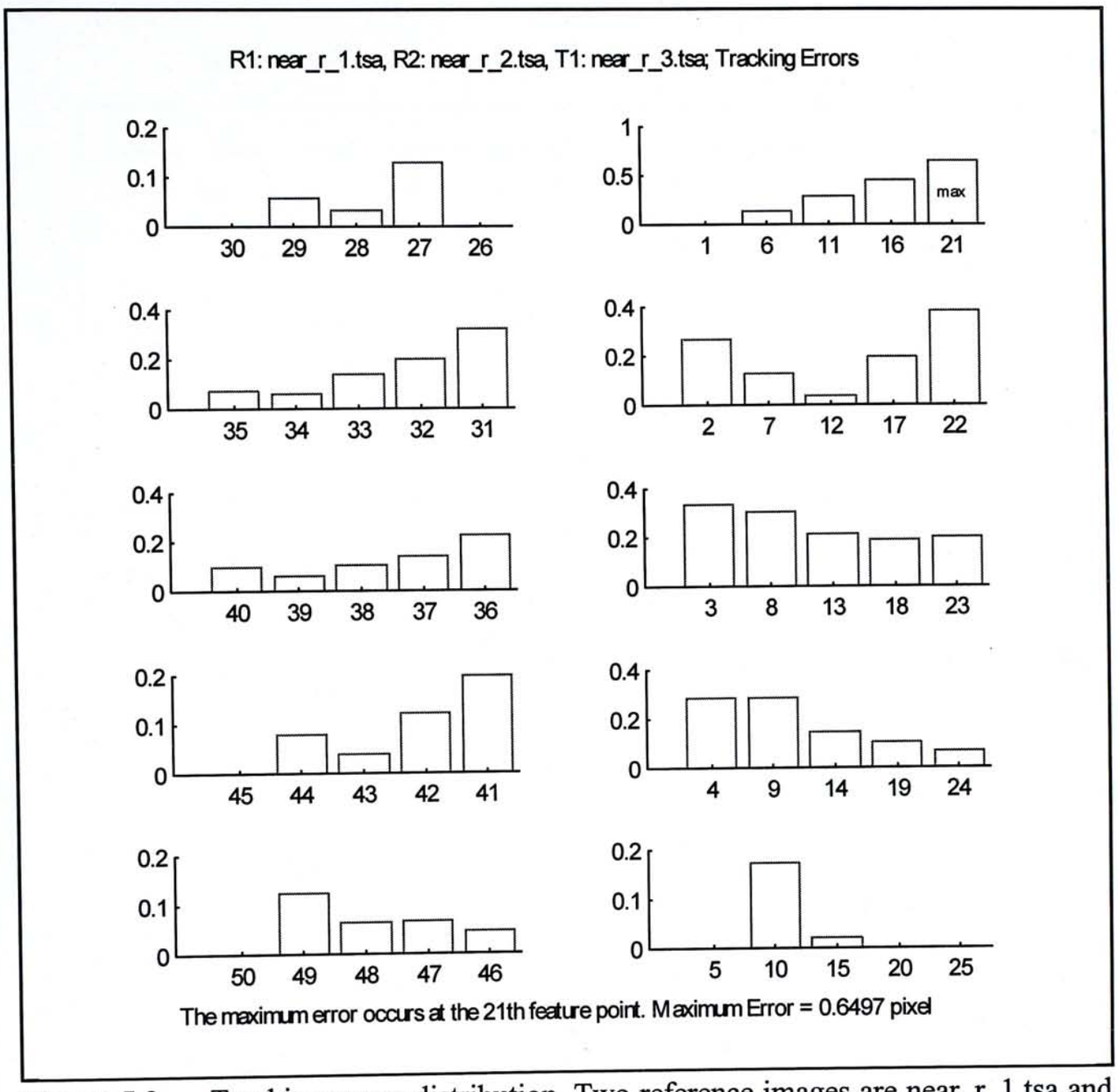


Figure 5.3 Tracking errors distribution. Two reference images are near_r_1.tsa and near_r_2.tsa. The target image is near_r_3.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

1.6e-012	0.0570	0.0328	0.1308	2.7e-012	3.5e-012	0.1434	0.2901	0.4541	0.6497
30	29	28	27	26	1	6	11	16	21
0.0745	0.0632	0.1419	0.1989	0.3222	0.2692	0.1300	0.0379	0.1944	0.3796
35	34	33	32	31	2	7	12	17	22
0.0991	0.0599	0.1046	0.1381	0.2253	0.3350	0.3039	0.2149	0.1853	0.2023
40	39	38	37	36	3	8	13	18	23
3.6e-012	0.0781	0.0390	0.1217	0.1960	0.2852	0.2856	0.1443	0.1022	0.0692
45	44	43	42	41	4	9	14	19	24
4.1e-012	0.1246	0.0648	0.0676	0.0459	1.5e-011	0.1742	0.0216	2.0e-011	2.1e-011
50	49	48	47	46	5	10	15	20	25

R1: near_r_1.tsa, R2: near_r_2.tsa, T1: near_r_3.tsa

The maximum error occurs at the 21th feature point. Maximum Error = 0.6497 pixel

Figure 5.4 Tracking errors distribution. Two reference images are near_r_1.tsa and near_r_2.tsa. The target image is near_r_3.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

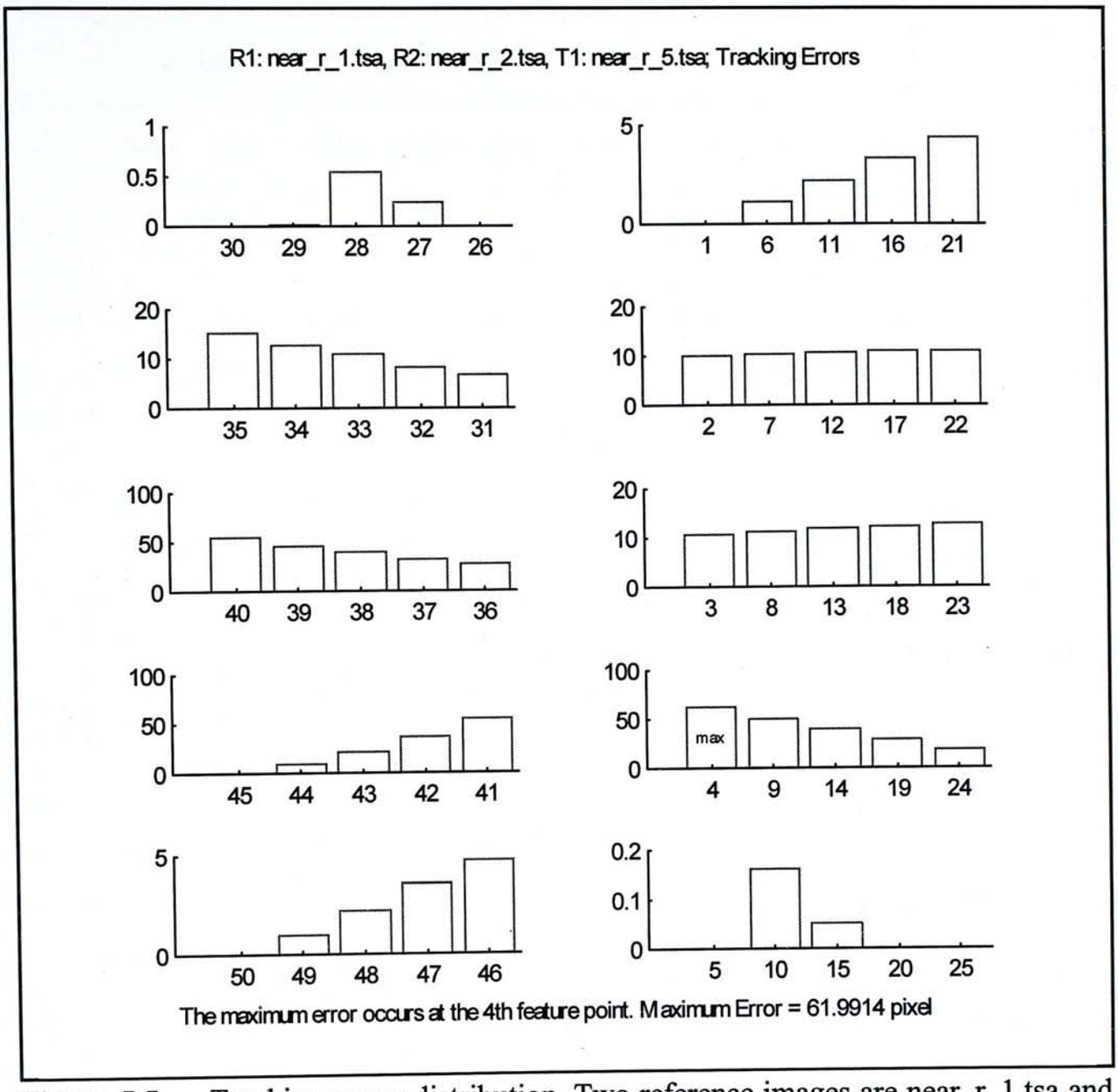


Figure 5.5 Tracking errors distribution. Two reference images are near_r_1.tsa and near_r_2.tsa. The target image is near_r_5.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

1.8e-010	0.0218	0.5524	0.2468	2.5e-010	3.0e-010	1.1024	2.1893	3.3222	4.3769
30	29	28	27	26	1	6	11	16	21
15.1126	12.7153	11.0389	8.3331	6.6122	10.1271	10.3199	10.6361	10.8068	10.9524
35	34	33	32	31	2	7	12	17	22
54.6967	45.4109	38.9496	32.4834	27.6202	10.6564	11.1049	11.9204	12.2184	12.5779
40	39	38	37	36	3	8	13	18	23
1.1e-009	9.7047	21.8873	36.8702	55.2363	61.9914	49.9458	38.8724	28.2083	18.4505
45	44	43	42	41	4	9	14	19	24
4.7e-010	1.0153	2.2129	3.5618	4.6839	1.0e-009	0.1613	0.0502	1.2e-009	1.3e-009
50	49	48	47	46	5	10	15	20	25

R1: near_r_1.tsa, R2: near_r_2.tsa, T1: near_r_5.tsa

The maximum error occurs at the 4th feature point. Maximum Error = 61.9914 pixel

Figure 5.6 Tracking errors distribution. Two reference images are near_r_1.tsa and near_r_2.tsa. The target image is near_r_5.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

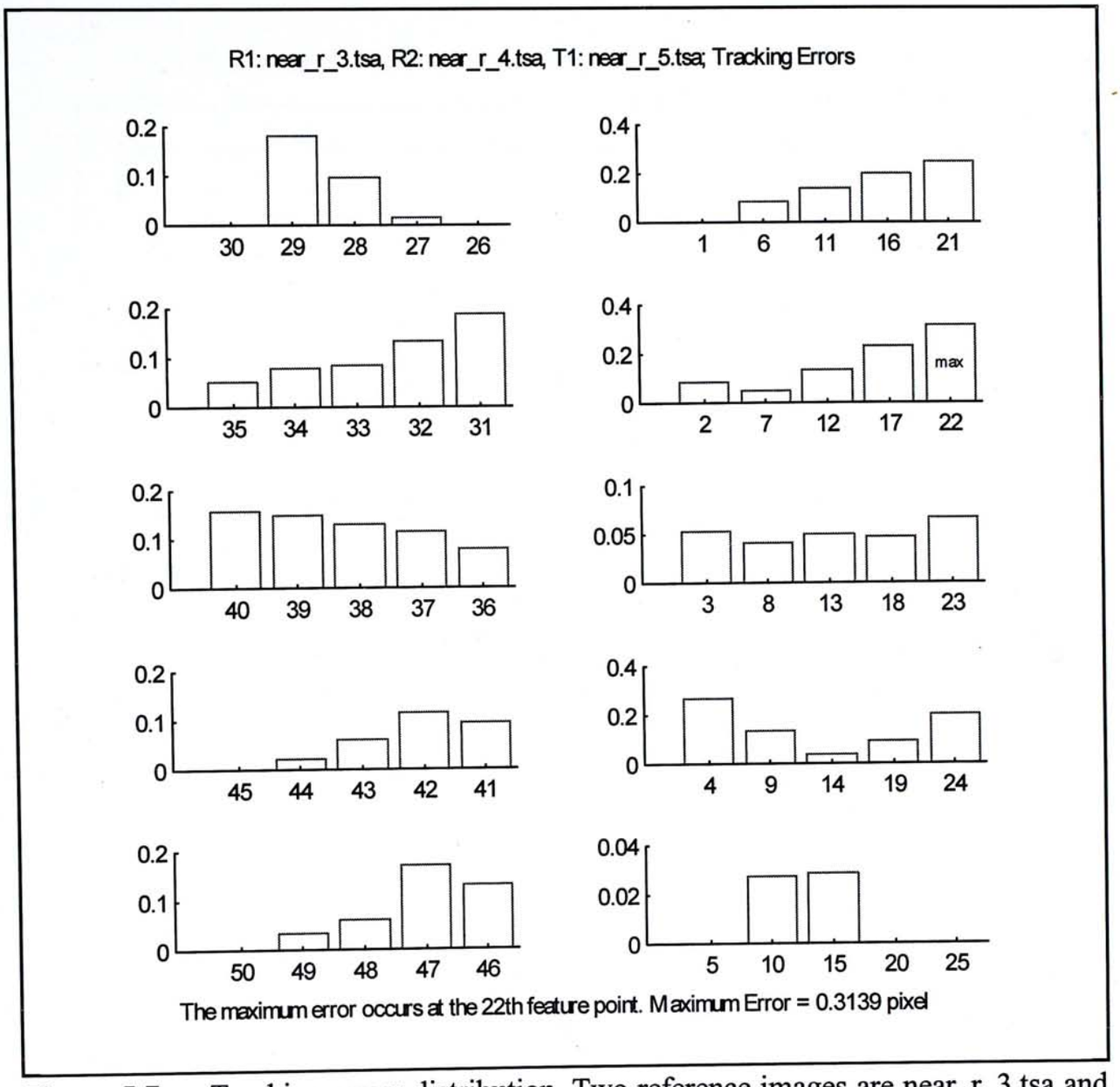


Figure 5.7 Tracking errors distribution. Two reference images are near_r_3.tsa and near_r_4.tsa. The target image is near_r_5.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

4.4e-011	0.1814	0.0960	0.0155	7.7e-011	9.5e-011	0.0851	0.1394	0.1997	0.2496
30	29	28	27	26	1	6	11	16	21
0.0518	0.0777	0.0841	0.1318	0.1878	0.0819	0.0468	0.1349	0.2306	0.3139
35	34	33	32	31	2	7	12	17	22
0.1588	0.1497	0.1298	0.1147	0.0798	0.0525	0.0408	0.0503	0.0477	0.0665
40	39	38	37	36	3	8	13	18	23
6.7e-011	0.0217	0.0595	0.1138	0.0928	0.2686	0.1315	0.0383	0.0938	0.1998
45	44	43	42	41	4	9	14	19	24
7.4e-011	0.0330	0.0608	0.1706	0.1312	1.7e-010	0.0276	0.0282	2.0e-010	2.1e-010
50	49	48	47	46	5	10	15	20	25

R1: near_r_3.tsa, R2: near_r_4.tsa, T1: near_r_5.tsa

The maximum error occurs at the 22th feature point. Maximum Error = 0.3139 pixel

Figure 5.8 Tracking errors distribution. Two reference images are near_r_3.tsa and near_r_4.tsa. The target image is near_r_5.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

5.3.2 Experiment 2. Real Image Sequence 2 of a Camera Calibration Block

In this experiment , we test the transfer algorithm on the image sequence ar_1.img .... ar_5.img as shown in Fig 2.6. The distance separating the camera and object is about 1.5 metres. The focal length of the lens is 16mm.The input values of robot joint variable to the active vision system and corresponding image frames are as follows:

Position	***X*** (mm)	***Y*** (mm)	***D*** (°)	***E*** (°)	Frame
1	0.000	0.000	0.000	0.000	ar_1.img
2	-140.000	-98.000	10.000	0.000	ar_2.img
3	-140.000	-98.000	10.000	6.500	ar_3.img
4	140.000	-98.000	-8.000	5.000	ar_4.img
5	0.000	-98.000	0.000	7.000	ar_5.img

Table 5.3 Input values and corresponding image frames

Nine triple of image frames are tested. The maximum error between transferred point and original point in the target frame is reported in Table 5.4 for each triple of image frames. Three examples of distributions of the tracking errors across feature points on the target frames are illustrated in Figure 5.9 - Figure 5.14, in both graphical and digital form.

