A Distributed System of Pan/tilt Cameras for 3D Tracking

A DISTRIBUTED SYSTEM OF PAN/TILT CAMERAS FOR 3D TRACKING

By

DAVID C. WOO, B. Eng.
McMaster University, Hamilton, Ontario, Canada

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Engineering

McMaster University

MASTER OF ENGINEERING (2000) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: A Distributed System of Pan/tilt Cameras for 3D Tracking

AUTHOR: David C. Woo

B. Eng.

McMaster University, Hamilton, Ontario, Canada

SUPERVISOR: Dr. David W. Capson

NUMBER OF PAGES: xi,91

Abstract

This thesis describes a distributed system of cameras for visually tracking feature points in 3D. The concept of a *network of cameras* is introduced. Two or more calibrated cameras from a network of cameras are used to triangulate the location of a point in 3D based on camera positions and pan/tilt angles. A survey of methods for interior and exterior calibration is provided and a method suitable for multiple cameras in arbitrary positions has been implemented.

A low-cost camera unit has been designed using off-the-shelf components that include a small CCD board camera and two servo-controlled mechanisms for pan and tilt. Experimental results demonstrate the performance of a network of cameras.

Acknowledgements

I would like to thank Dr. David Capson for giving me the opportunity to excel and learn by providing support for this research. To the other members of the Machine Vision and Image Analysis Laboratory namely, Jay, Phil, Gorav and Jeff, I owe a great debt of gratitude. Without their support, guidance and friendship, this thesis would not have been completed as easily. I would also like to thank the staff of the Electrical and Computer Engineering Department who take excellent care of their graduate students and especially Cheryl and Barb who never fail to make our lives easier. Finally, I would like to thank my family, Eliana *et al.*, Marnie, Christine and of course, Lauren for putting up with me and making my stay in Hamilton full of laughs and excitement.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

b	Baseline
d	Disparity
f	Camera Constant(Focal Length)
IR	Infrared
l	Line
$\vec{\ell}$	Direction of a Line
NOC	Network of Cameras
P	A Cartesian Coordinate Point
PanCal	Pan Servo Calibration Constant
PWM	Pulse Width Modulated
R	Rotation Matrix
RC	Radio Controlled
SSD	Sum of Squared Differences
SSSD	Sum of Sum of Squared Differences
$\vec{t}$	Translation Matrix
T	Homogeneous Transform Matrix
TiltCal	Tilt Servo Calibration Constant
(u, v)	A Point on the Image Plane
VASC	The Vision and Autonomous Systems Center
WCS	World Coordinate System

Chapter 1

Introduction

1.1 Computer Vision

Many outstanding issues with computer vision remain unsolved. Among them is correspondence between stereo pairs of images and occlusion. Correspondence is the matching of image points from one image to the same points in a second, while occlusion occurs when part or all of the image view is blocked. These issues were even more prominent in the past when imaging and computing technology was in early stages. High cost vision systems of the past often included only one imaging device, a custom hardware interface and a computer. Computer vision has dramatically changed in recent years due to the falling cost of computing. Off the shelf frame grabbers allow cameras to be connected to a typical PC with enough computing power to perform complex image processing at frame rates. However, correspondence and occlusion remain a problem.

One method to solve the correspondence and occlusion problem is to gather more information by using more cameras. This would have been too financially and computationally expensive in the past but is possible today. Two new issues arise when considering the use of multiple cameras. The first is in how a large number of cameras

can be connected to a computer and the second issue is the time consuming processing of so much image information. These two issues are addressed in the following sections.

1.2 Smart Cameras

Time consuming computer processing is often spent filtering and modifying the image to allow the correct information to be extracted. This type of preprocessing, such as low-pass or high-pass filtering, should not be the responsibility of the central processing computer when low-level operations could be performed by a simpler, lower cost computer. In this thesis, the idea of a *smart camera* is used.

A smart camera has increased functionality over a normal camera that only produces a video stream. An embedded processor in a smart camera can filter the images as they are generated and transmit the enhanced image. The functionality of a smart camera could include:

- auto focusing,
- auto exposure,
- iris control,
- camera movement control (i.e. pan/tilt),
- zoom control, and
- image preprocessing.

All of these functions can be built together in a single unit to produce a smart camera.

1.3 A Network of Cameras

A video multiplexer would allow many cameras to be connected through a frame grabber to a computer. However, this thesis proposes an alternative to a multiplexer

that takes advantage of a smart camera and the falling cost of networking. Computer networks have generated what is known today as the Internet. The immense demands on the Internet for data and information sharing, electronic commerce, business transactions and communication have reduced the cost of networking. Hence, a smart camera interface could be a high-speed network interface such as ethernet. A large number of cameras could be connected to a computer in a *network of cameras* (NOC).

A feature of a NOC is the ability to send information or commands to and from the camera. A central computer could control the pan/tilt capabilities of a smart camera or control the zoom. The functionality of a smart camera can be altered through the network. New parameters could be transmitted to the camera to adjust the filtering. A smart camera could be programed to autonomously control pan, tilt and zoom to search for people and report through the network when a person is detected.

Figure 1.1 shows an example of a NOC. This configuration illustrates how a network can take on many different topologies. In this example, the visual security system has many smart cameras that maybe located throughout a building connected through an ethernet switch or in other buildings connected through the Internet. Figure 1.1 also shows some examples of smart cameras with wireless links, power zoom and pan/tilt capabilities.

A smart camera could drastically reduce bandwidth load on a network. For example, to transmit video at 30 frames per second, 24 bits of color and image size of 640 pixels by 480 pixels, the bandwidth required would be 221.2 Mbits per second for just the video data. Multiple cameras transmitting video at this rate would not be possible using current copper-based ethernet networks. However, a smart camera would not have to necessarily transmit images at 30 frames per second. A smart camera may not have to transmit any images at all and instead process the images and transmit high-level data such as the position of an object in the image. Hence,

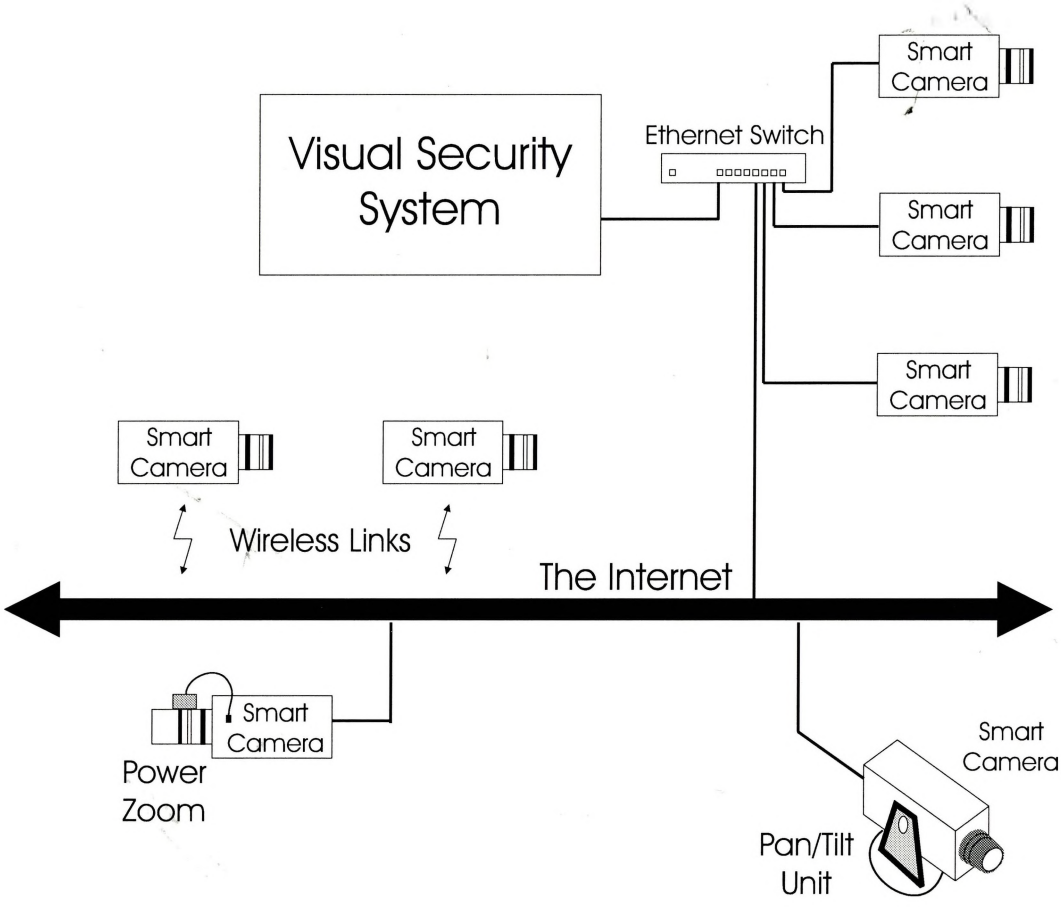

Figure 1.1: A Network of Smart Cameras

bandwidth requirements could be reduced to hundreds of bits per second as opposed to millions of bits per second by using smart cameras.

1.4 Tracking

Computer vision is often used in tracking. For example, cameras are used in manufacturing to detect positions of parts on conveyor belts and to track parts as they enter and leave work cells. A NOC can be used in tracking applications with the advantages outlined in the previous sections. Tracking using a NOC introduces new issues such as:

- number of cameras to use,
- location of cameras,
- smart camera features (zoom, pan/tilt), and
- extent of smart camera autonomy.

Camera locations will effect the view of the object being tracked and the tracking accuracy. Figure 1.2 shows an object location and domain of movement. Three possible configurations for 4 cameras, C0, C1, C2 and C3, are depicted in Fig. 1.2a, 1.2b and 1.2c. Optimal parameters for a NOC such as camera location is application dependent. This thesis will explore criteria for camera location and other NOC parameters.

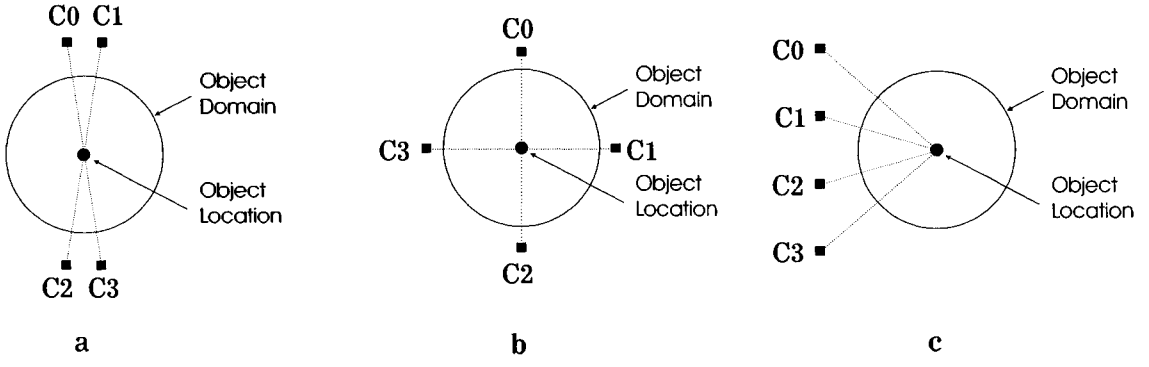

Figure 1.2: Possible Locations for Four Cameras

1.5 History

1.5.1 Active Vision

Vision systems often concern themselves with using camera sensors to model or recreate the observed environment in an attempt to provide control information for applications such as mobile robots [35] or virtual reality worlds [8]. Active vision techniques use the control of the camera pose relative to the object of interest to both select manageable portions of the environment relevant to the current task as well as to simplify the extraction of the required parameters from those selected views.

Active vision is not limited to pointing the camera in a certain direction, but includes any adaptive response to the demands of the task given the environment. This would include increasing zoom to enhance resolution of a smaller area, and/or adjusting the iris to change the amount of light reaching the CCD and/or changing the depth of field. For stereo information derived from two cameras, active cameras can be verged to improve disparity resolution for a fixed baseline [20] (see also §1.5.3). Depth can also be found by varying the focus of the camera lens thereby requiring only one camera and the correspondence problem is avoided altogether [22].

The Vision and Autonomous Systems Center (VASC) at Carnegie Mellon University, Pittsburgh, has used a large array of fixed cameras about the perimeter of an environment as an alternative to an active vision system to create a virtual reality [32]. Once the cameras are calibrated, multiple-baseline stereo [30] is used to create depth maps from which a model is generated and texture-mapped. The array of cameras mounted along the walls or in a geodesic dome eliminates the problem of position but incurs a host of problems inherent with static sensors. To reduce the likelihood of occlusion, many cameras must be used which leads to bandwidth problems, while the computational cost of constructing a model from stereo data at each time instant is formidable. Calculating accurate, dense, high resolution depth maps from pairs of images is far from being solved, necessitating the need for careful filtering.