	First_Ref	Second_Ref	Target_Ref	Maximum_error(pix)
1	ar_1.img	ar_2.img	ar_3.img	39.7102
2	ar_1.img	ar_2.img	ar_4.img	3.8378
3	ar_1.img	ar_2.img	ar_5.img	1.3363
4	ar_2.img	ar_3.img	ar_4.img	9.8476
5	ar_2.img	ar_3.img	ar_5.img	4.7709
6	ar_3.img	ar_4.img	ar_5.img	24.2157
7	ar_1.img	ar_5.img	ar_2.img	4.2796
8	ar_1.img	ar_5.img	ar_3.img	4.5489
9	ar_1.img	ar_5.img	ar_4.img	4.8800

Table 5.4 Maximum errors in pixels between the tracked feature points in the target frame and the original feature points in the target frame

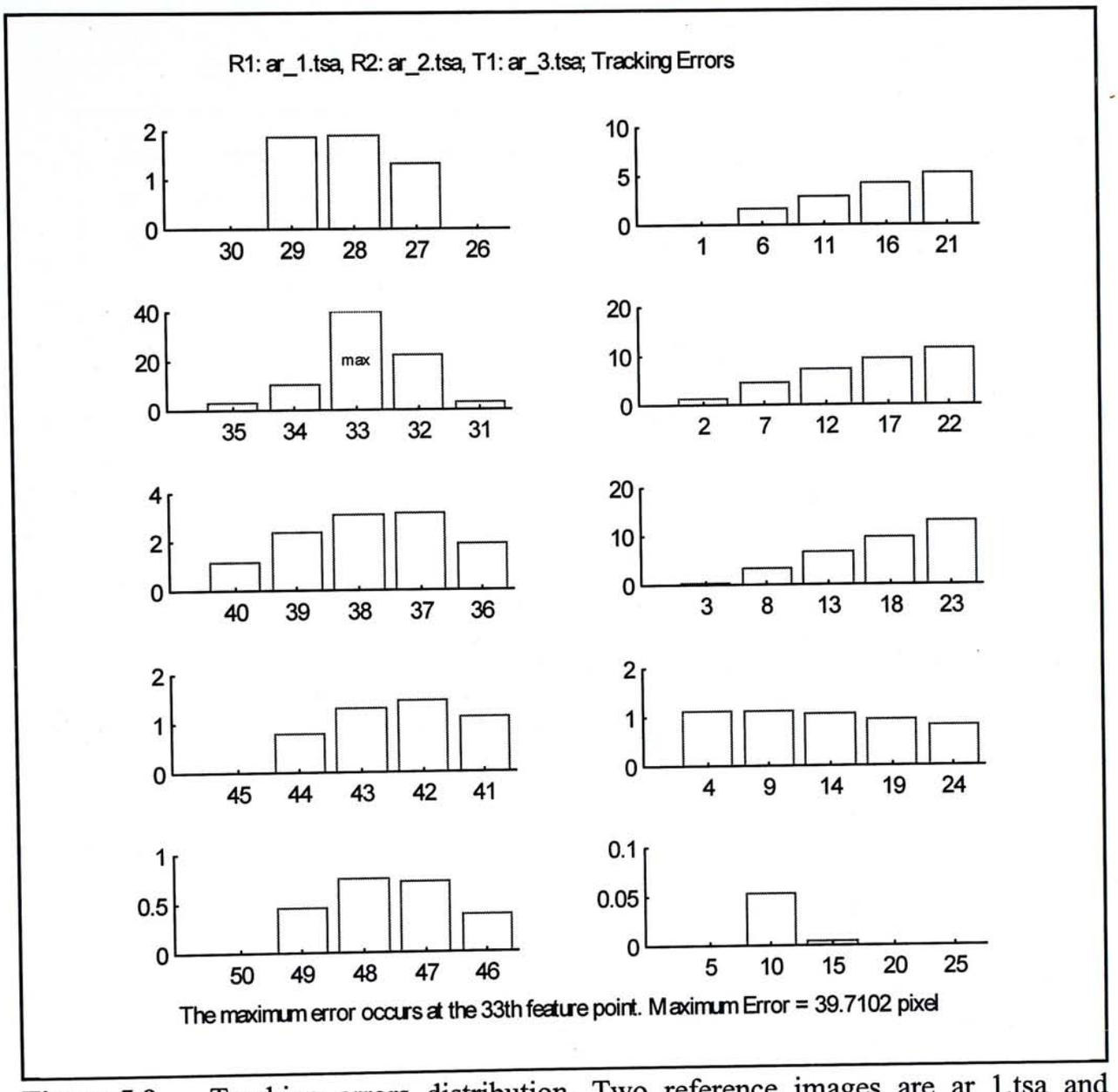


Figure 5.9 Tracking errors distribution. Two reference images are ar_1.tsa and ar_2.tsa. The target image is ar_3.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

7.3e-011	1.8879	1.9231	1.3218	8.5e-011	9.5e-011	1.6094	2.8596	4.2296	5.3181
30	29	28	27	26	1	6	11	16	21
3.1847	10.5187	39.7102	22.4893	3.0111	1.1962	4.5468	7.1795	9.5249	11.3895
35	34	33	32	31	2	7	12	17	22
1.1589	2.3897	3.0716	3.1727	1.9084	0.2896	3.3692	6.5245	9.7015	13.1210
40	39	38	37	36	3	8	13	18	23
3.2e-011	0.7932	1.2940	1.4552	1.1255	1.1103	1.1363	1.0726	0.9457	0.8260
45	44	43	42	41	4	9	14	19	24
2.6e-011	0.4590	0.7377	0.7159	0.3831	1.2e-010	0.0536	0.0046	1.6e-010	1.7e-010
50	49	48	47	46	5	10	15	20	25

R1: ar_1.tsa, R2: ar_2.tsa, T1: ar_3.tsa

The maximum error occurs at the 33th feature point. Maximum Error = 39.7102 pixel

Figure 5.10 Tracking errors distribution. Two reference images are ar_1.tsa and ar_2.tsa. The target image is ar_3.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

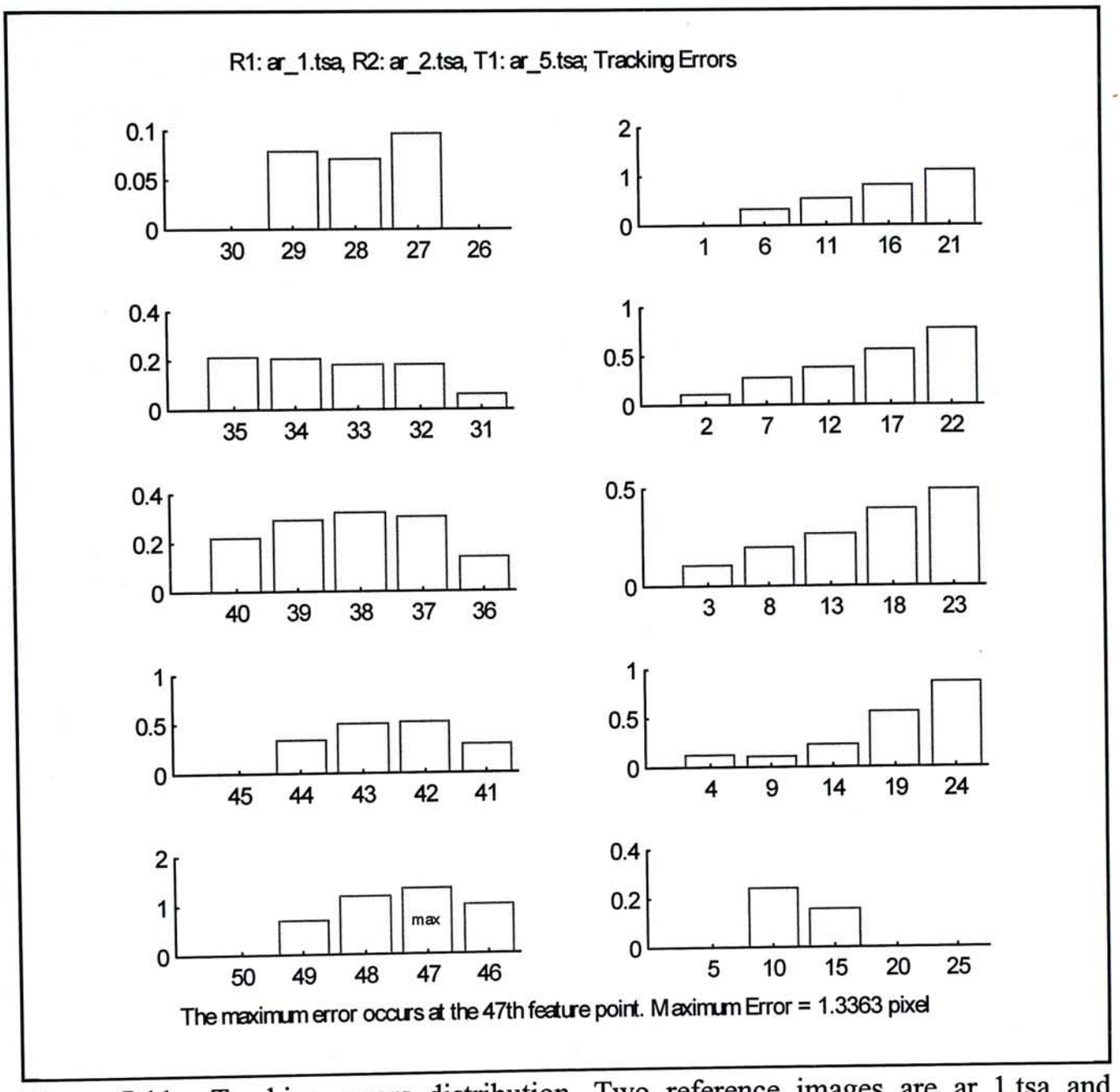


Figure 5.11 Tracking errors distribution. Two reference images are ar_1.tsa and ar_2.tsa. The target image is ar_5.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

1.3e-011	0.0790	0.0714	0.0972	1.9e-011	2.3e-011	0.3440	0.5437	0.8086	1.1319
30	29	28	27	26	1	6	11	16	21
0.2146	0.2079	0.1847	0.1805	0.0611	0.1129	0.2771	0.3785	0.5533	0.7700
35	34	33	32	31	2	7	12	17	22
0.2205	0.2927	0.3211	0.3036	0.1394	0.1079	0.1935	0.2632	0.3929	0.4909
40	39	38	37	36	3	8	13	18	23
6.2e-011	0.3258	0.5074	0.5170	0.2851	0.1183	0.1045	0.2278	0.5639	0.8654
45	44	43	42	41	4	9	14	19	24
1.8e-010	0.7013	1.1912	1.3363	1.0103	4.7e-010	0.2417	0.1578	7.0e-010	8.0e-010
50	49	48	47	46	5	10	15	20	25

R1: ar_1.tsa, R2: ar_2.tsa, T1: ar_5.tsa

The maximum error occurs at the 47th feature point. Maximum Error = 1.3363 pixel

Figure 5.12 Tracking errors distribution. Two reference images are ar_1.tsa and ar_2.tsa. The target image is ar_5.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