The VASC system is an example of where a NOC could be applicable. The array of fixed cameras could be replaced with a smaller number of cameras capable of locating and following the object of interest. Orientation of the cameras could be passed over the network to allow the use of the multiple-baseline method. A camera pair could be seen as a smart camera that does not produce an image but rather a depth field. Hence, a NOC could provide vital improvements that ease the computational cost of performing real-time virtual views.

1.5.2 The Biological Model

Active vision has been proven to be superior to non-active vision in many applications. The human vision system is a prime example of active vision says Marr [26]. Human eyes have a very small fovea (1.5 deg. diameter), where visual acuity is best and can be likened to a camera lens where the lens distortion is usually lower in the middle. Eye movements are needed to both shift gaze to objects of interest, bringing the selected portion of the retinal image to the fovea and to maintain stable gaze on selected targets as they move. If the target should change its position abruptly, or

the object of interest changes, a single saccadic eye movement can take the line of sight to the new position of the target with considerable accuracy [7].

Kowler [21], outlines four examples of human active vision relevant to active vision systems for robotics. They are:

- selection of the target for smooth eye movements,
- predicting the future position of targets,
- planning sequences of saccades, and
- saccades to selected targets in the presence of irrelevant visual backgrounds.

Kowler concludes that target selection is based on the representations of the objects and not on isolated sensory cues, that is, what to track is a high level decision. Furthermore, the control of smooth eye movements include projections of the expected target motion several hundred milliseconds into the future. It is not difficult to see how these two conclusions may be applied to an active vision system. Indeed many biological vision systems have developed sophisticated ways of moving their eyes to exploit the benefits associated with eye movements.

Sharkey *et al.* [34], have developed a modular stereo head platform to test biological based active vision theories. This mechanical system, based on geared DC motors, is able to accurately track an object moving at 8m/sec, at a range of 2m from the cameras at 25 frames/sec. This design has lead to the development of the Yorick series of active stereo camera systems [33], which has been used to test gaze control, dynamic vergence [2], and saccadic motion [28], primarily for virtual reality telepresence. The implementation is capable of performing a real-time surveillance task in a changing, unstructured environment using optical flow at a coarse scale across an entire image and at a fine scale in a central foveal region required by the different saccade and pursuit actions. It also uses a prediction scheme to determine the likely future movement of the object.

Depth information is an integral part of human visual perception. Two common approaches for the determination of depth are described next.

1.5.3 Depth from Multiple Views

The literature contains many examples of tracking and range mapping in 3D using two or three camera stereo matching [15, 18, 20, 9]. Stereo matching involves a known *baseline*, which is the distance between two cameras and determining *matching* points in the images. A point in one image matches or *corresponds* to a point in the second image when both points are the projection of the same real point in the 3D space. Sharkey *et al.* [34] used binocular vision where two cameras are placed along a straight line. A trinocular system, as used in [23, 38], has cameras placed in a triangle to improve the depth measurement. Figure 1.3 illustrates the scene point projecting on to the image planes of two cameras at p_l and p_r. The distance between the camera axes is the baseline b while f is the distance from the image plane to the focal point c_l or c_r. The distance from the cameras to the object point is the depth z. In stereo matching systems the measured disparity $d = (x_l - x_r)$ between matched points is related to the distance (depth) z by:

$$z = \frac{bf}{(x_l - x_r)} \,. \tag{1.1}$$

Equation 1.1 is usually written as:

$$d = bf\frac{1}{z} \tag{1.2}$$

Much research has been devoted to dealing with the fundamental tradeoff between the ease of matching and the accuracy dictated by this equation. For a given distance, Eq. 1.1 indicates that disparity is proportional to the baseline. Hence, a longer baseline would allow a better estimate of the distance. However, a longer baseline

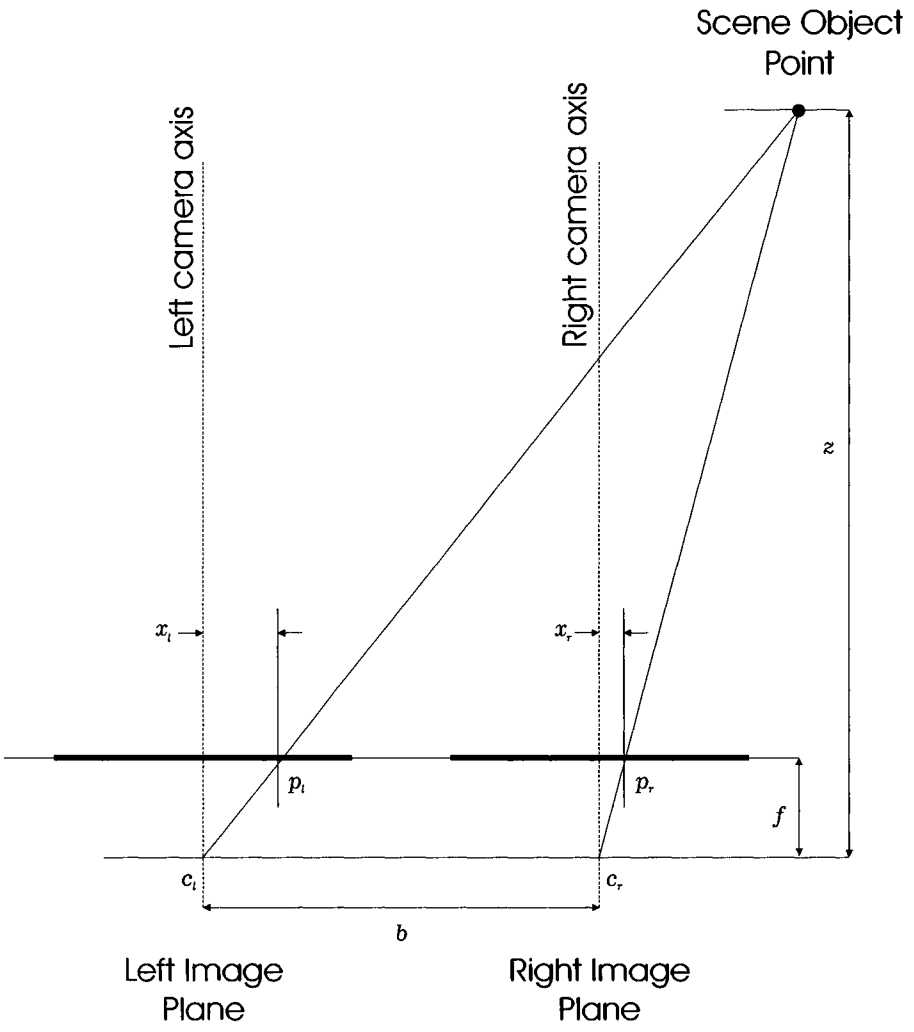

Figure 1.3: Depth from a Stereo Image Pair

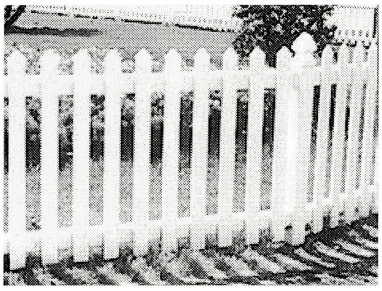

Figure 1.4: Picket Fence

makes matching more difficult, time consuming and increases the chance of a false match.

A solution to this tradeoff is a coarse-to-fine control strategy [12]. This strategy uses a low resolution to reduce false matches and then a higher resolution where more precise disparity measurements are calculated. Another approach to reduce false matches and increase precision is to use multiple images sampled along a camera path [1]. Consecutive images would provide a short baseline, reducing false matches, and can be integrated with information from other images in the sequence for a longer baseline. Kalman filtering has also been used to integrate the images [27]. These techniques however, still succumb to the inherent ambiguity in matching produced by, for example, a repeated pattern. In a repeated pattern, such as the picket fence in Fig. 1.4, matching is ambiguous because there can be more than one match for a given pattern.

Multiple-baseline Stereo

The multiple-baseline method, as used in the VASC system, was originally proposed by Okutomi and Kanade [30] and has been widely used. The stereo matching method uses multiple stereo pairs with different baselines generated by a lateral displacement. Matching is performed by computing the sum of the sum of squared-difference (SSSD)

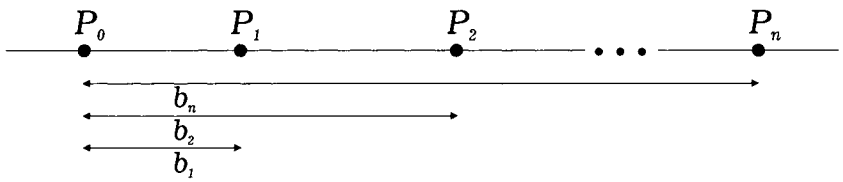

Figure 1.5: Multiple Baseline Camera Positions

values with respect to the inverse disparity for each stereo pair. Okutomi and Kanade [30] begin with cameras at position $P_0, P_1, ..., P_n$ along a line with optical axes perpendicular to the line resulting in a set of stereo pairs with baselines $b_1, b_2, ..., b_n$ as shown in Fig. 1.5. Each image can be represented as intensity function $f_i(x)$, $i \in \{0, 1, 2...n\}$. The sum of the squared difference (SSD), $e_{d_i}(x, d_i)$, over a window W beginning at pixel position x of image $f_o(x)$ for the candidate disparity d_i is defined as:

$$e_{d_i}(x, d_i) \equiv \sum_{j \in W} (f_o(x + j) - f_i(x + d_i + j))^2 \tag{1.3}$$

where the $\sum_{j \in W}$ means the summation over the window W for each position j. The value of d_i that minimizes $e_{d_i}(x, d_i)$ is used as the estimate of the disparity at position x. The SSD with respect to the inverse distance $\zeta = \frac{1}{z}$ is given by:

$$e_{\zeta_i}(x, \zeta) \equiv \sum_{j \in W} (f_o(x + j) - f_i(x + b_i f \zeta + j))^2 \tag{1.4}$$

where b_i and f are the baselines and focal length respectively. The SSD values with respect to the ζ of all the pairs are summed to produce the sum of the SSD or SSSD:

$$e_{\zeta_{(1,2...n)}}(x, \zeta) = \sum_{i=1}^{n} e_{\zeta_i}(x, \zeta). \tag{1.5}$$

Okutomi and Kanade [30] report that the advantage of using the SSSD function with respect to the inverse disparity is the unique and clear minimum at the correct matching position, even when the underlying intensity patterns of the scene include

ambiguities or repetitive patterns. Hence, with a small iterative search for each pixel, good matches can be found from multiple stereo images. It is easy to see that a search on a pixel-by-pixel level can be time consuming although this method would lend itself well to parallel computations since each search is independent of the next.

1.5.4 Depth from a Single View

A drawback from extracting depth from multiple views is the high computational expense. Also, many applications such as robot navigation and object tracking do not require the depth of all points in the image. Often, only the depth of one point is required to attain the 3D position of a robot or an object.

Depth information can be extracted from a single image provided there is a priori knowledge about the objects in the image. There are many publications using this technique involving natural or artificial features in the image. An artificial feature could be a marker such as the diamond shape used in [11] or a cube as used in [5]. House corners provide a natural feature in [6], while [13] describes the use of parametric planar or nonplanar curves to determine depth.

The basic theory in each of these techniques is based on expressing the 3D representation of the real world feature as a 2D perspective projection on the image plane. Both a house corner and one corner of a cube provide essentially three lines that meet at one point that are at right angles to each other. A set of equations for the perspective projections of these lines onto the image plane can be developed by using the *law of collinearity* [14, 39], which states that an object point, its perspective projection and the camera focal point all lie along a straight line. Refer to Appendix B for further details. Since the perspective projection is dependent upon the view of the camera, the equations yield the *pose* of the camera with respect to the three lines. Given the pose and the length of one of the lines, the depth from the vertex can be determined.

For example, Chou and Tsai [6], proposed an approach using house corners (corners of rooms) as natural features for robot location. For this method to work the assumption that the distance from the camera to the ceiling of the room is known a priori. Given a monocular image of a house corner, Chou and Tsai describe the three lines through the corner point (u_p, v_p) in image coordinates, as $u_p + a_i v_p + c_i = 0,\ i \in \{1, 2, 3\}$. In the first step, the six coefficients $a_i,\ c_i$, are derived in terms of the desired camera location $P_{ORG} = (P_{ORG_x},\ P_{ORG_y},\ P_{ORG_z})$ and the camera orientation as given by $\psi,\ \theta$, and δ (these are often referred to as the pan, tilt and swing angles respectively of the camera). Beginning with the perspective projection equations as defined in Appendix B and Fig. B.1:

$$\begin{cases} u_p = fP_x/P_z \\ v_p = fP_y/P_z \end{cases} \tag{1.6}$$

for any point $P = (P_x, P_y, P_z)$ in the camera coordinate system where (u_p, v_p) is the location of P in the image. After some manipulation, the equations for $a_i,\ c_i$, are:

$$a_1 = \frac{-P_{ORG_z} \sin\psi \sin\delta + (P_{ORG_y} \sin\theta - P_{ORG_z} \cos\psi \cos\theta) \cos\delta}{-P_{ORG_z} \sin\psi \cos\delta - (P_{ORG_y} \sin\theta - P_{ORG_z} \cos\psi \cos\theta) \sin\delta}, \tag{1.7}$$

$$c_1 = \frac{f(-P_{ORG_y} \cos\theta - P_{ORG_z} \cos\psi \sin\theta)}{-P_{ORG_z} \sin\psi \cos\delta - (P_{ORG_y} \sin\theta - P_{ORG_z} \cos\psi \cos\theta) \sin\delta}, \tag{1.8}$$

$$a_2 = \frac{P_{ORG_z} \cos\psi \sin\delta + (-P_{ORG_x} \sin\theta - P_{ORG_z} \sin\psi \cos\theta) \cos\delta}{P_{ORG_z} \cos\psi \cos\delta - (-P_{ORG_x} \sin\theta - P_{ORG_z} \sin\psi \cos\theta) \sin\delta}, \tag{1.9}$$

$$c_2 = \frac{f(P_{ORG_x} \cos\theta - P_{ORG_z} \sin\psi \sin\theta)}{P_{ORG_z} \cos\psi \cos\delta - (-P_{ORG_x} \sin\theta - P_{ORG_z} \sin\psi \cos\theta) \sin\delta}, \tag{1.10}$$

$$a_3 = \frac{(P_{ORG_x} \sin\psi - P_{ORG_y} \cos\psi) \sin\delta + (P_{ORG_x} \cos\psi + P_{ORG_y} \sin\psi) \cos\theta \cos\delta}{(P_{ORG_x} \sin\psi - P_{ORG_y} \cos\psi) \cos\delta - (P_{ORG_x} \cos\psi + P_{ORG_y} \sin\psi) \cos\theta \sin\delta}, \tag{1.11}$$

$$c_3 = \frac{f(P_{ORG_x}\cos\psi + P_{ORG_y}\sin\psi)\sin\theta}{(P_{ORG_x}\sin\psi - P_{ORG_y}\cos\psi)\cos\delta - (P_{ORG_x}\cos\psi + P_{ORG_y}\sin\psi)\cos\theta\sin\delta} . \tag{1.12}$$

Combining these equations and solving for δ:

$$A_\delta \tan^2\delta + B_\delta \tan\delta + C_\delta = 0 , \tag{1.13}$$

where

$$\begin{aligned} A_\delta &= f^2c_3^2a_1a_2 + c_3^2c_1c_2 + f^2c_1c_3 + f^2c_2c_3 + f^4(1 + a_1a_2), \\ B_\delta &= f^2c_3^2(a_1 + a_2) + f^2c_3c_1(a_2 + a_3) - f^2c_2c_3(a_3 + a_1) - 2f^4a_3(1 + a_1a_2) \quad (1.14) \\ C_\delta &= f^2c_3^2 + c_3^2c_1c_2 + f^2c_1c_3a_2a_3 + f^2c_2c_3a_1a_3 + f^4(1 + a_1a_2)a_3^2 . \quad (1.15) \end{aligned}$$

Now, the coefficients A_δ, B_δ and C_δ are all expressed in terms of known values a_i, c_i and f. Therefore, Eq. 1.13 can be solved to obtain:

$$\tan\delta = \frac{-B_\delta \pm \sqrt{B_\delta^2 - 4A_\delta C_\delta}}{2A_\delta} , \tag{1.16}$$

where the sign before the square root was found to be minus by experimentation. After the value δ is computed, it can be substituted into the equations used to derive the above equations to find values for θ, ψ, P_{ORG_x}, and P_{ORG_y}. P_{ORG_z} is known a priori as the distance from the ceiling to the camera.

The obvious drawback of the single view techniques is the need for a priori knowledge. Furthermore, multiple views provide more information for the matching of the a priori knowledge to the image content. A single view is also prone to problems created by occlusion. The advantage however, is in the computational efficiency as compared to the multiple view techniques. The system described in this thesis takes advantage of long baselines to gain accuracy and uses natural features in a single view to simplify matching.

1.5.5 Summary

The important ideas presented in the above sections are:

- biological systems such as the human vision system exemplify the advantages of active vision,
- common vision systems often use stereo to determine depth,
- matching is required in stereo depth measurements and is increasingly difficult with multiple baseline systems requiring time consuming searches, and
- depth from a single view relies on a priori information but can be computed quickly.

Using a Yorick stereo camera head, Sharkey *et al.* [34] achieved some of the goals of active vision while keeping real-time performance. High torque motors, high precision optical encoders and a modular custom machined mount however, would likely make the system expensive. The control system contains tightly coupled, high speed transputers and custom electronic hardware which would also add to the system cost.

The VASC geodesic dome for virtualized reality [32] uses 51 video cameras each with their own video recorder capable of adding the necessary time stamp to each frame. The more recent VASC project [17], with 49 cameras fixed along the perimeter of a room uses 17 computers to record and time stamp a series of images. The physical constraints and cost of such a system may limit practicality. VASC has not been able to process the data in real-time to produce virtual realities. All of the processing is done off-line after the image sequences have been recorded. There is an obvious need to make better use of new technologies and to explore new methods to combat the financial and computational cost of vision systems.

Today, CCD technology and CMOS technology has reduced the cost of cameras while public demand for computers and Internet applications have dramatically reduced the cost of computing. One other aspect of today's technology that may be taken advantage of is the computer inter-connectivity. Networks such as ethernet are common and can be built at low cost. This thesis takes advantage of these technologies and combines them with active vision and the old method of triangulation, forming the basis for an inexpensive 3D tracking system using off-the-shelf components.

1.6 Thesis Outline

This new system unites the single view approach of §1.5.4 with active vision, §1.5.1 and uses triangulation to calculate the 3D position of an object. Four smart cameras with pan/tilt mechanisms has been built using inexpensive board cameras and hobby servos. The smart cameras can be positioned arbitrarily and calibrated to form a NOC. Using any two cameras in the network which actively track a common target, the target position can be triangulated.

Artificial or natural features are actively and independently tracked by each camera, keeping the feature in the center of the image thereby simplifying correspondence and keeping the relevant image information in the fovea of the lens. The camera placement becomes arbitrary and can be adapted for range measurements as required by the application such as 3D model generation, multimedia applications, robotic servo control and security tracking. With a large network of pan/tilt cameras observing a target, tracking can be dynamically based on two or more cameras. Tracking responsibility can be passed along from one camera set to another as the target moves out of view of certain cameras and into others. Thus, targets may be tracked throughout any area, such as along an assembly line or along a highway.

As with any practical system there are trade-offs. Although the proposed system

integrates single view tracking with the benefits of active vision to reduce the computational cost, there is a computational penalty in reducing the cost of the mechanical system. The high cost of a Yorick sensor gains precision and speed while the low cost of this system sacrifices precision and speed. To compensate, an increased software burden is required. However, when the overall practicality of this system is assessed, the advantages are readily applicable to commercial and industrial applications.

This thesis describes the implementation of a simple NOC. The goal was to demonstrate the feasibility of a NOC with a simple tracking application and to explore the advantages of a NOC in distributing the computational load amongst the smart cameras. The NOC was designed to allow for arbitrary placement of cameras and to calculate the 3D position of the object being tracked. A low-cost pan/tilt camera unit was developed and is described in Chapter 2.

Chapter 2 also outlines the calibration process used to calibrate the NOC and also describes the process used to calibrate the pan/tilt control system. This calibration along with the extrinsic parameters are of key importance since the resulting feature location is dependent on the accuracy of these parameters.

The tracking algorithm, as described in Chapter 3, is designed to track a feature point and keep the feature point in the center of the image. Each camera tracks the feature point independently. This implies the optical axis of each camera will pass through the feature point. Hence, with two or more calibrated cameras, the 3D location of the feature point can be calculated. For simplicity, an infrared point source was used as a feature point (target). Chapter 3 also describes the method used to find the intersection of the optical axis.

Experimental results are outlined in Chapter 4. Several experiments were conducted to test the system. A representation of the error is plotted for several of the tests and observations are listed.

Conclusions and future directions are formed in Chapter 5.

Chapter 2

Pan/Tilt Camera Units

2.1 Pan/Tilt Camera Units

2.1.1 Introduction

To reap the benefits of active vision and to maintain a low-cost system, a pan/tilt camera unit was designed using readily available inexpensive parts. Precision pan/tilt units such as the Yorick [33]system and the Cohu [29] system, could make the cost of a network of pan/tilt cameras prohibitive in some cases. Similarly, the cost of high quality digital cameras and low distortion lenses can increase the cost. Hence, there is a need for a low-cost solution. A pan/tilt camera unit was produced using a EM200-L60 board level camera with a 6mm lens and two Futaba S3003 servos at an approximate cost of CND$300. A photo can be seen in Fig. 2.1.

2.1.2 Board Cameras

The EM200-L60 board level camera with a 6mm lens is widely used where low image quality is tolerable. Image quality has limited it to multimedia and surveillance applications such as Internet video and hidden security cameras. The specifications of

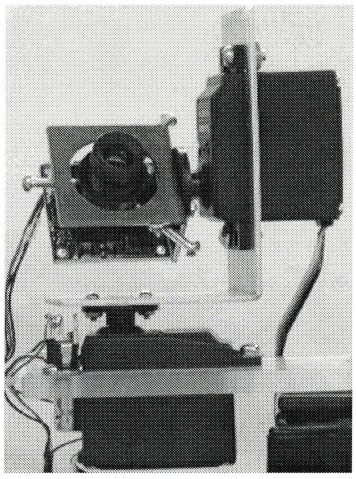

Figure 2.1: The Pan/tilt Camera Unit

this camera are listed in Table 2.1. The small size and low mass allow the mechanical system to be smaller and require less power. Whereas stereo matching algorithms are sensitive to outliers, which can be caused by low image quality, the tracking system used here is not sensitive to outliers and will tolerate the image quality produced by the camera. A photo of the board level camera may be seen in Fig. 2.2. Although the threaded lens mount allows the image to be focused on the image plane, the coarseness and looseness of the thread does not keep the lens well aligned.

2.1.3 Servos

The servos are self-contained units employing a poled or coreless DC motor with gear head reduction to increase torque and to allow rotational feedback. The servo contains all the control circuitry to convert the pulse width modulated (PWM) input signal to rotational position of the output shaft. Hence, only three wires need be connected to a servo: power, ground and the PWM signal. On a typical radio controlled (RC) system, a multichannel receiver has a servo connected to each PWM channel which

Table 2.1: Specifications of the EM200-L60 Board Level Camera

Dimensions:	L:3.2cm x W:3.2cm x H:3.2cm
TV System:	EIA Standard
Image Sensor:	1/3" CCD Interline Transfer 512(H) x 492(V) Pixels
Scanning System:	2:1 Interlaced
Sync. System:	Internal
Scanning Frequency:	15.734KHz (H) / 59.94Hz (V)
Resolution:	380 (H) / 330(V) TV Lines
Video Output (VBS):	Video Output 0.714Vp-p Sync. Output 0.286Vp-p
S/N Ratio:	45dB
Minimum Illuminance:	0.3 lux (faceplate sensitivity)
γ Characteristic:	0.45
Image Out:	Combined Video Signal 1.0Vp-p/75W
Electronic Shutter:	Auto Linear (1/60 - 1/80,000 sec)
AGC:	More than 12dB
Power Consumption:	118mA
Power Supply:	8 - 11 VDC (9 VDC Standard)
Operating Temperature:	$-10°C$ to $+50°C$

controls mechanical systems such as engine throttle, aireolons, flaps, or steering. RC servos are made to handle adverse conditions such as being mounted next to a gas engine in a model airplane or the bouncing incurred on a RC car race track. RC servos are thus robust and not sensitive to mechanical vibrations or impacts making them well suited for low accuracy positioning.

Two Futaba S3003 servos are used to rotate the EM200-L60 camera in pan and tilt directions. This servo type is considered a standard servo only due to the popularity of the size and performance, thus making it one of the most inexpensive servos to purchase. The specifications of the Futaba S3003 servo are listed in Table 2.2. Servos with increased accuracy, speed, torque or special size are available usually at a higher cost. A photo of the servo may be seen in Fig. 2.2.