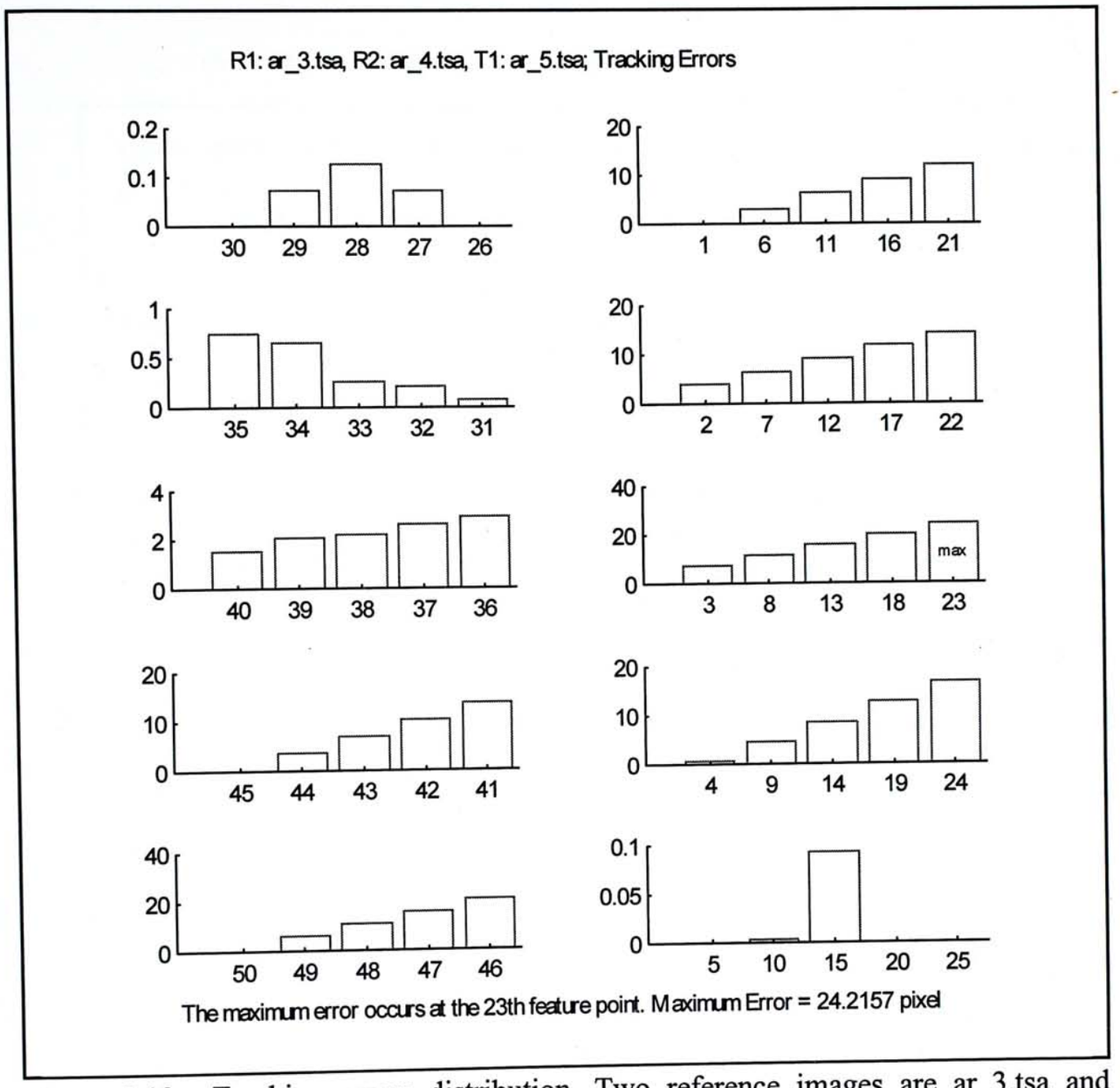


Figure 5.13 Tracking errors distribution. Two reference images are ar_3.tsa and ar_4.tsa. The target image is ar_5.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

1.6e-010	0.0722	0.1267	0.0719	1.7e-010	1.9e-010	3.0369	6.3578	9.0375	12.2355
30	29	28	27	26	1	6	11	16	21
0.7423	0.6572	0.2605	0.2055	0.0773	3.8061	6.4648	9.1904	11.6899	14.3735
35	34	33	32	31	2	7	12	17	22
1.5144	2.0318	2.1677	2.5846	2.9201	7.1436	11.5019	15.6988	20.0307	24.2157
40	39	38	37	36	3	8	13	18	23
7.0e-010	3.7514	6.9419	10.3264	13.6658	0.5265	4.6761	8.5000	12.7485	16.5869
45	44	43	42	41	4	9	14	19	24
2.4e-009	5.9112	10.6561	15.5718	20.5288	2.8e-009	0.0037	0.0922	3.3e-009	3.4e-009
50	49	48	47	46	5	10	15	20	25

R1: ar_3.tsa, R2: ar_4.tsa, T1: ar_5.tsa

The maximum error occurs at the 23th feature point. Maximum Error = 24.2157 pixel

Figure 5.14 Tracking errors distribution. Two reference images are ar_3.tsa and ar_4.tsa. The target image is ar_5.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

5.3.3 Experiment 3. Real Image Sequence 3 of a Camera Calibration Block

In this experiment , we test the transfer algorithm on the image sequence tu_1.img .... tu_6.img as shown in Fig 4.8. The distance separating the camera and object is about 3.0 metres.The focal length of the lens is 25mm. This situation approximated the orthographic projection.The calibration block was placed on a turn table. The rotaion angle ranges over 30 degrees with approximately 5 degrees as the increment of one step.

Fourteen triple of image frames are tested. The maximum error between transferred point and original point in the target frame is reported in Table 5.5 for each triple of image frames. Two examples of distributions of the tracking errors across feature points on the target frames are illustrated in Figure 5.15 - Figure 5.18, in both graphical and digital form.

	First_Ref	Second_Ref	Target_Ref	Maximum_error(pix)
1	tu_1.img	tu_2.img	tu_3.img	28.0475
2	tu_1.img	tu_2.img	tu_4.img	29.9092
3	tu_1.img	tu_2.img	tu_5.img	43.7398
4	tu_1.img	tu_2.img	tu_6.img	47.5909
5	tu_2.img	tu_3.img	tu_4.img	37.0246
6	tu_2.img	tu_3.img	tu_5.img	110.9894
7	tu_2.img	tu_3.img	tu_6.img	119.4898
8	tu_3.img	tu_4.img	tu_5.img	68.3772
9	tu_3.img	tu_4.img	tu_6.img	132.4895
10	tu_4.img	tu_5.img	tu_6.img	10579.0000
11	tu_1.img	tu_6.img	tu_2.img	9.5465
12	tu_1.img	tu_6.img	tu_3.img	139.9666
13	tu_1.img	tu_6.img	tu_4.img	37.8864
14	tu_1.img	tu_6.img	tu_5.img	54.0567

Table 5.5 Maximum errors in pixels between the tracked feature points in the target frame and the original feature points in the target frame

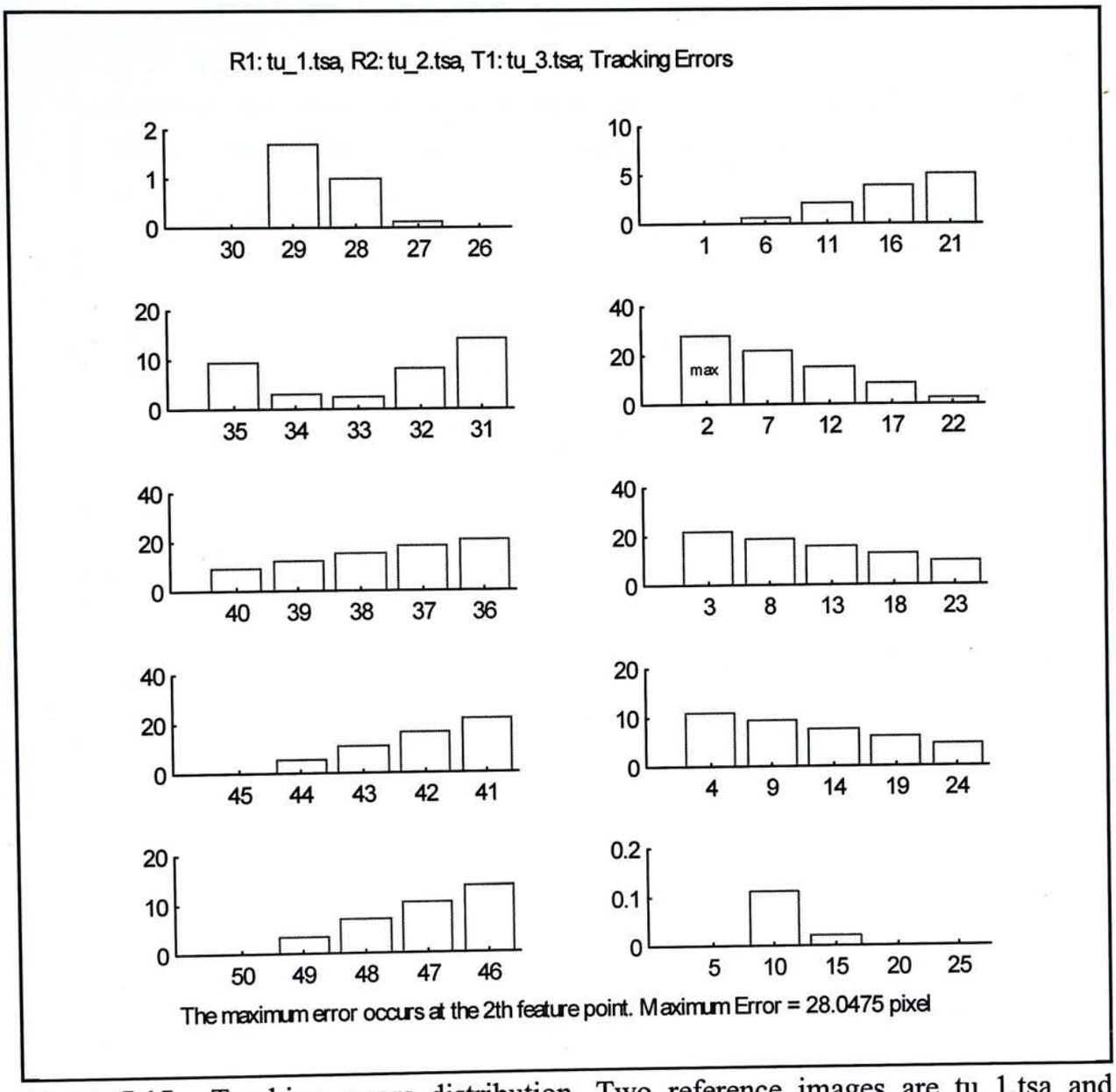


Figure 5.15 Tracking errors distribution. Two reference images are tu_1.tsa and tu_2.tsa. The target image is tu_3.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

3.0e-010	1.7096	1.0066	0.1148	3.9e-010	4.6e-010	0.6025	2.0483	3.8831	5.1913
30	29	28	27	26	1	6	11	16	21
9.3713	3.0231	2.5668	8.2122	14.2616	28.0475	21.7283	15.2737	8.7638	2.5673
35	34	33	32	31	2	7	12	17	22
9.1964	12.2607	15.0963	17.9716	20.7907	21.6826	18.6388	15.5681	12.4599	9.4330
40	39	38	37	36	3	8	13	18	23
2.5e-010	5.4929	10.9860	16.5782	22.0785	10.8279	9.2970	7.6991	6.1231	4.5627
45	44	43	42	41	4	9	14	19	24
2.3e-010	3.4729	6.8649	10.2718	13.7085	3.6e-010	0.1109	0.0219	4.3e-010	4.5e-010
50	49	48	47	46	5	10	15	20	25

R1: tu_1.tsa, R2: tu_2.tsa, T1: tu_3.tsa

The maximum error occurs at the 2th feature point. Maximum Error = 28.0475 pixel

Figure 5.16 Tracking errors distribution. Two reference images are tu_1.tsa and tu_2.tsa. The target image is tu_3.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

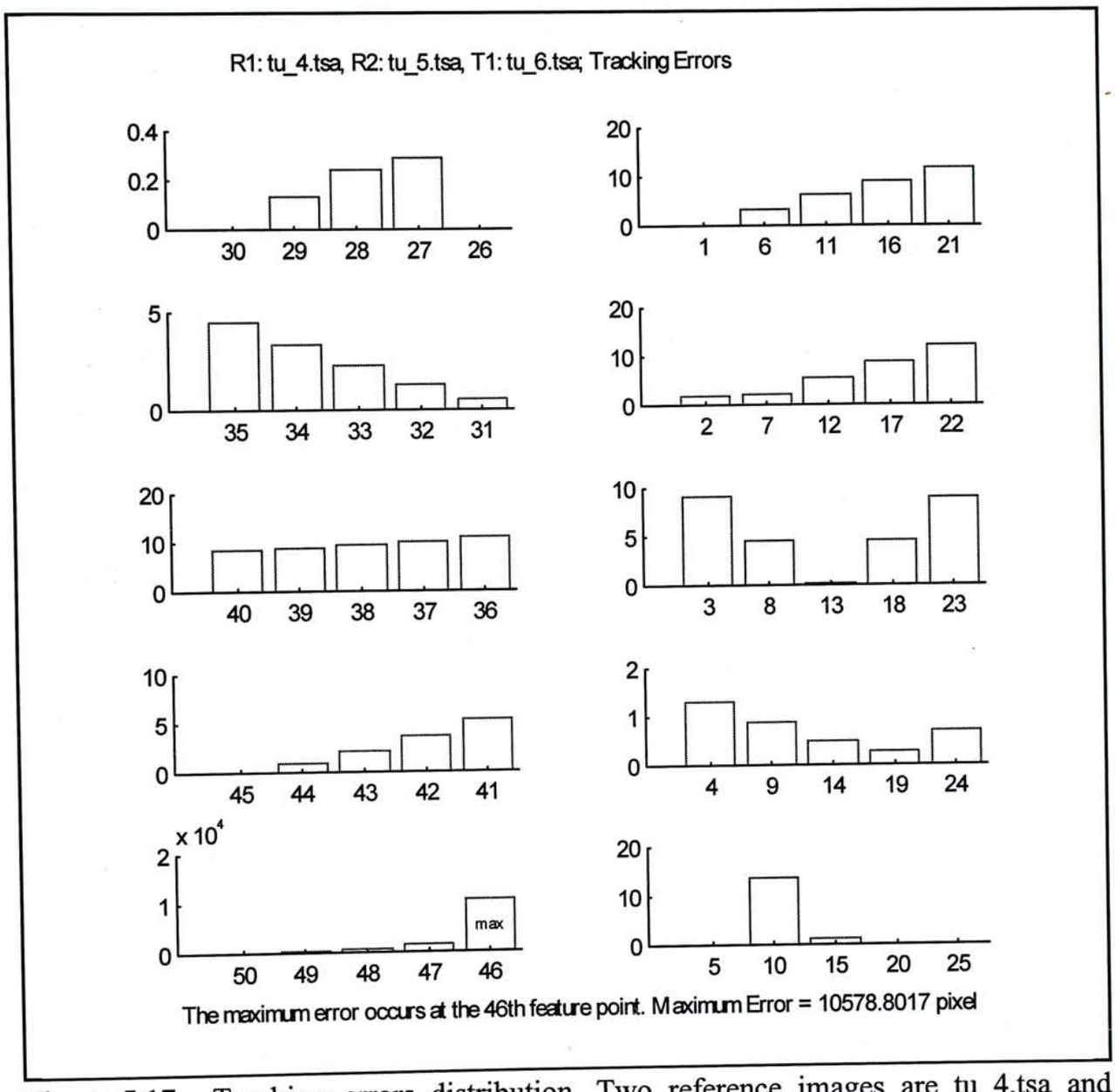


Figure 5.17 Tracking errors distribution. Two reference images are tu_4.tsa and tu_5.tsa. The target image is tu_6.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