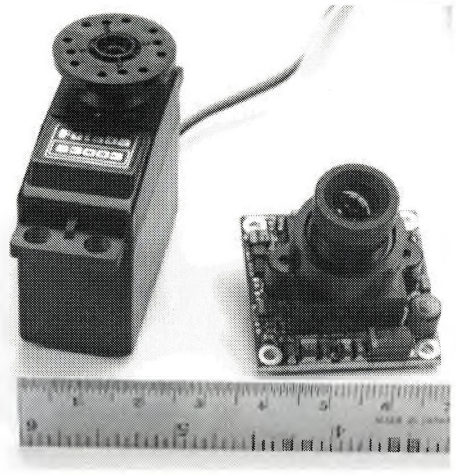

Figure 2.2: The Futaba S3003 Servo and EM200-L60 Board Level Camera

Table 2.2: Specifications of the Futaba S3003 Servo

Dimensions:	L:4.04cm: x W:1.69cm x H:3.58cm
Weight:	43g
Torque:	0.297Nm
Transit Time:	0.22sec/60°

The PWM Input Signal

A servo controls the servo motor to move the output shaft to the position corresponding to the pulse width of the PWM input signal. The servo continuously updates the position based on the pulse width. Since the feedback position is directly coupled to the output shaft, the servo will attempt to maintain the position even when an external torque is applied to the output shaft. Should the servo lose the PWM input signal it will maintain the last known position. Most standard servos have a mid-position set at a pulse width of 1.5ms with a period of 30ms (refer to Fig. 2.3).

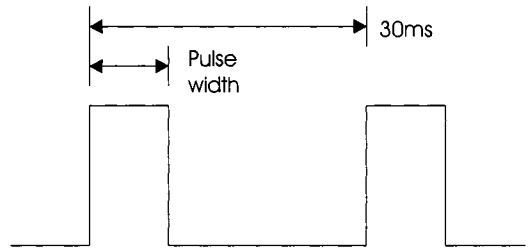

Figure 2.3: Pulse Width Modulated Signal

Servo Controller

The servo PWM signals were generated using an Intel 8031 microcontroller as seen in the block diagram in Fig. 2.4. Servo positions were issued from a host PC connected to the microcontroller via the serial port. The interface provides 13-bit resolution in the PWM signal which is of greater resolution than that of the servo positioning. The servo positions are updated at 33Hz and the servos are capable of positioning the camera at a rate of 270 degrees per sec.

The PWM control software for the microcontroller was written in assembly language. A simple, 24-bit packet system was developed to send servo position updates from a PC to the microcontroller via the RS232 serial link. The first 8 bits of a packet is a header defining the beginning of a packet. The next 3 bits is the servo identification and the remaining 13 bits represent the servo position. The microcontroller main program continually polls the serial port looking for packets. When a packet arrives the program updates a circular buffer stored in memory containing the servo positions.

The PWM signals are generated with an 8-bit latch connected to the data bus of the 8031. Referring to Fig. 2.5, L.0 to L.7 are pins 0 to 7 on the latch and provide PWM signals for servos 0 to 7 respectively. Eight PWM signals can thus be created. The servo position for servo 0 is read from the circular buffer and an internal timer

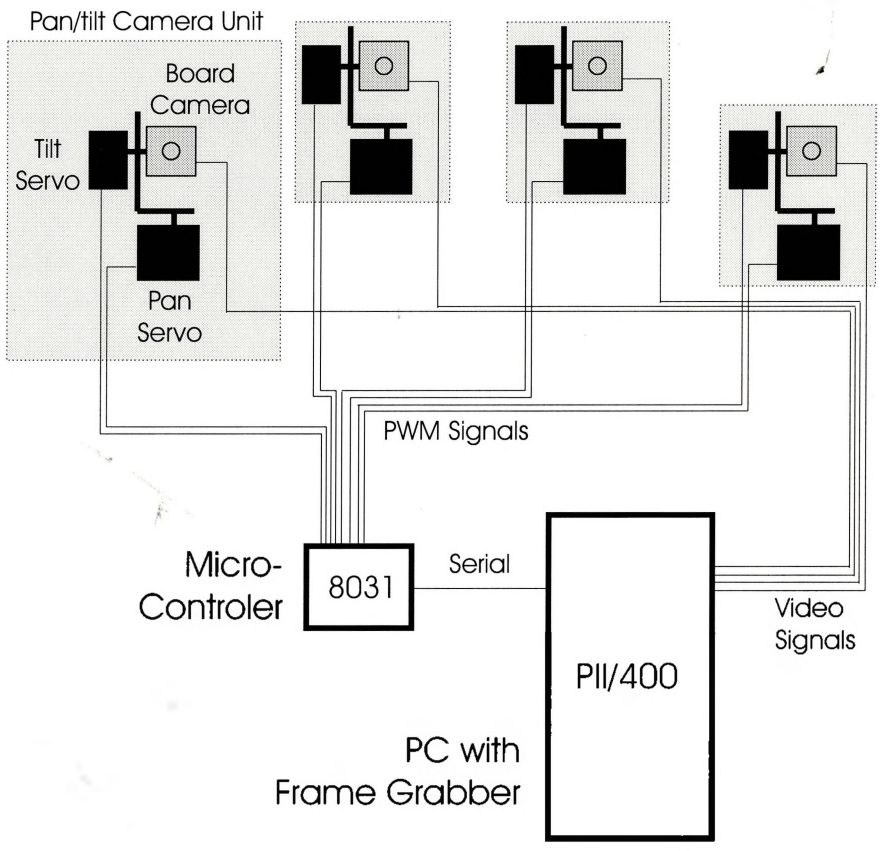

Figure 2.4: Interconnections of the Micro-controller, Pan/tilt Camera Units and PC

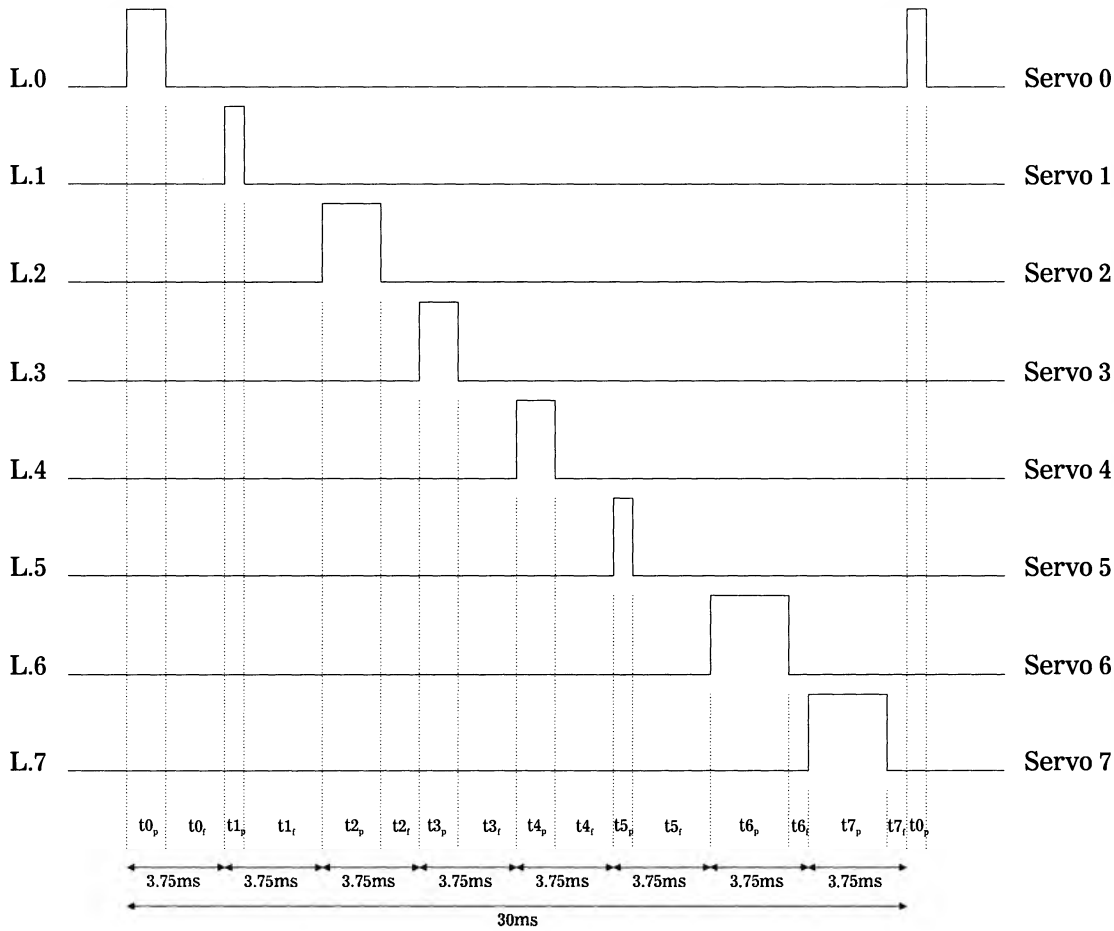

Figure 2.5: Timing Diagram for Servo PWM Signals

(integral to the microcontroller) is set to the corresponding pulse width time, $t0_p$. When the timer finishes, it is reset to the time $t0_f = 3.75 - t0_p$. Upon completion of $t0_f$, the process is started again by reading the servo position for servo 1 from the circular buffer. Each time the timer finishes, the pin on the latch corresponding to the servo pulse width being created, is switched low or the pin corresponding to the next servo PWM signal is switched high. Hence, when all 8 servos have been completed and timing for servo 0 begins again, 30ms will have passed. Thus each PWM signal will have a period of 30ms. This limits the pulse width of any given signal to 3.75ms. This is not a problem since the servo maximum position corresponds to a pulse width of 2.4ms.

2.1.4 Mechanical Design

The board level camera is held in a 3 point mount connected to the tilt servo allowing the camera to be arbitrarily positioned with respect to the rotational axes of pan and tilt servos. An "L" bracket couples the pan and tilt servos, keeping the axes of rotation orthogonal. A key feature of this design is the mounts, which ensure the axes of rotation pass through the camera focal point. This implies the image will be invariant to the pan and tilt movements [19]. This ensures that given a pan or tilt movement, no previously visable points will be occluded by other static points within the scene. This is important, since it implies there is no fundamental change in the information about a scene at different camera orientations. Refer to §3.2.2 for an explanation of the results when point of rotation is not at the focal point. A mounting plate holds the pan servo and provides connectors for video, power and a standard threaded camera tripod mounting hole. A diagram of the design can be seen in Fig. 2.6.

The final pan/tilt camera unit can position the camera to point in any direction within a hemisphere since each servo has 180° of rotation. Along with the ease of

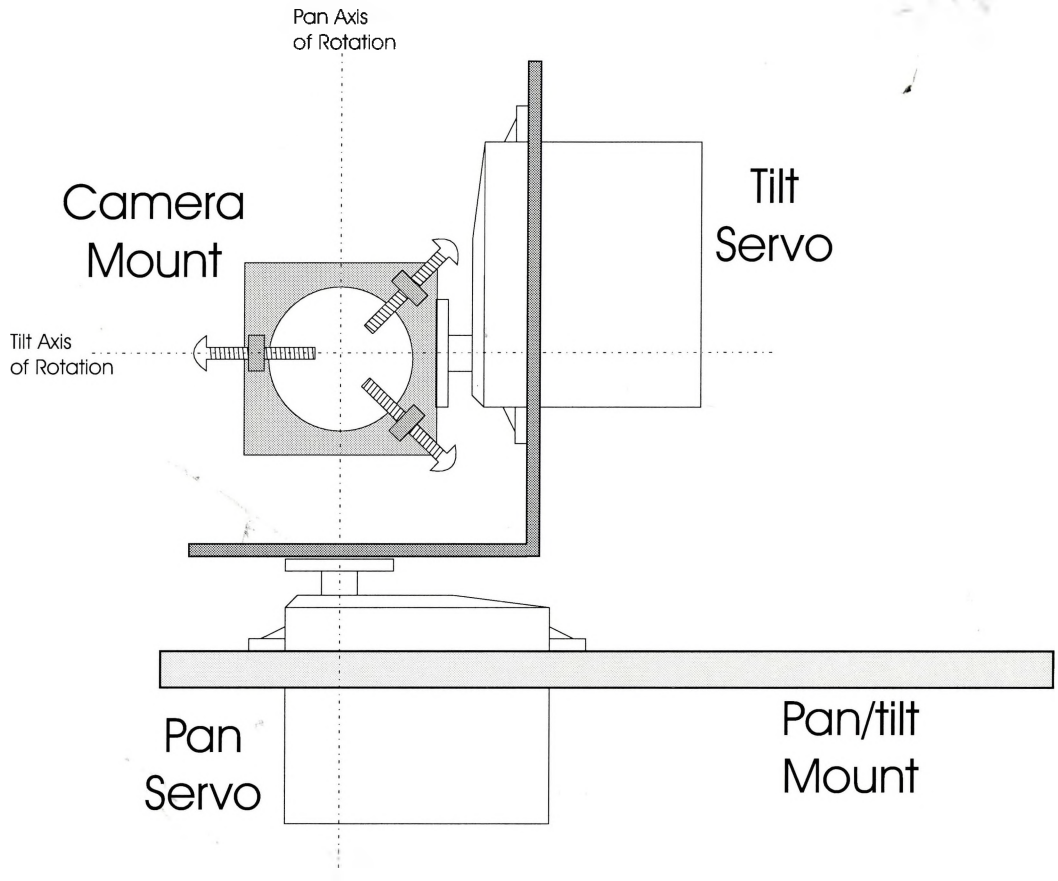

Figure 2.6: Mechanical Design of the Pan/tilt Camera Unit

control, the pan/tilt camera unit is well suited for active vision applications where precise absolute positioning is not critical.