3.0e-011	0.1333	0.2433	0.2892	4.5e-011	5.3e-011	3.4099	6.2526	9.1713	11.7804
30	29	28	27	26	1	6	11	16	21
4.4878	3.3547	2.2974	1.2699	0.5108	1.7393	2.0073	5.4044	8.8389	12.0080
35	34	33	32	31	2	7	12	17	22
8.4625	8.8565	9.4460	10.1064	11.0201	9.1124	4.4909	0.2202	4.5771	8.8746
40	39	38	37	36	3	8	13	18	23
1.4e-010	0.8748	2.1481	3.6020	5.3670	1.2946	0.8713	0.4781	0.2851	0.6887
45	44	43	42	41	4	9	14	19	24
9.7e-009	209.3843	595.6720	1578.2247	10578.8017	5.3e-008	13.5400	1.2177	1.2e-008	9.9e-009
50	49	48	47	46	5	10	15	20	25

R1: tu_4.tsa, R2: tu_5.tsa, T1: tu_6.tsa

The maximum error occurs at the 46th feature point. Maximum Error = 10578.8017 pixel

Figure 5.18 Tracking errors distribution. Two reference images are tu_4.tsa and tu_5.tsa. The target image is tu_6.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

5.3.4 Experiment 4. Synthetic Image Sequence of a Camera Calibration Block

We list here a synthetic data_set for justifying our implementation of the algorithm. Employing R.Y. Tsai's RAC based camera calibration method, we obtained a set of intrinsic and extrinsic parameters for a camera mounted on an active vision system undergone a sequence of movements which included 5 stops. Without considering the radial distorsion, the 3D world coordinates are transformed in the camera coordinate frame by using the corresponding extrinsic parameters obtained at the right position at first, then the 3D camera coordinates are mapped onto the image plane by using the intrinsic parameters obtained at that position. The whole process of synthesizing the data can be viewd as a Euclidean transformation(R and T) followed by a perspective transformation (pin hole model) and finally completed by a affine transformation (scaling the coordinates on the image plane into the frame buffer coordinates).

Nine triple of image frames are tested on the synthetic image sequence prj_ar1.img .... prj_ar_5.img. The maximum error between transferred point and oringinal point in the target frame is reported in Table 5.6 for each triple of image frames. Two examples of distributions of the tracking errors across feature points on the target frames are illustrated in Figure 5.19 - Figure 5.22, in both graphical and digital form.The experimental results showed that, the showing of the algorithm is perfect when no radial distorsion is involved.

	First_Ref	Second_Ref	Target_Ref	Maximum_error(pix)
1	prj_ar1.img	prj_ar2.img	prj_ar3.img	5.1220e-08
2	prj_ar1.img	prj_ar2.img	prj_ar4.img	4.5851e-07
3	prj_ar1.img	prj_ar2.img	prj_ar5.img	2.1480e-07
4	prj_ar2.img	prj_ar3.img	prj_ar4.img	1.4029e-06
5	prj_ar2.img	prj_ar3.img	prj_ar5.img	7.1996e-07
6	prj_ar3.img	prj_ar4.img	prj_ar5.img	9.4219e-07
7	prj_ar1.img	prj_ar5.img	prj_ar2.img	9.2399e-08
8	prj_ar1.img	prj_ar5.img	prj_ar3.img	1.1328e-07
9	prj_ar1.img	prj_ar5.img	prj_ar4.img	1.3012e-07

Table 5.6 Maximum errors in pixels between the tracked feature points in the target frame and the original feature points in the target frame

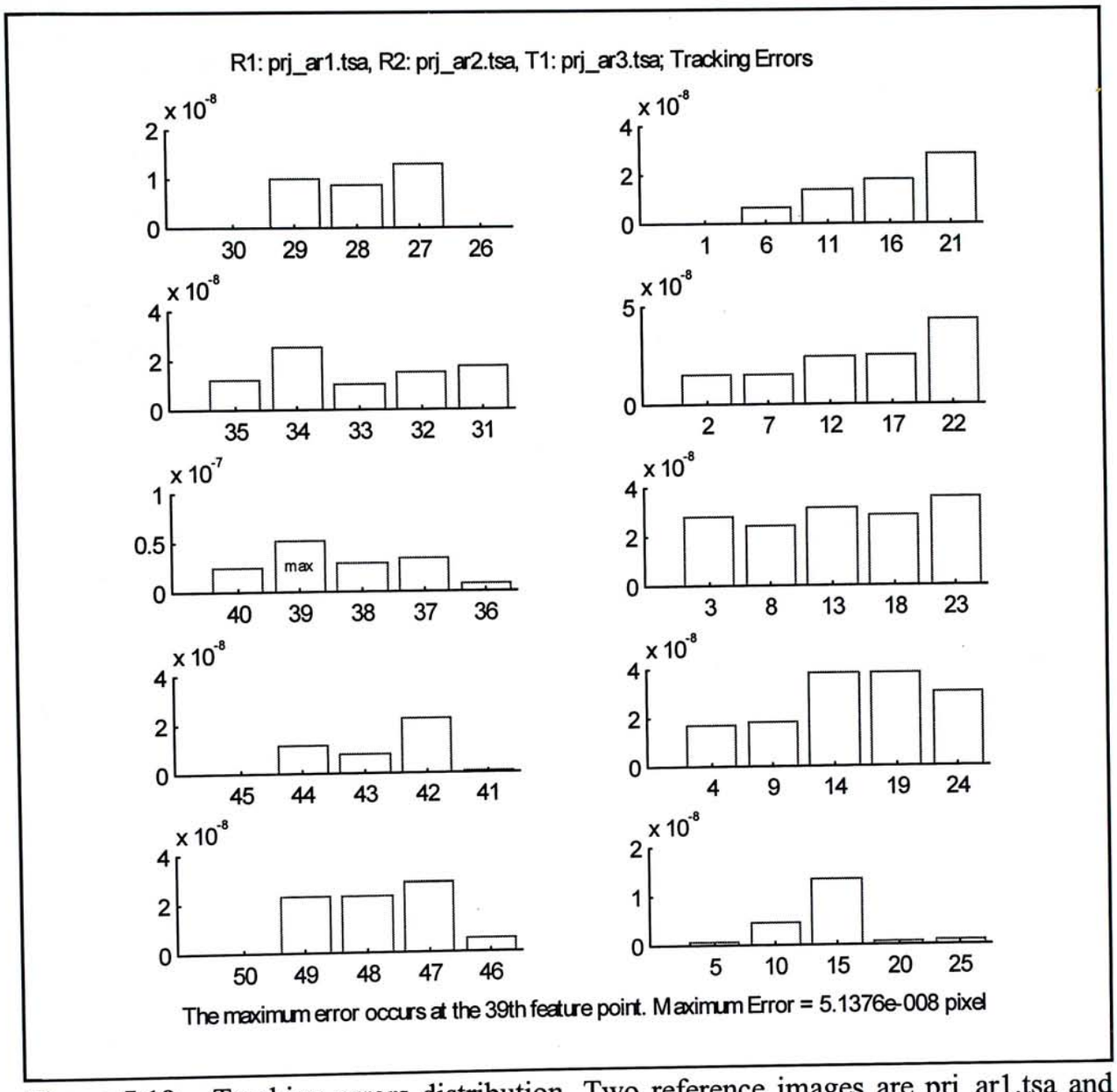


Figure 5.19 Tracking errors distribution. Two reference images are prj_ar1.tsa and prj_ar2.tsa. The target image is prj_ar3.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

4.0e-011	1.0e-008	8.8e-009	1.3e-008	1.2e-010	1.6e-010	6.7e-009	1.4e-008	1.8e-008	2.8e-008
30	29	28	27	26	1	6	11	16	21
1.2e-008	2.5e-008	1.0e-008	1.5e-008	1.7e-008	1.5e-008	1.5e-008	2.4e-008	2.5e-008	4.3e-008
35	34	33	32	31	2	7	12	17	22
2.4e-008	5.1e-008	2.8e-008	3.3e-008	8.2e-009	2.8e-008	2.4e-008	3.2e-008	2.8e-008	3.6e-008
40	39	38	37	36	3	8	13	18	23
9.8e-011	1.2e-008	7.7e-009	2.2e-008	7.2e-010	1.7e-008	1.8e-008	3.8e-008	3.8e-008	3.0e-008
45	44	43	42	41	4	9	14	19	24
1.4e-010	2.3e-008	2.3e-008	2.9e-008	5.7e-009	5.9e-010	4.5e-009	1.3e-008	7.3e-010	7.7e-010
50	49	48	47	46	5	10	15	20	25

R1: prj_ar1.tsa, R2: prj_ar2.tsa, T1: prj_ar3.tsa

The maximum error occurs at the 39th feature point. Maximum Error = 5.1376e-008 pixel

Figure 5.20 Tracking errors distribution. Two reference images are prj_ar1.tsa and prj_ar2.tsa. The target image is prj_ar3.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

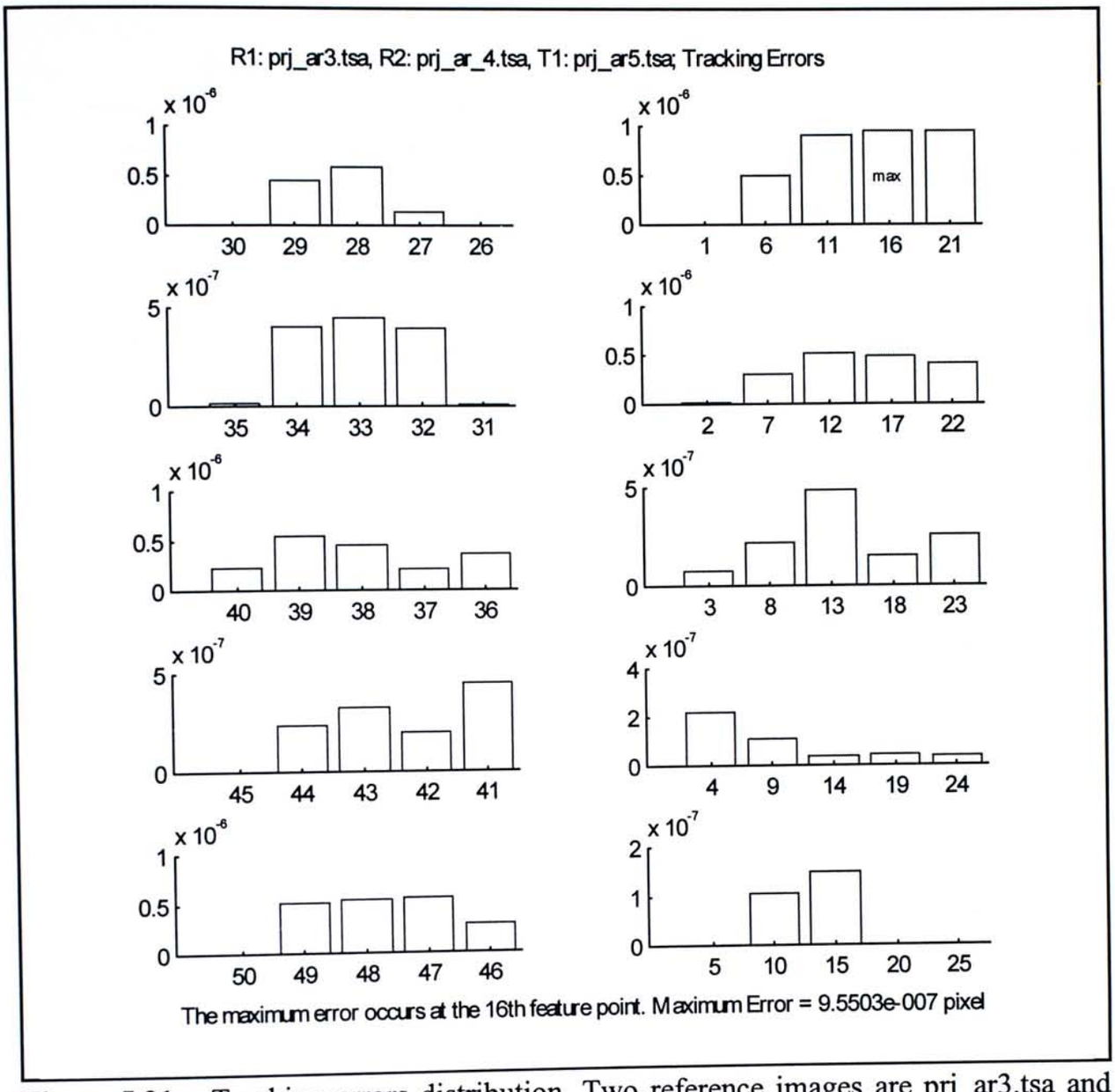


Figure 5.21 Tracking errors distribution. Two reference images are prj_ar3.tsa and prj_ar4.tsa. The target image is prj_ar5.tsa. The vertical axes of subplots represent the tracking errors in pixels. The horizontal axes of subplots represent the feature point indices.

Errors between Transferred Points and Original Points

1.6e-010	4.5e-007	5.9e-007	1.4e-007	2.2e-010	2.4e-010	4.9e-007	9.0e-007	9.6e-007	9.5e-007
30	29	28	27	26	1	6	11	16	21
1.7e-008	4.1e-007	4.5e-007	3.9e-007	7.7e-009	1.5e-008	3.0e-007	5.1e-007	4.9e-007	4.1e-007
35	34	33	32	31	2	7	12	17	22
2.3e-007	5.4e-007	4.5e-007	2.1e-007	3.7e-007	7.3e-008	2.2e-007	4.8e-007	1.5e-007	2.6e-007
40	39	38	37	36	3	8	13	18	23
1.4e-010	2.3e-007	3.3e-007	2.0e-007	4.5e-007	2.2e-007	1.1e-007	3.9e-008	4.2e-008	3.7e-008
45	44	43	42	41	4	9	14	19	24
1.3e-010	5.1e-007	5.5e-007	5.6e-007	2.8e-007	2.0e-010	1.1e-007	1.5e-007	2.4e-010	2.4e-010
50	49	48	47	46	5	10	15	20	25

R1: prj_ar3.tsa, R2: prj_ar_4.tsa, T1: prj_ar5.tsa

The maximum error occurs at the 16th feature point. Maximum Error = 9.5503e-007 pixel

Figure 5.24 Tracking errors distribution. Two reference images are prj_ar3.tsa and prj_ar4.tsa. The target image is prj_ar5.tsa. The unit of tracking errors is pixel. The integers in the figure represent the indices of feature points. The numbers in fixed point and with scientific notation represent the tracking errors.