2.2 Camera Calibration

2.2.1 Introduction

To take measurements from the information contained in an image from a camera, there must be a model of the camera describing how the image is formed. The simplest and perhaps most widely used model is the *pinhole* camera model. The pin hole camera model describes how light projects onto the image plane and is further discussed in Appendix B. However, no camera conforms to the model perfectly. In practice, rays of light do not project onto the image plane as predicted by the pinhole camera model. Most lenses come with an adjustable iris which further complicates matters. The iris size determines the amount of light that reaches the image plane and also affects the depth of field. As the iris gets smaller, i.e. approaching the ideal pinhole, the depth of field increases toward infinity. To compensate for lens distortions and finite depth of field, cameras are usually calibrated by finding parameters to correct the distorted image. Simple single element lenses often produce the most distorted images. Although compound lenses have additional elements to correct for lens distortions, none are perfect. Calibration of a camera to correct the effects of lens imperfections on the camera model is referred to as *interior orientation* and is discussed in §2.2.2.

Whereas the interior orientation concerns the effects of lens distortions, the *exterior orientation* concerns the exterior characteristics. This is also known as finding the *pose* of the camera. That is, determining the location and the relative orientation of the camera in the world coordinate system (WCS). The 3D tracking system of this

thesis is largely dependent upon an accurate measurement of the exterior orientation. Exterior Orientation is described in §2.2.3.

Once the camera interior and exterior characteristics have been determined, the pan/tilt servo mechanism must be calibrated as triangulation is based upon the angles created by the pan/tilt positions. Pan/tilt calibration is discussed in §2.3.

2.2.2 Interior Orientation

Interior orientation is frequently referred to as finding the *intrinsic* parameters of the camera system. This process can often be difficult and is widely discussed in much of the literature. Application specific requirements dictate the terms of the inevitable compromise between computational speed and accuracy.

The most accurate way of obtaining the intrinsic parameters is to use a nonlinear optimization method such as in [3] and [37]. This facilitates the use of any arbitrarily accurate and complex models for the camera system. However, most nonlinear optimizations require an accurate initial guess to start and require a computer-intensive nonlinear search.

Brown [3] used plumb (straight) lines to determine the intrinsic parameters, K_1, K_2, K_3, representing *radial* distortion and H_1, H_2, H_3, representing *decentering* distortion. The image of physically plumb lines through an imperfect lens will not be straight lines. Hence, it is possible to determine the values of the above parameters in such a way as to correct the lines in the image. Artificial or natural lines can be used. However, in practice it is common to use an artificial object with lines to ensure consistent calibration results when using more than one camera. Figure 2.7 shows an example of radial distortion. In Fig. 2.7a, dr is the radial distortion and dt is the tangential distortion. The solid box in Fig. 2.7b has no distortion whereas the dashed box inside is the same box with negative radial distortion and the dashed box outside has positive radial distortion.

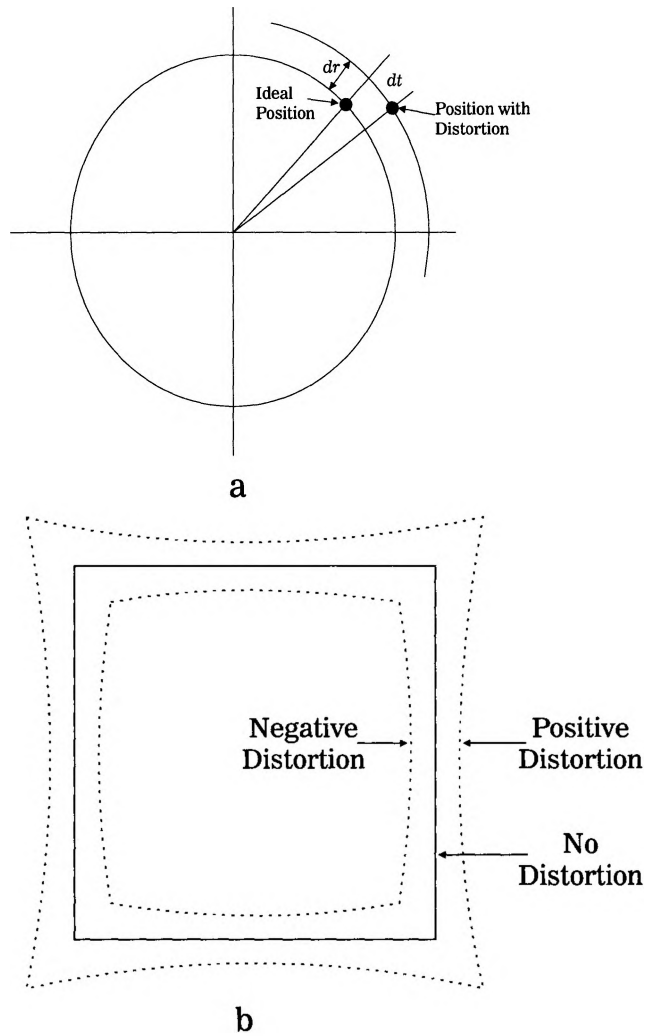

Figure 2.7: Radial Distortion

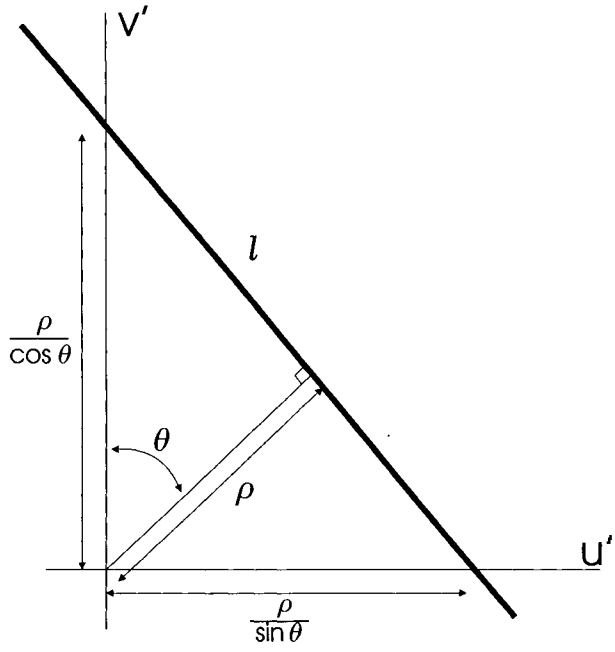

Figure 2.8: Plumb Line Method

Brown describes the equation of an arbitrary line, l, on a plane, as seen in Fig. 2.8, as:

$$u' \sin\theta + v' \cos\theta = \rho \tag{2.1}$$

where ρ denotes the distance along the normal of l passing through the origin and θ is the angle between the normal and the v'-axis. However l on the image plane will not be straight. The point $(u,\ v)$ on a line on the image plane can be corrected by using:

$$u' = u + \bar{u}(K_1 r^2 + K_2 r^4 + K_3 r^6 + ...) + [H_1(r^2 + 2\bar{u}^2) + 2H_2\bar{u}\bar{v}][1 + H_3 r^2 + ...], \tag{2.2}$$

$$v' = v + \bar{v}(K_1 r^2 + K_2 r^4 + K_3 r^6 + ...) + [2H_1\bar{u}\bar{v} + H_2(r^2 + 2\bar{v}^2)][1 + H_3 r^2 + ...], \tag{2.3}$$

in which $(u_p,\ v_p)$ is the principal point and:

$$\begin{aligned} \bar{u} &= u - u_p\ , \\ \bar{v} &= v - v_p\ , \\ r &= [(u-u_p)^2 + (v-v_p)^2]^{1/2}. \end{aligned} \tag{2.4}$$

Letting (u_{ij}, v_{ij}) be the jth point on the ith line and substituting into Eq. 2.2, 2.3 and 2.4, gives an observational equation of the form:

$$f(u_{ij}, v_{ij}, u_p, v_p, K_1, K_2, K_3, H_1, H_2, H_3, \theta_i, \rho_i) = 0. \tag{2.5}$$

A total of $8+2m$ equations in the form of Eq. 2.5 from m plumb lines are used to solve for the radial and decentering distortion coefficients iteratively, in a least-squares sense. The order of the resulting normal equations is also $8 + 2m$ and thus increases linearly with the number of lines m. Ordinarily, this would set a practical limit on the number of lines that can be processed simultaneously. However the block diagonality of the matrix of normal equations can be exploited to generate a practical algorithm for any number of lines. Brown's particular method however, does not solve for the extrinsic parameters as is common with iterative optimization techniques.

Tsai, in [37], also uses a nonlinear optimization but solves for intrinsic and extrinsic parameters. The pinhole camera model used in Tsai [37] includes:

f	effective focal length,
κ_1	1st order radial lens distortion,
$C_x,\ C_y$	coordinates of center of radial lens distortion,
s_x	scale factor to account for any uncertainty due to frame grabber horizontal scanline resampling,
R	rotational orientation with respect to the WCS, and
$\vec{t} = [t_x, t_y, t_z]^T$	translational orientation with respect to the WCS.

This method uses N coplanar points with known inter-point spacing where N is much larger than five, or N non-coplanar points with known locations where N is much larger than seven. Using either method, results show that $N > 60$ produces best results.

The parameters listed above in the pinhole camera model are described by the following equations. The rigid body transformation from the object world coordinate system $({}^{W}P_x, {}^{W}P_y, {}^{W}P_z)$ to the camera 3D coordinate $({}^{C}P_x, {}^{C}P_y, {}^{C}P_z)$ is:

$$\begin{bmatrix} {}^{C}P_x \\ {}^{C}P_y \\ {}^{C}P_z \end{bmatrix} = R \begin{bmatrix} {}^{W}P_x \\ {}^{W}P_y \\ {}^{W}P_z \end{bmatrix} + \vec{t}, \tag{2.6}$$

where;

$$R \equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \; and \tag{2.7}$$

$$\vec{t} \equiv \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{2.8}$$

are the rotation and translation matrices respectively. See Appendix A for further details. The undistorted pinhole camera model perspective projection transforms the camera coordinate system point $({}^{C}P_x, {}^{C}P_y, {}^{C}P_z)$ to image coordinates (u, v):

$$u = f\frac{{}^{C}P_x}{{}^{C}P_z}, \tag{2.9}$$

$$v = f\frac{{}^{C}P_y}{{}^{C}P_z}. \tag{2.10}$$

The radial distortion is modeled with:

$$u_d + D_x = u, \tag{2.11}$$

$$v_d + D_y = v, \tag{2.12}$$

where (u_d, v_d) are the true distorted image coordinates and:

$$D_x = u_d(\kappa_1 r^2 + \kappa_2 r^4 + ...) , \quad (2.13)$$

$$D_y = v_d(\kappa_1 r^2 + \kappa_2 r^4 + ...) , \quad (2.14)$$

$$r = \sqrt{u_d^2 + v_d^2}. \quad (2.15)$$

Real image point (u_d, v_d) is transformed into the frame grabber sampled image coordinates (u_f, v_f) by:

$$u_f = \frac{s_x u_d}{d'_x} + C_x , \quad (2.16)$$

$$v_f = \frac{v_d}{d_y} + C_y , \quad (2.17)$$

where:

(u_f, v_f) row and column numbers of the image pixel,

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

$d'_x = d_x \frac{N_{cx}}{N_{fx}}$,

d_x center to center distance between adjacent sensor elements in X direction,

d_y center to center distance between adjacent sensor elements in Y direction,

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

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

The last four of these parameters can usually be obtained from the manufacturer specifications of the image device. The calibration procedure in Tsai [37] for non-coplanar points is outlined below in 8 steps.

Step 1

Compute (u_d, v_d) using Eq. 2.16, 2.17.