5.3.5 Discussions on the Experimental Results

A conclusion can be drawn from the above-mentioned experimental results. It can be seen that this tracking program yielded quasi-acceptable results when it was applied in a situation where the perspective effects are the dominant one. Obviously, it demonstrated the worst performance in a case of orthographic projection (experiment 3).

One major cause for the error incured by the algorithm when it was applied on realistic image sequences might be the radial distorsion of the camera lense. It is well known that, the ideal pin hole model can be represented by a perspective transformation. But in real situation, the lens distorsion has to be taken into account so that the established camera model can fit well into the real situation.The results from experiment 4 show that, the performance of the transfer algorithm is perfect when no radial distorsion is involved.

5.4 Transfer Experiments on the Image Sequence of a Human Face Model

Since all experiments in last section was performed on several different image sequences about a camera calibration block, object points are distributed on two orthogonal plane in 3D space. In order to test the transfer algorithm on a more general case or curved surface, an image sequence of a human face model was taken by the active vision system.This sequence composed of 14 frames. The focal length of lens is 12 mm. The distance separating the camera and object is about 1.16m.

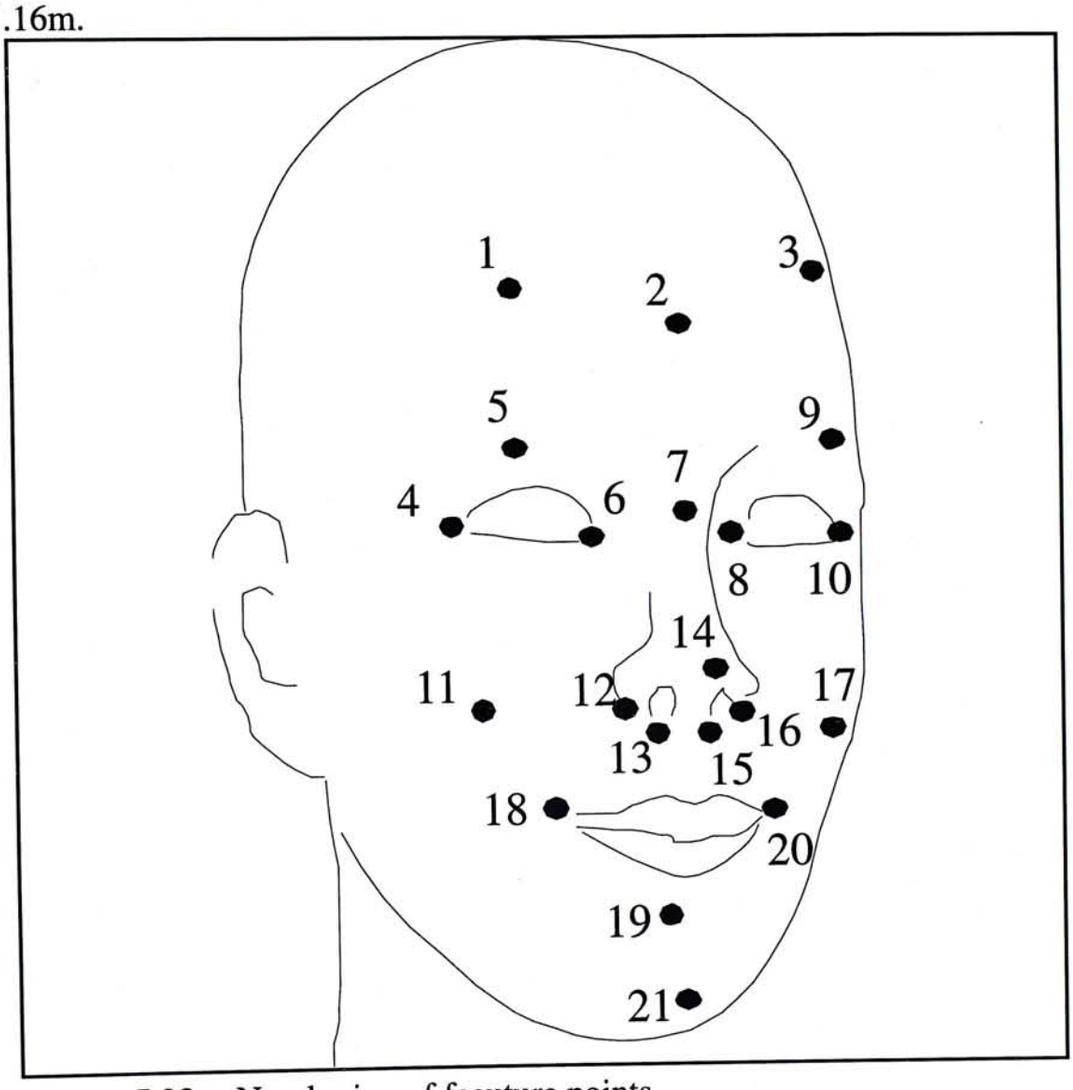


Figure 5.23 Numbering of feauture points

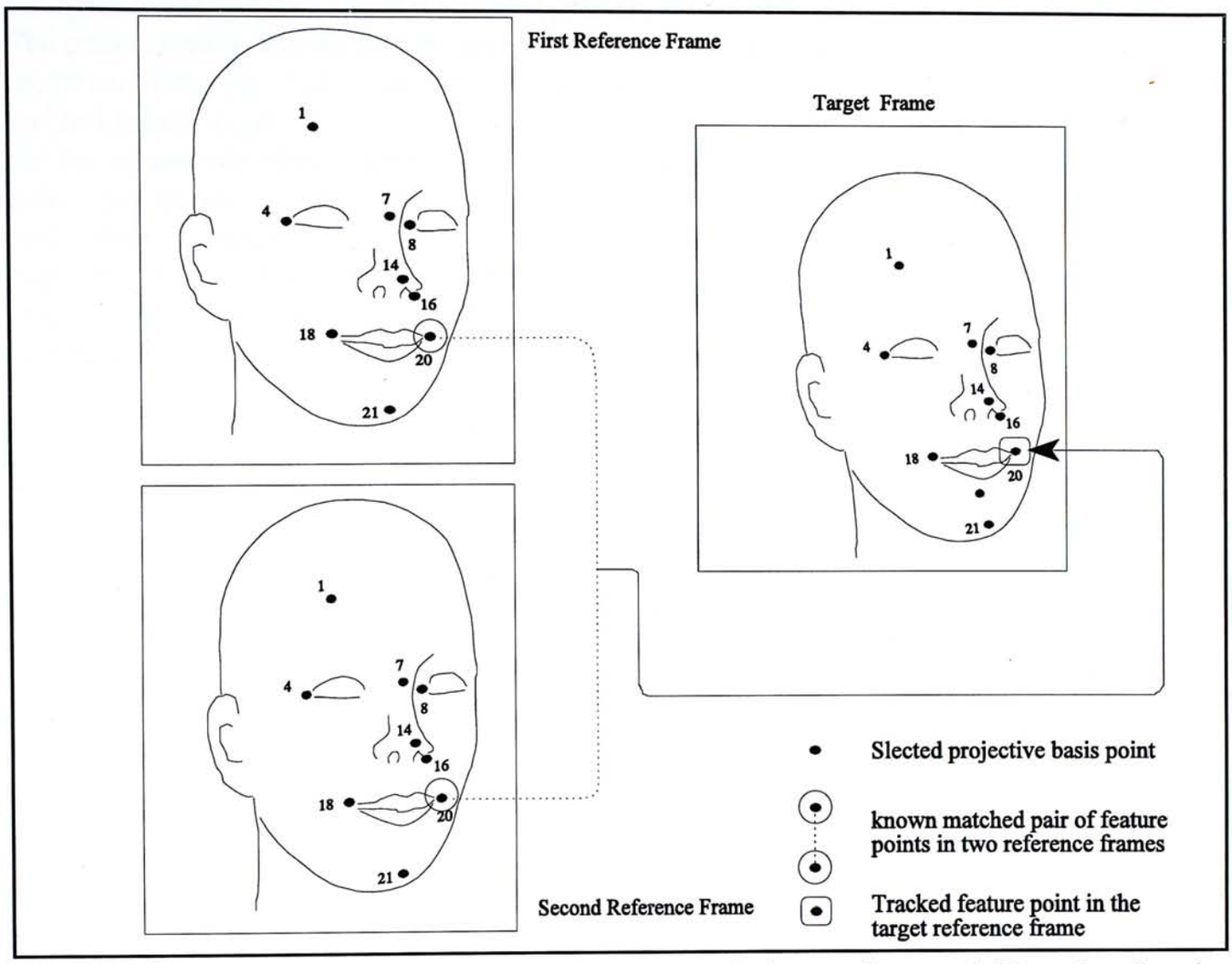


Figure 5.24 Tracking feature points on the surface of a human face model based on 8-point projective invariants.

In this experiment, the 8-point projective invariants based tracking algorithm are applied to an image sequence of a curved surface. There are 21 black ink dots scattered on the surface of a model of human face. The indices of feature points are illustated in Figure 5.23. The image coordinates of the image of those feature points are obtained using mouse on each frame through out the image sequence. A MATLAB program has been developed to implement the algorithm. In this program, one can arbitrarily select the 8 projective basis points as in Figure 5.24. The close dependency of tracking error on the geometry of projective basis points was shown clearly.

Here we have used two distinct 8-point projective basis set in the tracking process. Good results are yielded with one 8-point projective basis set and bad results are yielded with the other.

In the first experiment, we use eight feature point (1, 3, 4, 6, 8, 10, 14, 20) as the known 8-point correspondences across three image frames. The selected 8-point correspondences on the first reference frame and the second reference frame are shown on Fig 5.25 and Fig 5.26 respectively. The tracking result and the distribution of tracking error across feature points on the target frame are illustrated in Fig. 5.27 - Fig. 5.31. The maximum tracking error is 8.2678 pixel occurring at the 2nd feature point.

In the second experiment, we use eight feature point (1, 2, 4, 5, 7, 11, 18, 21) as the known 8-point correspondences across three image frames. The selected 8-point correspondences on the first reference frame and the second reference frame are shown on Fig 5.32 and Fig 5.33 respectively. The tracking result and the distribution of tracking error across feature points on the target frame are illustrated in Fig. 5.34 - Fig. 5.37. The maximum tracking error is 107.0571 pixel occurring at the 6th feature point.

Figure 5.25 The 8-point correspondences on the first reference frame.

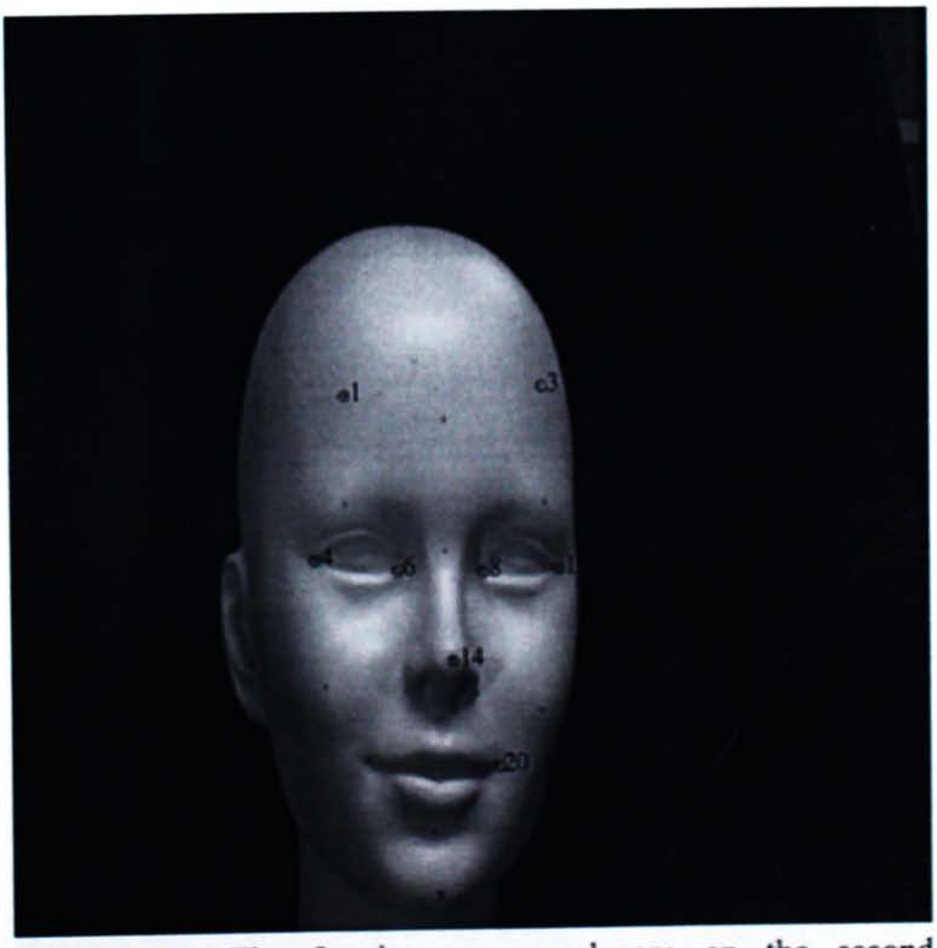


Figure 5.26 The 8-point correspondences on the second reference frame.

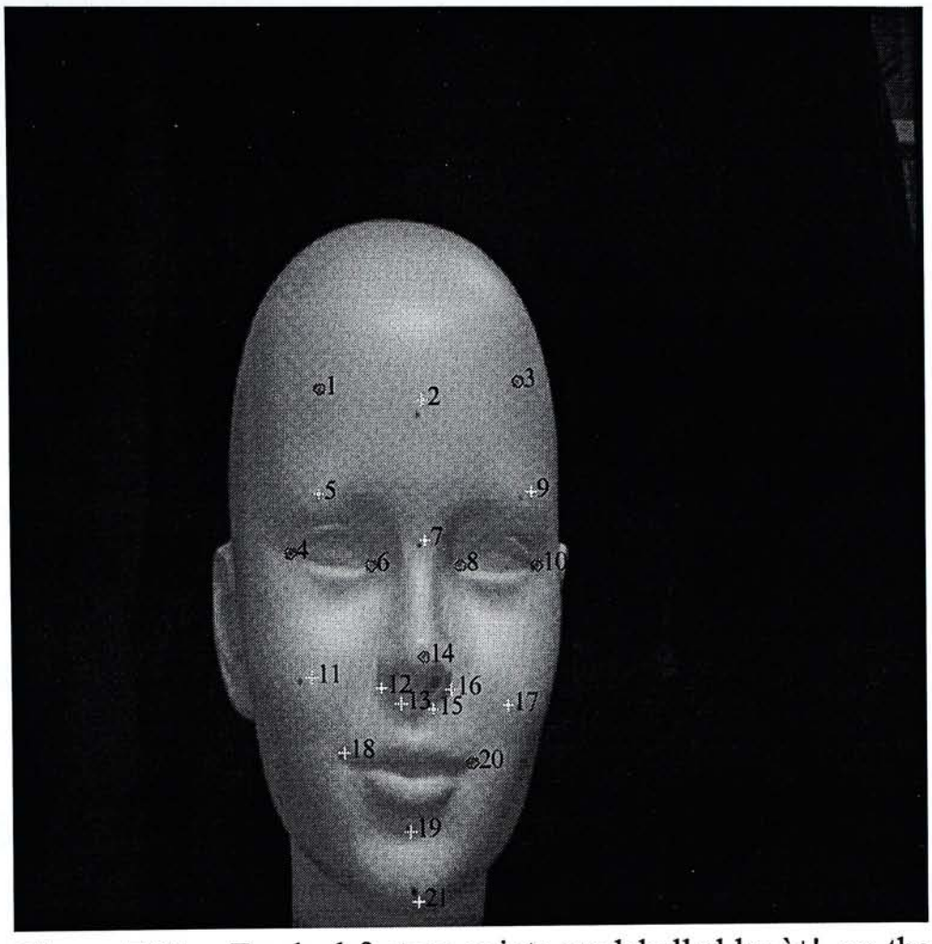


Figure 5.27 Tracked feature points are labelled by `+' on the target frame.

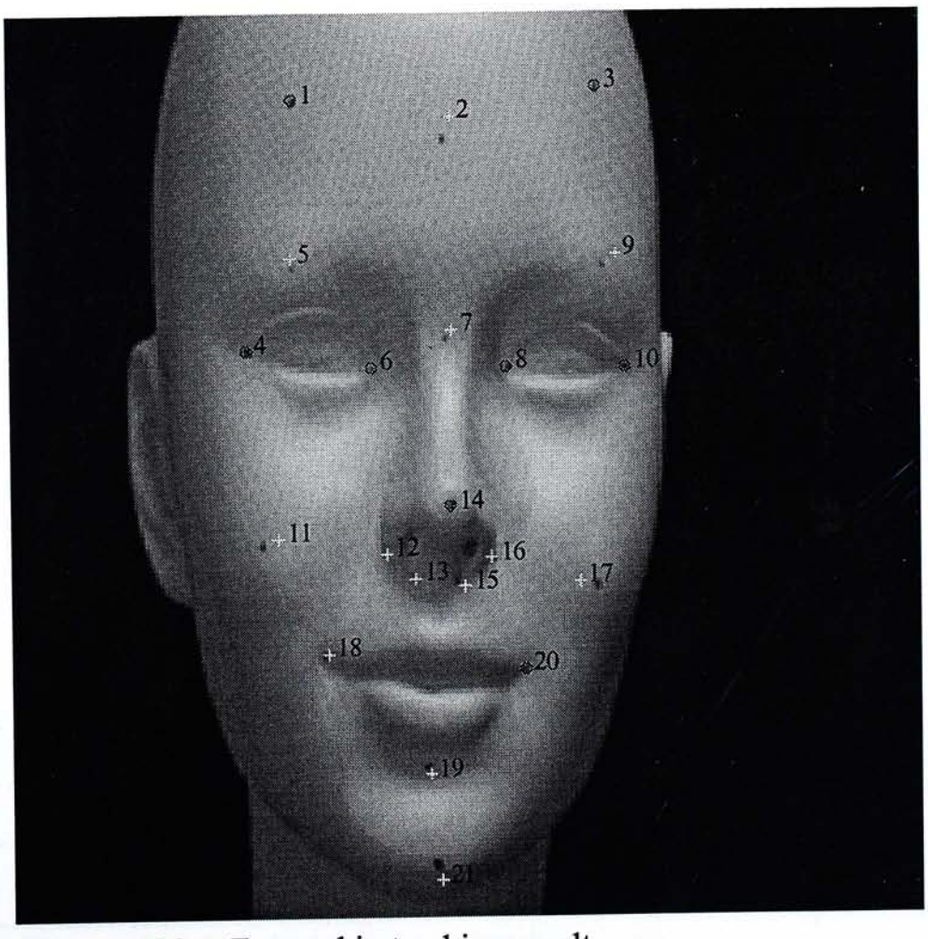


Figure 5.28 Zoomed in tracking result

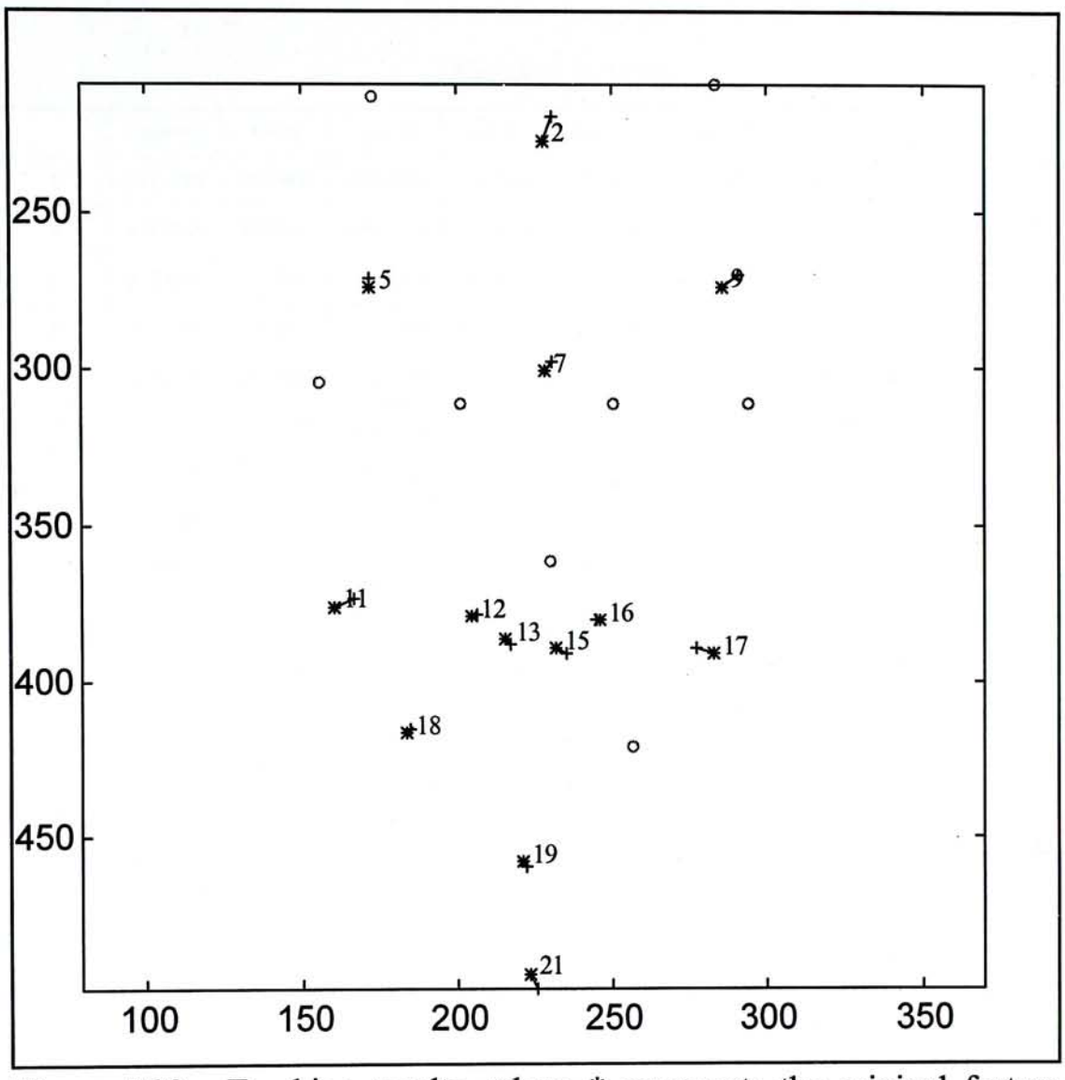


Figure 5.29 Tracking results, where * represents the original feature points in the target frame , + represents the tracked point.

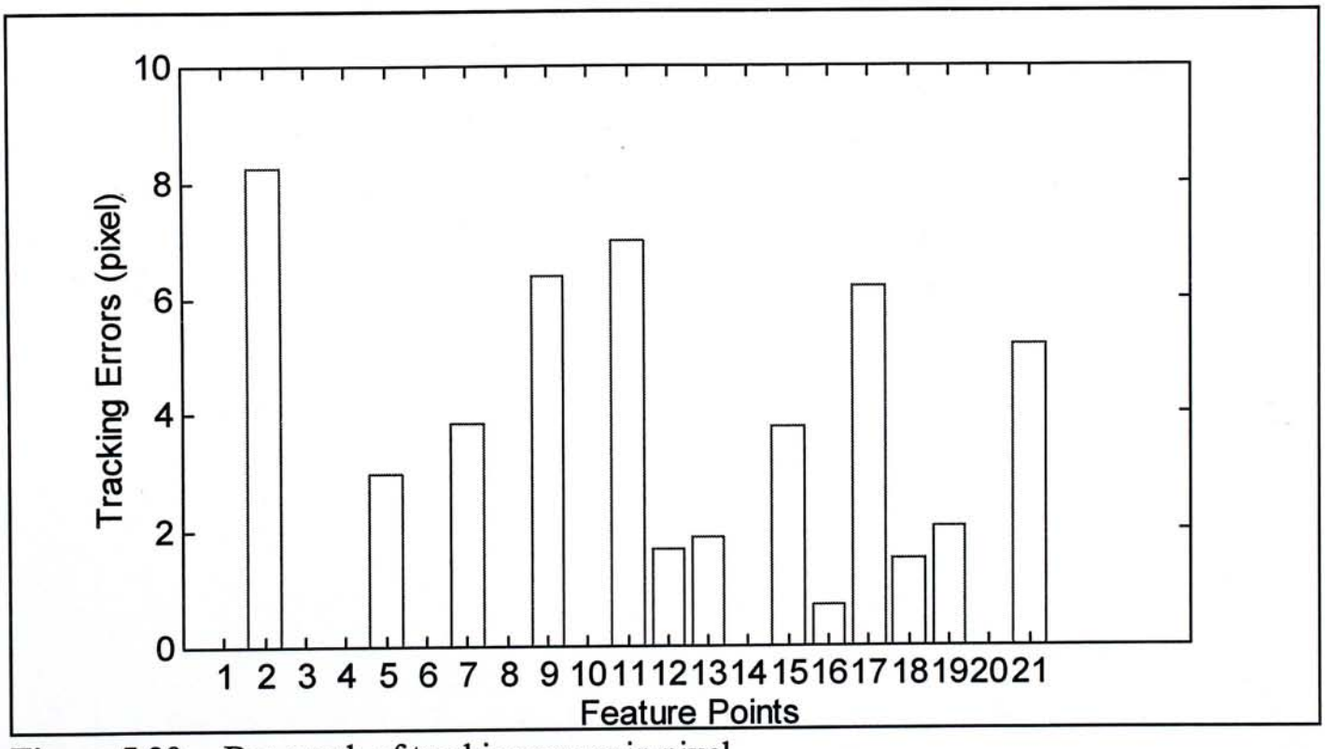


Figure 5.30 Bar graph of tracking errors in pixel.

Tracking Errors

	uref1	vref1	uref2	vref2	utarget	vtarget	utracked	vtracked	trk_error
1	222.0280	215.5827	186.3366	213.7457	172.9385	213.1946	172.9385	213.1946	0.0000
2	278.0896	229.6909	242.0899	227.5905	227.7201	227.1657	230.3485	219.3268	8.2678
3	321.5281	210.7562	295.5981	209.6297	283.6046	209.1503	283.6046	209.1503	0.0000
4	203.4646	307.6573	168.7500	305.0464	155.6584	304.3747	155.6584	304.3747	0.0000
5	224.2557	276.4708	186.3366	274.3634	171.8355	273.8588	171.7383	270.8894	2.9710
6	248.0169	312.8551	214.7745	311.4075	201.2485	310.9926	201.2485	310.9926	0.0000
7	278.0896	302.0883	242.4641	300.9304	228.0878	300.3304	230.5521	297.3874	3.8385
8	292.9404	312.1126	262.6699	311.7817	250.5151	310.9926	250.5151	310.9926	0.0000
9	327.0971	274.2431	298.2174	274.3634	285.4429	273.4911	290.6875	269.8517	6.3836
10	328.5822	310.9988	304.9526	311.4075	294.2668	311.3602	294.2668	311.3602	0.0000
11	213.8601	379.3122	174.7369	376.8896	160.4380	376.0686	166.8573	373.2780	6.9996
12	257.2986	381.1686	219.2647	380.2572	204.5574	379.3776	206.1388	378.8501	1.6670
13	269.1792	388.5939	230.1160	388.1151	215.5873	386.7308	217.0059	387.9916	1.8979
14	288.1139	363.7189	246.2059	362.6706	229.9261	361.7299	229.9261	361.7299	0.0000
15	285.1437	390.8215	247.3284	389.9860	231.7644	389.3045	235.1283	391.0436	3.7869
16	294.0542	381.1686	259.6765	381.3798	245.7355	380.4806	245.0378	380.5332	0.6997
17	325.2408	389.7077	295.9723	391.4827	282.8693	391.1428	276.9171	389.4529	6.1874
18	236.5075	419.4092	197.9363	417.3014	183.6007	416.5114	184.7173	415.5152	1.4964
19	272.5206	458.0212	234.9804	458.0873	220.7345	458.0572	221.8840	459.7620	2.0562
20	303.3359	420.1518	270.5278	421.7916	256.7654	421.2910	256.7654	421.2910	0.0000
21	276.6046	494.0343	238.3480	494.0089	223.3082	494.0880	225.5860	498.7657	5.2029

Figure 5.31 (uref1, vref1) and (uref2,vref2) are the image coordinates of feature points in the two reference frames. (utarget,vtarget) and (utracked,vtracked) are the original and tracked image coordinates of feature points in the target frame. trk_error is the tracking error between original feature points and tracked feature point in the target frame.