Step 2

Compute $t_y^{-1}s_xr_1$, $t_y^{-1}s_xr_2$, $t_y^{-1}s_xr_3$, $t_y^{-1}s_xt_x$, $t_y^{-1}r_4$, $t_y^{-1}r_5$, and $t_y^{-1}r_6$. For each calibration point i with $({}^WP_{i_x}, {}^WP_{i_y}, {}^WP_{i_z})$ as the 3D world coordinate and (u_{i_d}, v_{i_d}) as the modified image coordinate computed in Step 1, the seven parameters can be found by solving:

$$\left[\begin{array}{cccccccc} v_{i_d}{}^WP_{i_x} & v_{i_d}{}^WP_{i_y} & v_{i_d}{}^WP_{i_z} & v_{i_d} & -u_{i_d}{}^WP_{i_x} & v_{i_d}{}^WP_{i_y} & v_{i_d}{}^WP_{i_z} \end{array} \right] L = u_{i_d} \ , \quad (2.18)$$

where:

$$L \equiv \left[\begin{array}{cccccccc} t_y^{-1}s_xr_1 & t_y^{-1}s_xr_2 & t_y^{-1}s_xr_3 & t_y^{-1}s_xt_x & t_y^{-1}r_4 & t_y^{-1}r_5 & t_y^{-1}r_6 \end{array} \right]^T . \quad (2.19)$$

Step 3

Compute $|t_y|$. Let:

$$\begin{aligned} a_1 &= t_y^{-1}s_xr_1 \ , \\ a_2 &= t_y^{-1}s_xr_2 \ , \\ a_3 &= t_y^{-1}s_xr_3 \ , \\ a_4 &= t_y^{-1}s_xt_x \ , \\ a_5 &= t_y^{-1}r_4 \ , \\ a_6 &= t_y^{-1}r_5 \ , \ and \\ a_7 &= t_y^{-1}r_6. \end{aligned} \quad (2.20)$$

Then:

$$|t_y| = (a_5^2 + a_6^2 + a_7^2)^{-1/2}. \quad (2.21)$$

Step 4

Determine the sign of t_y. Assume the sign of t_y is positive. Then compute:

$$\begin{aligned} r_1 &= (t_y^{-1} r_1) \cdot t_y \ , \\ r_2 &= (t_y^{-1} r_2) \cdot t_y \ , \\ r_4 &= (t_y^{-1} r_4) \cdot t_y \ , \\ r_5 &= (t_y^{-1} r_5) \cdot t_y \ , \\ t_x &= (t_y^{-1} t_x) \cdot t_y \ , \\ x &= r_1^W P_x + r_2^W P_y + t_x \ , \ and \\ y &= r_4^W P_x + r_5^W P_y + t_y . \end{aligned} \tag{2.22}$$

If x and u_d have the same sign and y and v_d have the same sign then $sgn(t_y) = +1$, else $sgn(t_y) = -1$.

Step 5

Find s_x using:

$$s_x = (a_1^2 + a_2^2 + a_3^2)^{1/2} |t_y| \ . \tag{2.23}$$

Step 6

Compute $r_1, r_2, ...r_9, t_x$. Use:

$$\begin{aligned} r_1 &= a_1 \cdot t_y / s_x \ , \\ r_2 &= a_2 \cdot t_y / s_x \ , \\ r_3 &= a_3 \cdot t_y / s_x \ , \\ r_4 &= a_5 \cdot t_y \ , \\ r_5 &= a_6 \cdot t_y \ , \\ r_6 &= a_7 \cdot t_y \ , \ and \\ t_x &= a_4 \cdot t_y . \end{aligned} \tag{2.24}$$

Given r_i, $i \in \{1, ..., 6\}$, 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.

Step 7

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:

$$\begin{bmatrix} P_{i_y} & -d_y v_{i_f} \end{bmatrix} \begin{bmatrix} f \\ t_z \end{bmatrix} = w_i d_y v_{i_f} \ , \tag{2.25}$$

where $y_i = r_4 {}^W P_{i_x} + r_5 {}^W P_{i_y} + (r_6 \cdot 0) + t_y$ and $w_i = r_7 {}^W P_{i_x} + r_8 {}^W P_{i_y} + (r_9 \cdot 0)$. With N object calibration points, this yields an overdetermined system of linear equations that can be solved for the unknowns f and t_z.

Step 8

Compute the exact solution for f, t_z, κ_1 by using:

$$d'_y v_f + d_y v_f \kappa_1 r^2 = f \frac{r_4\ {}^W P_x + r_2\ {}^W P_y + r_3\ {}^W P_z + t_y}{r_7\ {}^W P_x + r_8\ {}^W P_y + r_9\ {}^W P_z + t_z}\ , \tag{2.26}$$

where $r = \sqrt{(s_x^{-1} d'_x u_f)^2 + (d_y v_f)^2}$.

Tsai [37] also outlines the procedure when using coplanar points. The process is similar. As it is difficult to acquire the camera model parameters as a reference to verify the results of a calibration method, it is common to use the calibrated model parameters and assess the accuracy of measurements of the real 3D world. Tsai [37] uses three types of measurements to assess the calibrated parameters:

- accuracy of 3D coordinate measurement obtained through stereo triangulation,
- radius of ambiguity in ray tracing, and
- accuracy of 3D measurement.

In the first test, the accuracy of camera calibration is assessed by comparing the difference between the known 3D coordinates of the test points to the coordinates derived from using two calibrated cameras in a stereo configuration. The second test involves using the calibrated parameters to back project from the origin through the image plane to where the actual physical point lies and measuring the difference between the actual point and the projected point. The difference is referred to by Tsai as the *radius of ambiguity*. The third test measures physical parameters of a test object through a single calibrated camera and compares the results to the actual physical parameters. In the testing, Tsai [37] reports the best average error to be 0.4 thousands of an inch and a maximum error of 1.8 thousands of an inch. The test object was 1 inch square and the total depth range was 4.5 inches. The time it took to calibrate the camera was 1.5sec on a 68000-based MASSCOMP minicomputer circa 1987.

Linear techniques involve using the perspective transformation and usually a geometric property of the image objects [36, 40]. Linear techniques can be computed easily and quickly but have several disadvantages. Not being able to estimate parameters used in nonlinear models of the lens such as radial distortion is one disadvantage. (See Fig. 2.7 for an example of radial distortion.) A second disadvantage stems from the redundant parameterization of the camera model that can lead the algorithm to produce a fit between experimental observations and the model when in fact there is no fit. The possibility of mismatch increases with noisy data. As discussed in the next section, a linear technique is used to find the exterior parameters.

In some applications interior orientation is not an issue because the intrinsic parameters can be found in advance and as long as the camera system is not altered, the parameters need not to be recalculated. However, if a complex model is used and parameters are found a priori, the subsequent image processing includes using the parameters to correct the image. In this thesis, these computations are avoided completely by reducing the dependence on the image. The tracking method, discussed in detail in §3.2, is not affected by factors such as radial lens distortion. Furthermore, the image is not used directly in the calculation of the 3D location of the target as would occur in multiple-baseline stereo methods.

The only intrinsic parameter required in this system is the camera constant f which is used in all camera models and cannot be eliminated. The camera constant is only used in the exterior orientation and was experimentally determined using Tsai's method [37].

2.2.3 Exterior Orientation

Exterior orientation is often referred to as the *pose* of the camera and can be represented by a *homogeneous transform*. The camera pose is the transform, ${}^{W}_{C}T$, from

the camera coordinate system to the world coordinate system:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where r_{nm}, $n, m \in \{1, 2, 3\}$ are the elements of the rotation matrix, ${}^{W}_{C}R$, and $[t_x\ t_y\ t_z]^T$ is the translation vector ${}^{w}_{C}\vec{t}$. Refer to Appendix A for further details. Each camera coordinate system has its origin located at the focal point with the positive Z-axis pointing out from the camera along the optical axis. The rotation matrix elements are derived from the rotational angles γ, β, α, corresponding to yaw, pitch and roll and are also known as Euler angles. Hence, to find the exterior orientation, γ, β, α, t_x, t_y, and t_z, must be found.

A linear technique was chosen to calibrate the extrinsic parameters. The nonlinear methods are often not practical for real-time applications due to the computational requirements of the optimization. In this thesis, run-time pose estimation was considered and methods were chosen appropriately. Hence nonlinear methods were not attempted. Although the run-time exterior calibration was not implemented, it is viewed as a future possibility.

The camera location determination problem was formally defined by Fischler and Bolles [10], as follows: "Given a set of m control points, whose three-dimensional coordinates are known in some coordinate frame, and given an image in which some subset of the m control points is visible, determine the location (relative to the coordinate system of the control points) from which the image was obtained." There has been a wealth of solutions to this problem. In most cases a linear solution is used and in some cases the linear solution is further refined with a nonlinear optimization.

Chen [4], proposed a method which limited the control points to form a plane and a line. By further restricting the line to be perpendicular to the plane, the problem

reduces to only having to solve for 2 of the 6 extrinsic parameters.

Liu *et al.* [25], proposed a method whereby the control points form straight lines. First, the rotation matrix is found by a linear algorithm using 8 or more line correspondences, or by a nonlinear algorithm using 3 or more line correspondences, where the line correspondences are given or derived from point correspondences. Then, the translation vector can be obtained by solving a set of linear equations based on 3 or more line correspondences, or 2 or more point correspondences. The algorithm however, only works if the three Euler angles (rotation) are less than 30 degrees.

Horaud *et al.* [16], gives a solution to the perspective 4-point problem. The perspective 4-point problem is an elegant linear solution to determine the exterior orientation. Here only 4 non-coplanar point correspondences are required. The paper derives an analytic solution in the form of a biquadratic polynomial in one unknown. Four non-coplanar points are equivalent to a pencil of 3 non-coplanar lines which are used to represent the object coordinate frame. Two transformation matrices A_1 and A_2 are generated that relate the object coordinate frame defined by the 4 points, the image frame and the camera frame. Referring to Fig. 2.9:

$$A_1 = \begin{bmatrix} k'_x & (P_3)_x & (k' \times P_3)_x & (FJ)_x \\ k'_y & (P_3)_y & (k' \times P_3)_y & (FJ)_y \\ k'_z & (P_3)_z & (k' \times P_3)_z & (FJ)_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.27}$$

$$A_2 = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & d_x \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.28}$$

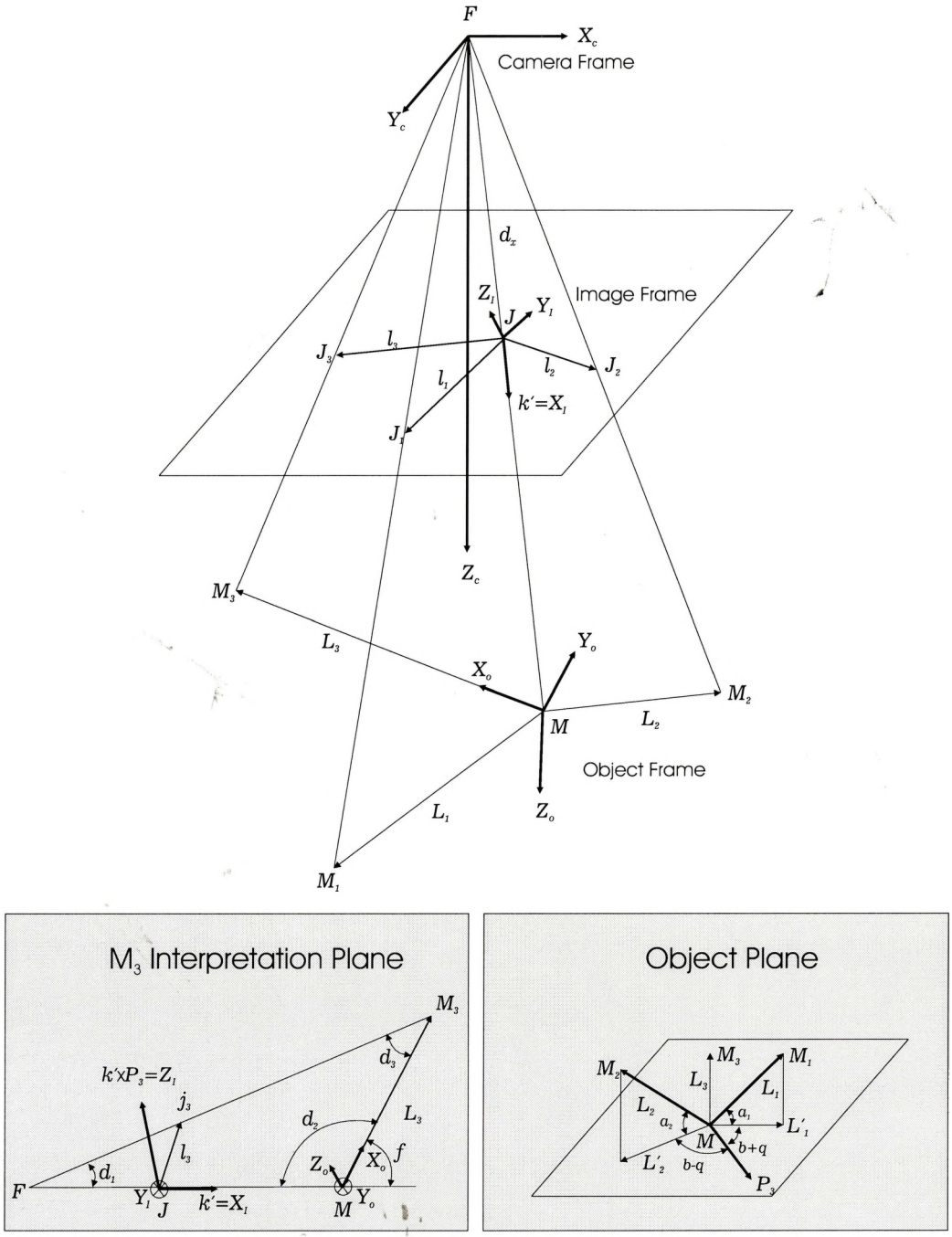

Figure 2.9: 4-Point Resection Solution

where:

$$k' = \frac{FJ}{\| FJ \|},$$
$$P_3 = \frac{l_3 \times k'}{\| l_3 \times k' \|}, \text{ and}$$
$$k' \times P_3 = \frac{k' \times (l_3 \times k')}{\| l_3 \times k' \|},$$

are all given knowing the 4-point correspondences. Horaud *et al.* [16] arbitrarily defines the object frame's X-axis to be along L_3. Furthermore, F with M and M_1, M_2 or M_3 form three planes referred to as *interpretation planes.* The M_3 interpretation plane shown in Fig. 2.9, is defined by F, M and M_3. P_3 is a vector that lies on the object plane, is normal to the M_3 interpretation plane and is considered to be the Y-axis of the object frame. To find ϕ, a biquadratic polynomial in terms of $\cos\phi$ is derived:

$$I_1 \cos^4\phi + I_2 \cos^3\phi + I_3 \cos^2\phi + I_4 \cos\phi + I_5 = 0. \tag{2.29}$$