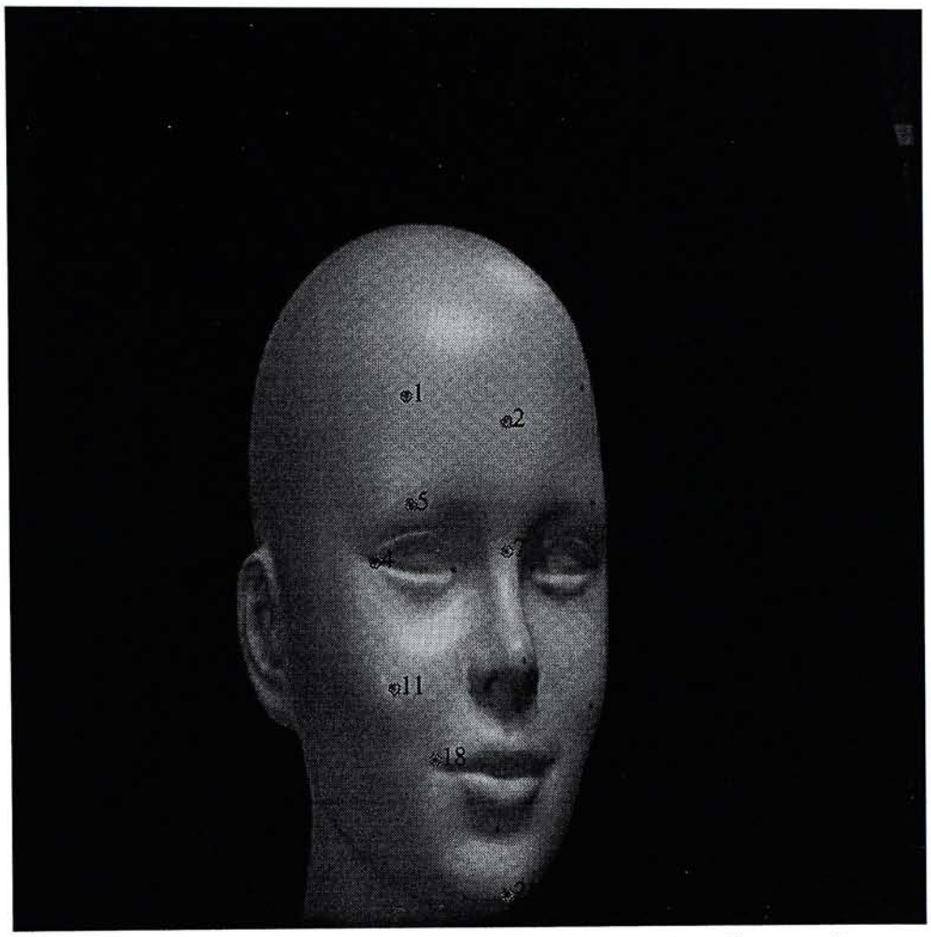


Figure 5.32 8-point correspondences on the first reference frame.

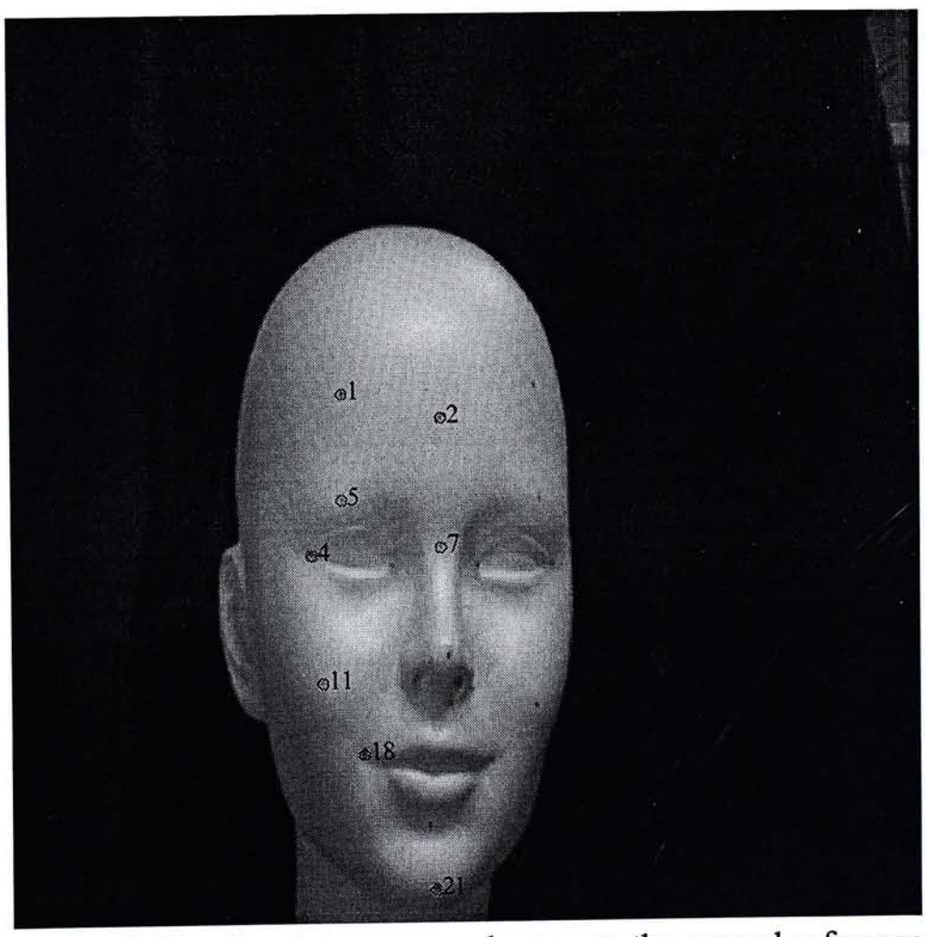


Figure 5.33 8-point correspondences on the second reference frame.

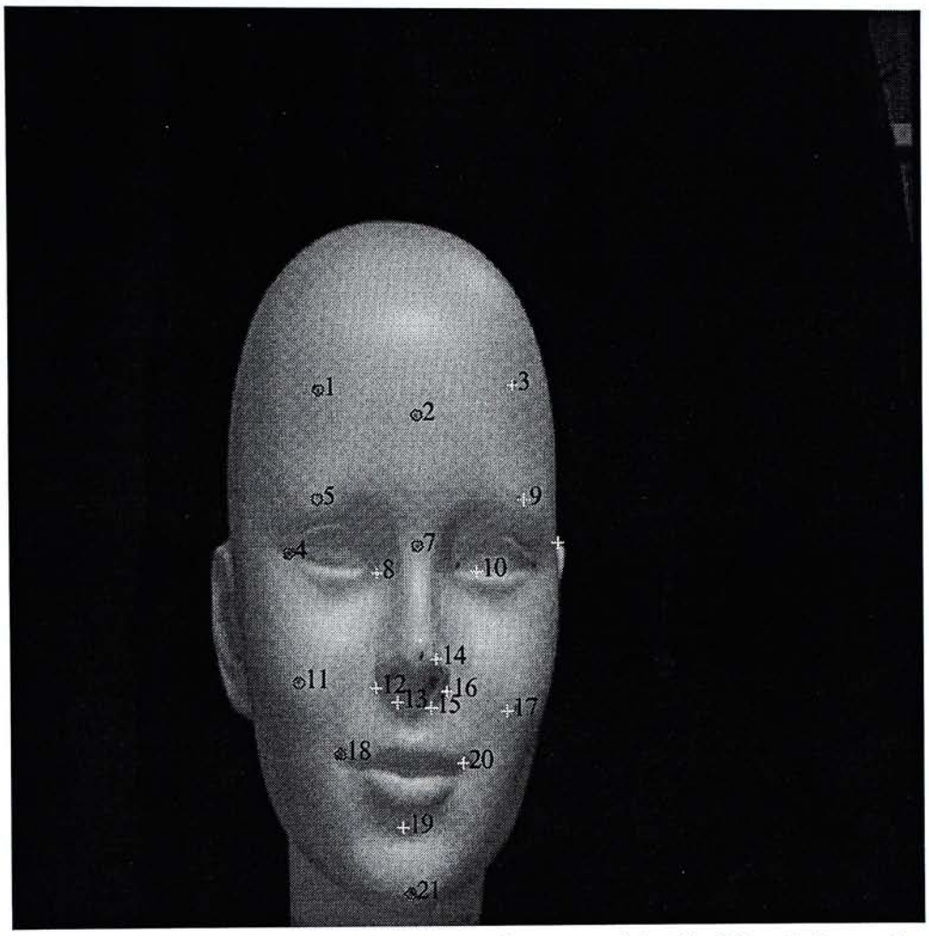


Figure 5.34 Tracked feature points are labelled by `+' on the target frame.

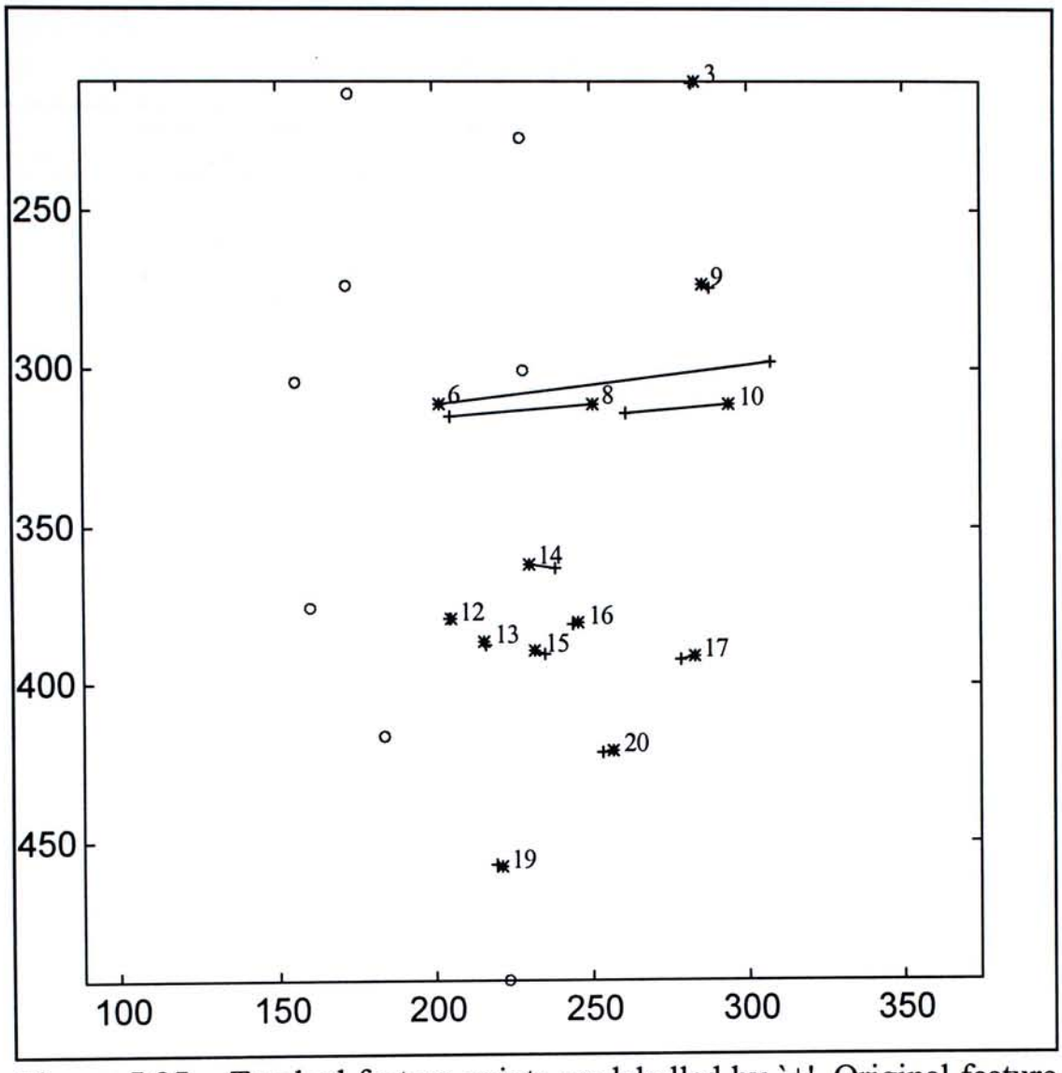


Figure 5.35 Tracked feature points are labelled by `+'. Original feature points are labelled by *.

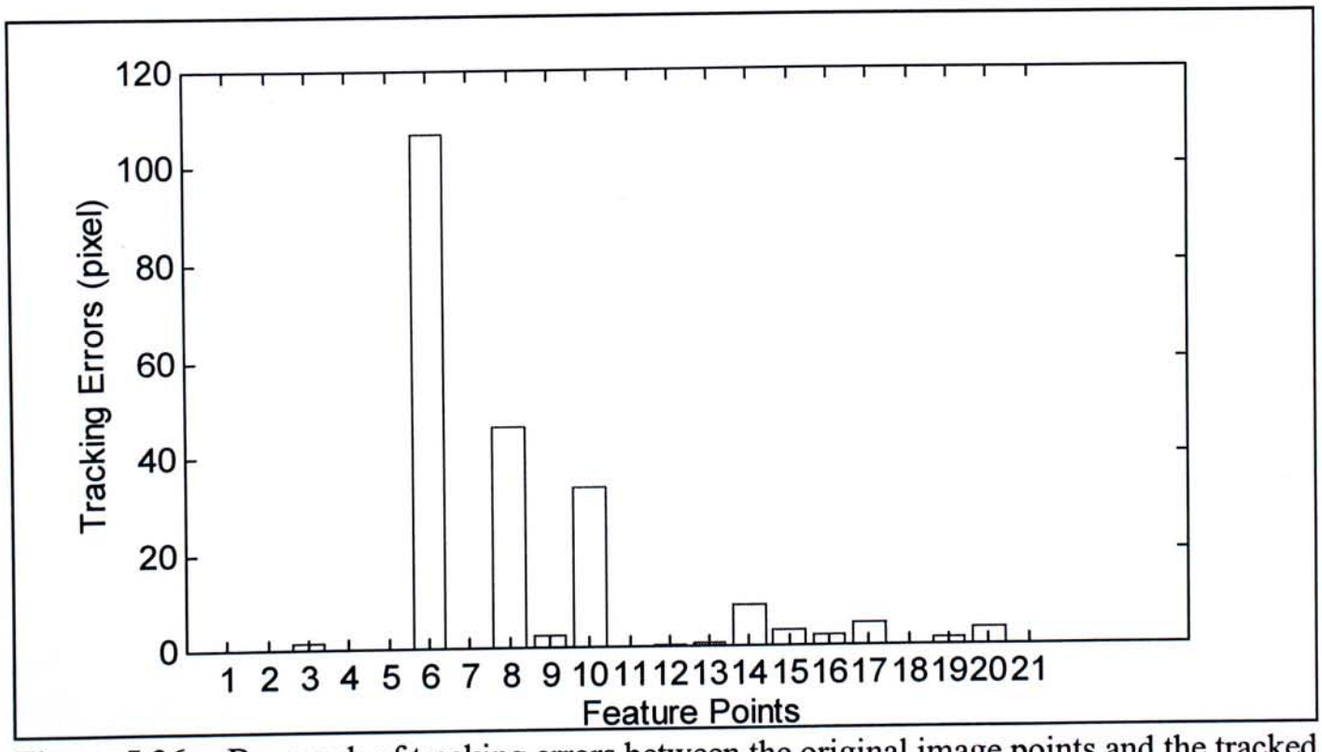


Figure 5.36 Bar graph of tracking errors between the original image points and the tracked feature points in the target frame.