In Eq. 2.29, $I_1 ... I_5$ are all functions of γ_1, γ_2, α_1, α_2, and β. In A_2:

$$d_x = \| FM \| - \| FJ \|, \tag{2.30}$$

where:

$$\| FM \| = \frac{\| FJ \| \| FJ_3 \times MM_3 \|}{\| FJ \times FJ_3 \|}, \text{ and} \tag{2.31}$$

$$\| MM_3 \| = \| FM \| \cos\delta_2 + \| FM_3 \| \cos\delta_3 . \tag{2.32}$$

Simplified versions to the solution are proposed for three cases: three colinear image points, a right vertex and four coplanar points. The last is the same problem solved in [6] where the pencil of three lines is formed from house corners where it can be assumed the lines are orthogonal. The method presented by Horaud *et al.* [16] was

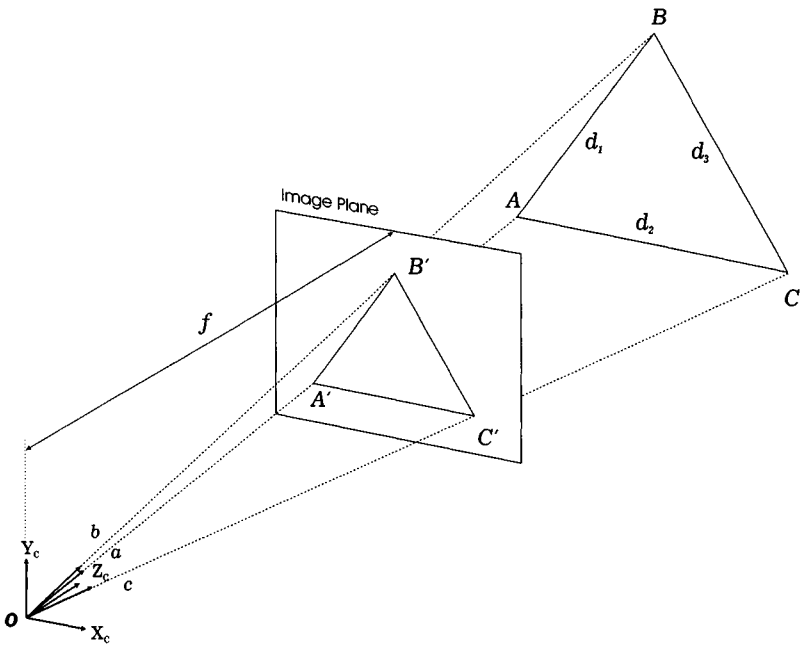

Figure 2.10: The Perspective Projection of a Triangle Onto the Image Plane

attempted as a possible solution to finding the extrinsic parameters but it failed to produce satisfactory results.

The 4-point perspective solution is advantageous because the four required points can be derived in almost any way. Hence a slight variation on the 4-point perspective problem was developed. Instead of directly solving the four point resection problem, a 3-point resection solution was used in conjunction with a fourth point to produce a unique solution. The solution by Linnainmaa *et al.* [24] to the 3-point resection problem is described below.

Triangle ABC, in the WCS, projects onto the image plane as $A'B'C'$, as shown in Fig. 2.10. The unit vectors in the directions OA', OB', and OC' are **a**, **b** and **c**, respectively. The lengths of the triangle sides AB, AC, and BC are d_1, d_2, d_3 and must be known in advance. The unknown distances from the origin, OA, OB, and

OC, are x, y, z, respectively. Then:

$$\begin{cases} (x\mathbf{a} - y\mathbf{b}) \cdot (x\mathbf{a} - y\mathbf{b}) = d_1^2 \\ (x\mathbf{a} - z\mathbf{c}) \cdot (x\mathbf{a} - z\mathbf{c}) = d_2^2 \\ (y\mathbf{b} - z\mathbf{c}) \cdot (y\mathbf{b} - z\mathbf{c}) = d_3^2 \ , \end{cases} \tag{2.33}$$

where the $(\cdot)$ represents the inner product operation. Simplifying Eq. 2.33 with $\mathbf{a} \cdot \mathbf{b}$, $\mathbf{a} \cdot \mathbf{c}$, and $\mathbf{b} \cdot \mathbf{c}$ as p_1, p_2, and p_3, respectively yields:

$$\begin{cases} x^2 - 2p_1xy + y^2 = d_1^2 \\ x^2 - 2p_2xz + z^2 = d_2^2 \\ y^2 - 2p_3yz + z^2 = d_3^2. \end{cases} \tag{2.34}$$

Equation 2.34 now takes the form:

$$\begin{cases} q_1x^2 + u^2 = d_1^2 \\ q_2x^2 + v^2 = d_2^2 \\ q_3x^2 + 2p_3uv + q_4 = q_5xu + q_6xv \ , \end{cases} \tag{2.35}$$

using:

$$u = y - p_1x \ , \tag{2.36}$$

$$v = z - p_2x \ , \tag{2.37}$$

and coefficients:

$$\begin{aligned} q_1 &= 1 - p_1^2 \ , \\ q_2 &= 1 - p_2^2 \ , \\ q_3 &= 2(p_1^2 + p_2^2 - p_1p_2p_3 - 1) \ , \\ q_4 &= d_1^2 + d_2^2 - d_3^2 \ , \\ q_5 &= 2(p_2p_3 - p_1) \ , and \\ q_6 &= 2(p_1p_3 - p_2). \end{aligned} \tag{2.38}$$

Manipulating the expressions of Eq. 2.35 yields:

$$r_1 x^4 + r_2 x^2 + r_3 = (r_4 x^2 + r_5)uv \ , \tag{2.39}$$

where:

$$\begin{aligned} r_1 &= q_3^2 + 4q_1 q_2 p_3^2 + q_1 q_5^2 + q_2 q_6^2 \ , \\ r_2 &= 2q_3 q_4 - 4(d_1^2 q_2 + d_2^2 q_1)p_3^2 - d_1^2 q_5^2 - d_2^2 q_6^2 \ , \\ r_3 &= 4d_1^2 d_2^2 p_3^2 + q_4^2 \ , \\ r_4 &= 4p_3 q_3 + 2q_5 q_6 \ , and \\ r_5 &= 4p_3 q_4. \end{aligned} \tag{2.40}$$

Finally, from Eq. 2.39:

$$s_1 x^8 + s_2 x^6 + s_3 x^4 + s_4 x^2 + s_5 = 0 \ , \tag{2.41}$$

where:

$$\begin{aligned} s_1 &= r_1^2 - q_1 q_2 r_4^2 \ , \\ s_2 &= 2r_1 r_2 + (d_1^2 q_2 + d_2^2 q_1)r_4^2 - 2q_1 q_2 r_4 r_5 \ , \\ s_3 &= r_2^2 + 2r_1 r_3 - d_1^2 d_2^2 r_4^2 + 2(d_1^2 q_2 + d_2^2 q_1)r_4 r_5 - q_1 q_2 r_5^2 \ , \\ s_4 &= 2r_2 r_3 - 2d_1^2 d_2^2 r_4 r_5 + (d_1^2 q_2 + d_2^2 q_1)r_5^2 \ , and \\ s_5 &= r_3^2 - d_1^2 d_2^2 r_5^2. \end{aligned} \tag{2.42}$$

Hence, Eq. 2.41 must be solved for x. Equation 2.41 is an 8^{th} order equation in x^2. Therefore, there is a maximum of four solutions for x since only nonnegative values are valid. Once x has been determined, it is easy to determine y and z using Eq. 2.33. To obtain a unique solution for x, subsets of three of the four points are used to create sets of solutions [31]. Each of the four triangles will produce four solutions. The common solution from the sets will give the location of each of the points. This method has two

drawbacks. First, the fourth order polynomial may not be easily solved. Secondly, the common solution may not be obvious. This method does however, improve on the three point resection by averaging the redundant information provided by the fourth point.

2.3 Pan/Tilt Calibration

For feature point tracking the servo pan/tilt mechanism must be calibrated. Two parameters are necessary:

PanCal number of degrees per servo position, and
TiltCal number of degrees per servo position.

Each servo is rotated through a physically measured number of degrees. Dividing the number of degrees by the change in pulse width yields the above parameters. This assumes the servo positioning is linear and will be discussed in Chapter 4.

Servo positioning error was determined based on the image information. The servo PWM signal is perturbed by the smallest amount and increased until there is a change in the image. Assuming no objects in the image are moving, a change in the image means the servo must have moved the camera. This gives an indication as to the minimum resolution of the pan/tilt camera units and the error in positioning.

2.4 Summary

Calibration of interior and exterior paramters has many issues that are application dependent. In this thesis, the dependence has been minimized by using the image information only to track the target and not to calculate depth. Rather, the depth is calculated from the pan/tilt angles from two pan/tilt camera units in a form of

triangulation. The error in the 3D position measurement caused by the pan/tilt camera unit should be constant and less significant than errors caused by obtuse and acute angles in the triangulation. This will be discussed further in Chapter 4.

Chapter 3

3D Point Determination

3.1 Introduction

Determining the 3D location of the feature point in space is dependent on the tracking of the feature point by each pan/tilt camera unit. The tracking system controls the pan/tilt unit maintaining the feature point in the center of the image frame, $(u, v) = (0, 0)$, as seen in Fig. 3.1. A separate algorithm monitors the pan/tilt angles to determine where the optical axis (Z-axis of the camera coordinate frame) is in the WCS. The camera i coordinate frame (X_i, Y_i, Z_i), shown with black lines, represents the current orientation with respect to the WCS. The corresponding red coordinate system represents the calibrated orientation of each camera and is calculated by the pan/tilt angles. Section 3.2 outlines the method used to track the feature point.

As the camera tracks the feature point, the feature point must lie on the optical axis. Hence, with two or more cameras it is possible to determine the 3D location of the feature point in the WCS. The key is finding where the optical axes intersect. It is unlikely that the optical axes will cross in 3D. Section 3.3 describes how the point of intersection is estimated.