Tracking Errors

	uref1	vref1	uref2	vref2	utarget	vtarget	utracked	vtracked	trk_error
1	222.0280	215.5827	186.3366	213.7457	172.9385	213.1946	172.9385	213.1946	0.0000
2	278.0896	229.6909	242.0899	227.5905	227.7201	227.1657	227.7201	227.1657	0.0000
3	321.5281	210.7562	295.5981	209.6297	283.6046	209.1503	282.2679	209.6476	1.4262
4	203.4646	307.6573	168.7500	305.0464	155.6584	304.3747	155.6584	304.3747	0.0000
5	224.2557	276.4708	186.3366	274.3634	171.8355	273.8588	171.8355	273.8588	0.0000
6	248.0169	312.8551	214.7745	311.4075	201.2485	310.9926	307.5124	297.9854	107.0571
7	278.0896	302.0883	242.4641	300.9304	228.0878	300.3304	228.0878	300.3304	0.0000
8	292.9404	312.1126	262.6699	311.7817	250.5151	310.9926	204.8504	315.2327	45.8611
9	327.0971	274.2431	298.2174	274.3634	285.4429	273.4911	287.7499	274.1418	2.3970
10	328.5822	310.9988	304.9526	311.4075	294.2668	311.3602	261.2619	314.2182	33.1285
11	213.8601	379.3122	174.7369	376.8896	160.4380	376.0686	160.4380	376.0686	0.0000
12	257.2986	381.1686	219.2647	380.2572	204.5574	379.3776	204.3083	379.3672	0.2493
13	269.1792	388.5939	230.1160	388.1151	215.5873	386.7308	215.8488	387.5584	0.8679
14	288.1139	363.7189	246.2059	362.6706	229.9261	361.7299	238.2635	363.0014	8.4338
15	285.1437	390.8215	247.3284	389.9860	231.7644	389.3045	234.8892	390.1833	3.2461
16	294.0542	381.1686	259.6765	381.3798	245.7355	380.4806	243.8865	381.1116	1.9537
17	325.2408	389.7077	295.9723	391.4827	282.8693	391.1428	278.4700	391.9408	4.4711
18	236.5075	419.4092	197.9363	417.3014	183.6007	416.5114	183.6007	416.5114	0.0000
19	272.5206	458.0212	234.9804	458.0873	220.7345	458.0572	219.4214	457.5033	1.4252
20	303.3359	420.1518	270.5278	421.7916	256.7654	421.2910	253.0771	421.3677	3.6891
21	276.6046	494.0343	238.3480	494.0089	223.3082	494.0880	223.3082	494.0880	0.0000

Figure 5.37 (uref1, vref1) and (uref2,vref2) are the image coordinates of feature points in the two reference frames. (utarget,vtarget) and (utracked,vtracked) are the original and tracked image coordinates of feature points in the target frame. trk_error is the tracking error between original feature points and tracked feature point in the target frame.

References

[1] O.D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?" in *Proc. European Conf. Comput. Vision,* 1992, pp. 563 - 578.

[2] Joseph L. Mundy, and Andrew Zisserman, " Appendix - projective geometry for machine vision", In J.Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pp. 463-519. The MIT press, 1992.

[3] A.Shashua, "Projective structure from uncalibrated images: structure from motion and recognition," In *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 8, pp. 778-790, 1994.

[4] Olivier Faugera, *Three-Dimensional Computer Vision*, Cambridge, MA : MIT Press, 1993.

Chapter 6

Conclusions and Future Researches

6.1 Contributions and Conclusions

We have accomplish the calibration for our novel active vision system. At first, Reg Willson's implementation of R. Y. Tsai's two stage RAC based camera calibration method was employed to do the camera calibration. Then, we have performed the head-eye calibration with a real image sequences which were taken by a camera fixed on the last link of the active system. The experimental results show that the estimated head-eye transformation can predict new camera pose quite satisfactorily.

We have proposed a new tracking method for tracking feature points for polyhedral objects over an image sequence. In our method, we have made use of the accurate active vision system for the tracking of the projective basis. This means the tracking has been reduced to conventional stereo matching along epipoloar lines. This method can be used to improve the speed and reliability of those shape from motion techniques requiring feature tracking.

We have rederive the 8-point projective invariants in the context of projective geometry. Tracking experiments on several image sequence of a calibration block were performed. It can be seen that this 8-point projective invariant based tracking program yielded quasi-acceptable results when it was applied in a situation where the perspective effects are the dominant one. It demonstrated the worst performance in a case of orthographic projection.

We have found that, in real situation, the lens distorsion has to be corrected at first so that the 8-point projective invariant can be used..The results from a experiment on synthetic data showed that, the performance of the tracking algorithm was perfect when no radial distorsion was involved.

6.2 Future Researches

We have suggested a new feature correspondenced based head-eye calibration technique. Three non_coplanar translation of robot are to be used to estimate the $\boldsymbol{R}_{cg}$. The orthonormality of the $\boldsymbol{R}_{cg}$ are ensured by an adoption of SVD based absolute orientation technique. It is well known that the enssential matrix $\boldsymbol{E}$ can be estimated given eight feature point correspondences across two images. Since $\boldsymbol{E} = \boldsymbol{T}_{cij} \times \boldsymbol{R}_{cij}$, and $\boldsymbol{R}_{cij}$ can be calculated as $\boldsymbol{R}_{cg}^{-1}\boldsymbol{R}_{gij}\boldsymbol{R}_{cg}$, we can get $\boldsymbol{T}_{cij}$ immediately. Finally, $\boldsymbol{T}_{cg}$ can be calculated with the following equation.

$$(\boldsymbol{R}_{gij}-\boldsymbol{I})\boldsymbol{T}_{cg} = \boldsymbol{R}_{cg}\boldsymbol{T}_{cij}-\boldsymbol{T}_{gij}$$

But please note that in this case, $\boldsymbol{T}_{cij}$ is determined only to an undetermined scale. Now the only problem remained is to determine this unknown scale.

To improve on our planar projective invariant based feature tracking method in the future, we shall use extended Kalman filter to predict the position of the projective basis in all the frames except for the first. Research will also be conducted on how to select the projective basis and how to change the reference frame dynamically. This shall make the tracking more accurate.

Research will be conducted toword the improvement of robustness of the tracking algorithm using 8-point projective invariants. We wish to construct the criterion of whether keeping or discarding the tracked feature point.

Bibliography

Chapter 1

[1] Aloimonos, I. Weiss and A. Bandyopadhyay, "Active vision" in *Proc. First Int. Conf. Comput. Vision*, June 1987, pp. 306-314.

[2] A.Shashua, "Projective structure from uncalibrated images: structure from motion and recognition," In *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 8, pp. 778-790, 1994.

[3] C. Fermüler, Y. Aloimonos, "The role of fixation in visual motion analysis," *Int. J. Comput. Vision*, 11:2, pp. 165-186, 1993.

[4] E. Krotkov, F. Fuma, J. Summers, "An agile stereo camera system for flexible image aquisition," *IEEE J. Robot. Autom.*, vol.4, no. 1, pp. 108-113,1988.

[5] A. L. Abbot, N. Ahuja, "Active surface reconstruction by integrating focus, vergence, stereo, and camera calibration," *Proc. 3rd Intern. Conf. Comput. Vis. Osaka, Japan*, pp. 489 - 492, 1990.

[6] T. Poggio *et al.*, "The MIT vision machine," *Proc. DARPA Image Understanding Workshop*, pp.177-198, 1988.

[7] C. M. Brown, "Kinematic and 3D motion prediction for gaze control," *Proc. Workshop on Interpretation of 3D scenes*, Austin, TX, pp. 145-141, 1989.

[8] J. J. Clark, N. J. Ferrier, "Modal control of atterntive vision system," *Proc. 2nd Intern. Conf. Comput. Vis.*, Tampa, FL, pp. 514-523, 1988.

[9] K. Pahlavan, J. O. Eklundh, "A head-eye system - Analysis and design," *Comput. Vis., Graph., Image Process.: Image Understanding Workshop*, vol. 56, no. 1, pp. 41-56, 1992.

[10] J. C. Fiala *et.al* , "TRICLOPS: A tool for studying active vision," *Int. J. Comput. Vision*, vol. 12, no. 2/3, pp. 231-250, 1994.

[11] Y.Aloimonos,I.Weiss,and A. Bandopadhay, "Active vision", *Int. J. Comput. Vision* 2(1), 1988, 333-356

[12] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method," *Technical Report CMU-CS-91-172*, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[13] Marvin J. Greenberg , Euclidean and Non-Euclidean Geometry , Development and

History, W. H. Freeman, San Francisco, 1980, pp. 367.

Chapter 2

[1] P. Beardsley, D. Murray, "Camera calibration using vanishing points," *Proc. British Machine Vision Conf.,* September, Leeds, pp. 22-24, 1992

[2] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vision*, 4, pp. 127-140, 1990.

[3] W. Faig, "Calibration of close-range photogrammetry systems: Mathematical formulation," *Photogrammetric Eng. Romote Sensing*, vol. 41, pp. 1479-1486,1975

[4] H. A. Martins, J. R. Birk, and R.B. Kelley, "Camera models based on data from two calibration planes," *Computer Graphics Image Processing*, vol. 17, pp. 173-180, 1981.

[5] R. Y. Tsai, "A versatile camera calibration techinque for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Trans. Robotics Automn.* vol. 3, no. 4, pp. 323-344, August, 1987.

[6] G. Q. Wei, S. D. Ma, "A complete two-plane camera calibration method and experimental comparisons," *Proc. 4th Intern. Conf. Comput. Vision, Berlin, May 11-14, 1993, pp. 133-138.*

[7] G. Q. Wei, S. D. Ma, "Implicit and Explicit Camera Calibration: Theory and Experiments," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol.16, no. 5, May,1994, pp. 469-480

[8] Y. Yakimovsky and R. Cunningham, "A system for extracting three dimensional measurements from a stereo pair of TV cameras," *Computer Graphics Image Processing*, vol. 7, pp. 195-210, 1978.

Chapter 3

[1] R. Y. Tsai, R. K. Lenz,"A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robotics Automa.*, vol. 5, no. 3, June 1989, pp. 345-358.

[2] Song De Ma and Guo-Qing Wei, "A self-calibration technique for active vision system," *in Proc. Asian Conf. Comput. Vision*, 1993, pp.628-631.

[3] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," *IEEE Trans. Robotics and Automat.*, vol. 5, no. 1, pp. 16-29, 1989.

[4] Y.Aloimonos,I.Weiss,and A. Bandopadhay, "Active vision", *Int. J. Comput. Vision* 2(1), 1988, 333-356

[5] R. Y. Tsai, "A versatile camera calibration techinque for high-accuracy 3D machine vision metrology using off-the-sehlf TV cameras and lenses," *IEEE Trans. Robotics Automn.* vol. 3, no. 4, pp. 323-344, August, 1987.

[6] T. Viéville,"Auto-calibration of visual sensor parameters on a robotic head," *Image and Vision Computing*, vol. 12, no. 4, pp. 227-237, May, 1994.

[7] G. Q. Wei, S. D. Ma, "A complete two-plane camera calibration method and experimental comparisons," *Proc. 4th Intern. Conf. Comput. Vision, Berlin,* May 11-14, 1993, pp. 133-138.

[8] R. Horaud, F. Dornaika, B. Boufama, and R.Mohr, "Self calibration of a stereo head mounted onto a robot arm," *Lecture Notes in Computer Science,* vol. 800. Jan-Olof Eklundh (Ed.), Computer Vision - ECCV'94, Springer-Verlag Berlin-Heidelberg 1994.

[9] J. Denavit, R.S. Hartenberg, "A kinematic notation for lower-pair machanisms based on matrices," *ASME Journal of Applied Mechanics*, June 1955, pp. 215-221.

[10] Robert M. Haralick, Hyonam Joo, "Pose Estimation from corresponding point data," *IEEE Trans. Systems, Man and Cybernetics*, vol 19, no. 6, November / December 1989. pp. 1426-1445.

Chapter 4

[1] Roger Mohr, Luce Morin, and Enrico Grosso, " Relative positioning with uncalibrated cameras ", In J.Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pp. 440-460. The MIT press, 1992.

[2] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method," *Technical Report CMU-CS-91-172*, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[3] C.J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery, *Technical Report CMU-CS-93-219*, Carnegie Mellon University, Pittsburgh, PA, December 1993.

[4] T. H. Wu, R. Chellppa and Q Zheng, "Experiments on Estimating Egomotion and Structure Parameters Using Long Monocular Image Sequences", *International Journal of Computer Vision*, 15, 77-103,1995.

[5] N. V. Efimov, Advanced Geometry, Mir, Moscow, 1978.

Chapter 5

[1] O.D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?" in *Proc. European Conf. Comput. Vision,* 1992, pp. 563 - 578.

[2] Joseph L. Mundy, and Andrew Zisserman, " Appendix - projective geometry for machine vision", In J.Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pp. 463-519. The MIT press, 1992.

[3] A.Shashua, "Projective structure from uncalibrated images: structure from motion and recognition," In *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 8, pp. 778-790, 1994.

[4] Olivier Faugera, *Three-Dimensional Computer Vision*, Cambridge, MA : MIT Press, 1993.

CUHK Libraries
003589447