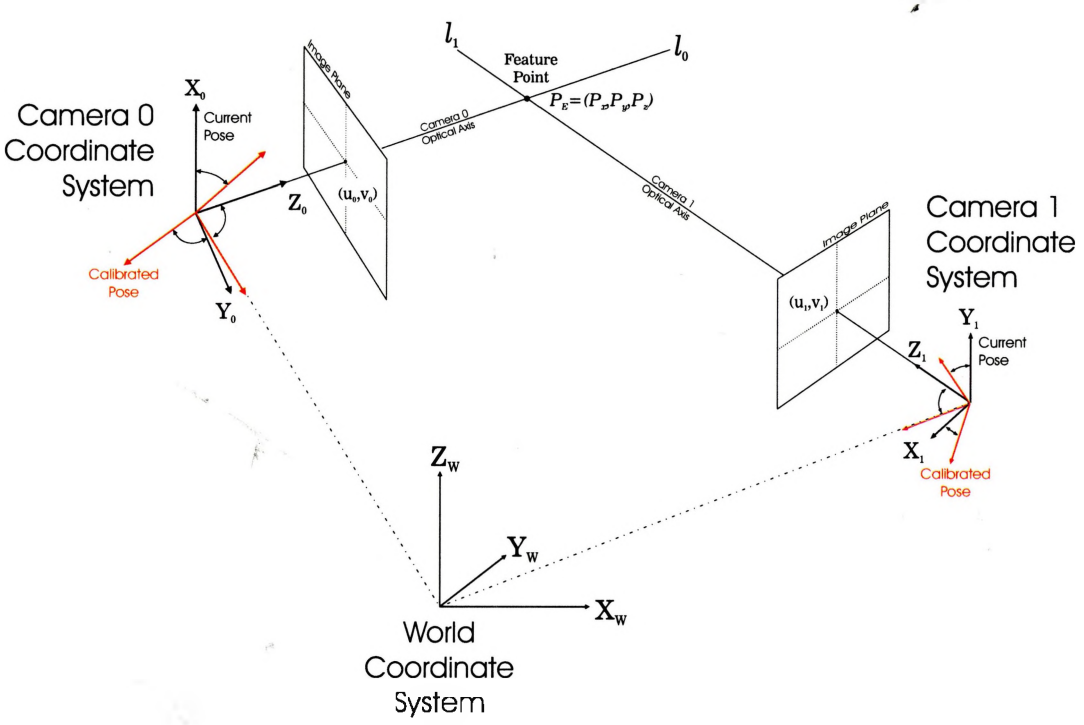

Figure 3.1: Pan and Tilt of Camera Coordinate Systems to Track Feature Point

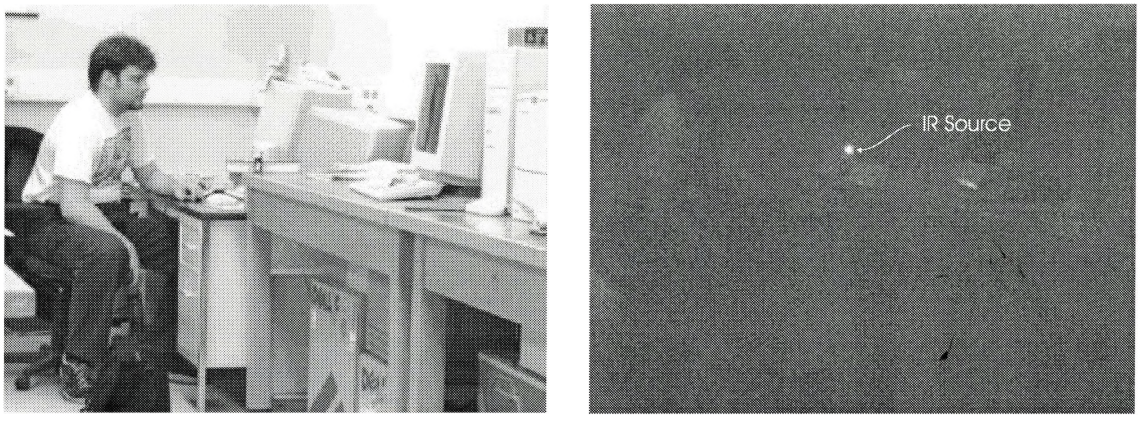

Figure 3.2: The Effect of Using a Light Filter

3.2 Tracking

Tracking involves processing the image and determining the change in pan/tilt angles to center the feature point in the image. This forces the Z-axis of the camera coordinate frame, which is also the optical axis of the camera, to pass through the feature point. Using *PanCal* and *TiltCal* which are the calibrated parameters to convert the servo positions to pan and tilt angles respectively, the pose of the camera can be determined. From the pose, the Z-axis can be represented as a line in WCS.

An infrared point source was used as the feature point to simplify the tracking algorithm. Adding a visible light filter to the front of the camera lens reduces the light from the background and allows any infrared light sources to stand out. Figure 3.2 shows the contrast in using a light filter. Figure 3.2a shows an image with no filter and Fig. 3.2b is the same image with the visible light filter. The labeled dot in Fig. 3.2b is the infrared light source and is present in both images. Once the image is digitized, a threshold is used to eliminate all but the white pixels. Then for each

white pixel, presumably from the infrared source, the centroid, (C_u, C_v), is found by:

$$C_u = \frac{\sum_{i=1}^{n} P_{i_u}}{n}, \qquad C_v = \frac{\sum_{i=1}^{n} P_{i_v}}{n}, \tag{3.1}$$

where n is the total number of white pixels and P_{i_u} and P_{i_v} are the u and v image coordinates of point P_i respectively. The pixel distance from the centroid to the center of the image is then used to update the servo position.

3.2.1 Determining the Current Pose

The calibrated or initial pose of a camera coordinate system with respect to the WCS is found during the calibration proceedure and represented by:

$$T_{CAL} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{CAL_x} \\ r_{21} & r_{22} & r_{23} & t_{CAL_y} \\ r_{31} & r_{32} & r_{33} & t_{CAL_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For a change of pan angle, ω, and change in tilt angle, θ, the *change in pose* from the calibrated position is represented by the homogeneous transform matrix:

$$T_{CH} = \begin{bmatrix} \cos\omega & \sin\omega\sin\theta & \sin\omega\cos\theta & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ -\sin\omega & \cos\omega\sin\theta & \cos\omega\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

which is the transform for a rotation about the X and Y-axes. Upon a pan/tilt movement the *current pose*, T_{CUR}, of the camera with respect to the WCS can be found by taking the calibrated pose T_{CAL} and multiplying by T_{CH}:

$$T_{CUR} = T_{CH} T_{CAL} \,. \tag{3.2}$$

3.2.2 Pose Error due to Misalignment of Camera

Pose error might be expected if the camera coordinate system origin is not at the center of rotation of the pan/tilt mechanism. However, this is not the case. Indeed, the translation $\vec{t}$ of the homogeneous transform representing the pose would be effected by a misalignment, but the rotation R, is invariant to misalignment. (See Appendix A for explanation of $\vec{t}$ and R.) For example, let T_P be an arbitrary pose:

$$
\begin{aligned}
T_P &= T_{CH} T_{CAL} \\
&= \begin{bmatrix} \cos\omega & \sin\omega\sin\theta & \sin\omega\cos\theta & t_{mis_x} \\ 0 & \cos\theta & -\sin\theta & t_{mis_y} \\ -\sin\omega & \cos\omega\sin\theta & \cos\omega\cos\theta & t_{mis_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{CAL_x} \\ r_{21} & r_{22} & r_{23} & t_{CAL_y} \\ r_{31} & r_{32} & r_{33} & t_{CAL_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{3.3}
$$

In Eq. 3.3, $[t_{mis_x}\ t_{mis_y}\ t_{mis_z}]^T$ is the translation due to the misalignment of the camera origin and the pan/tilt center of rotation. The third column of T_P would then be:

$$
\begin{bmatrix} r_{13}\cos\omega + r_{23}\sin\omega\sin\theta + r_{33}\sin\omega\cos\theta \\ r_{23}\cos\theta - r_{33}\sin\theta \\ -r_{13}\sin\omega + r_{23}\cos\omega\sin\theta + r_{33}\cos\omega\cos\theta \\ 0 \end{bmatrix}. \tag{3.4}
$$

It can easily be seen that the third column is not effected by $[t_{mis_x}\ t_{mis_y}\ t_{mis_z}]^T$. This also holds true for the first and second columns of T_P as well. However, the fourth column of T_P is:

$$
\begin{bmatrix} t_{CAL_x}\cos\omega + t_{CAL_y}\sin\omega\sin\theta + t_{CAL_z}\sin\omega\cos\theta + t_{mis_x} \\ t_{CAL_y}\cos\theta - t_{CAL_z}\sin\theta + t_{mis_y} \\ -t_{CAL_x}\sin\omega + t_{CAL_y}\cos\omega\sin\theta + t_{CAL_z}\cos\omega\cos\theta + t_{mis_z} \\ 1 \end{bmatrix}. \tag{3.5}
$$

Thus, the misalignment adds to the translation $\vec{t}_P$.

As will be seen in §3.3, the equation of the line representing the optical axis is fully given by the third and fourth columns of T_P. The third column is the direction of the line and the fourth column is a point on the line in the WCS. Due to the long distances (meters) from the cameras to the feature point as compared to the misalignment (millimeters), the equation of the line is more sensitive to directional error. Thus, a misalignment between the camera origin and pan/tilt center of rotation only has a small affect in the representation of the optical axes.

3.3 Intersection Estimation

Using only one camera it is not possible to find the location of the feature point in the WCS since the point could lie anywhere on the optical axis. Using two cameras to track the same feature point, the optical axes should intersect at the feature point location assuming the axes are not colinear. The problem now is to find where the two optical axes intersect.

The likelihood of the two optical axes intersecting at a point in 3D is very small. Referring to Fig. 3.1 and Fig. 3.3, the problem is to determine P_E, the estimate of the point of intersection. For the case of two cameras, P_E is assumed to be midway along the shortest line, l_p, that connects the two optical axes and is perpendicular to both optical axes. The method for finding its midpoint follows.

Let T_0 be the homogeneous transform from the camera to the WCS (current pose) for camera 0 and similarly T_1 the current pose of camera 1. Furthermore, let $r_{0_{nm}}$ where $n, m \in \{1, 2, 3\}$ and $\vec{t_0} = [t_{0_x}\ t_{0_y}\ t_{0_z}]^T$ be the rotational components and translation of T_0 and let $r_{1_{nm}}$ and $\vec{t_1} = [t_{1_x}\ t_{1_y}\ t_{1_z}]^T$ be the rotational components and translation of T_1. Since the columns of the rotation matrix are the projections of the camera frame unit vectors onto the world coordinate system, the normalized

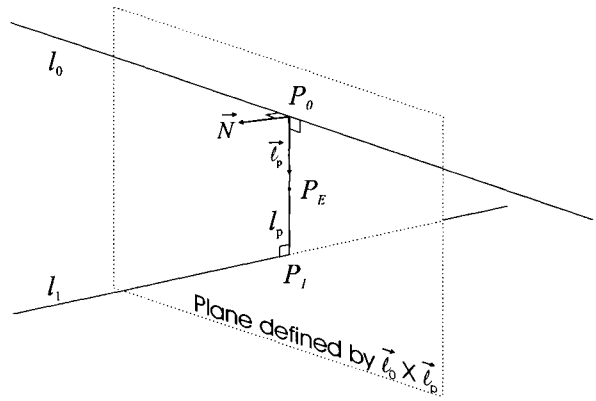

Figure 3.3: Determining the Intersection of Two Lines in 3D

direction of the Z-axis of the camera frame (also the optical axis) can be defined as:

$$\begin{aligned} \vec{\ell}_0 &= [r_{0_{13}}\ r_{0_{23}}\ r_{0_{33}}]^T\ , \\ \vec{\ell}_1 &= [r_{1_{13}}\ r_{1_{23}}\ r_{1_{33}}]^T\ , \end{aligned} \tag{3.6}$$

for cameras 0 and 1 respectively. Henceforth, let the lines that are the optical axes of camera 0 and camera 1 be referred to as l_0 and l_1 respectively. Therefore, the direction of l_0 is $\vec{\ell}_0$ and the direction of l_1 is $\vec{\ell}_1$. Then, the cross product of vectors $\vec{\ell}_0$ and $\vec{\ell}_1$ will give the direction of l_p:

$$\vec{\ell}_p = \frac{\vec{\ell}_0 \times \vec{\ell}_1}{\|\vec{\ell}_0 \times \vec{\ell}_1\|}\ . \tag{3.7}$$

Defining the plane in which both l_0 and l_p lie, requires the normal to the plane given by:

$$\vec{N} = \frac{\vec{\ell}_0 \times \vec{\ell}_p}{\|\vec{\ell}_0 \times \vec{\ell}_p\|}\ . \tag{3.8}$$

The point P_1 is where l_1 intersects the $\vec{\ell}_0 \times \vec{\ell}_p$ plane. The line l_p is completely defined

by $\vec{\ell}_p$ and the point P_1. Using the parametric form, l_1 can be represented in 3D as:

$$l_1 \begin{cases} P_{1_x} = t_{1_x} + dr_{1_{13}} \\ P_{1_y} = t_{1_y} + dr_{1_{23}} \qquad \infty^- < d < \infty^+ \\ P_{1_z} = t_{1_z} + dr_{1_{33}} \ , \end{cases} \tag{3.9}$$

and the plane as:

$$\vec{N}_x P_{1_x} + \vec{N}_y P_{1_y} + \vec{N}_z P_{1_z} + q = 0 \ , \tag{3.10}$$

where $q = \vec{N} \cdot \vec{t_0}$. Solving for d yields:

$$d = \frac{-(\vec{N} \cdot \vec{t_1}) - q}{\vec{N} \cdot \vec{\ell}_1} \ . \tag{3.11}$$

Substituting the value of d from Eq. 3.11 into Eq. 3.9 gives the point P_1.

Finally, to find the midpoint P_E, along l_p between l_0 and l_1, the point of intersection P_0 of l_0 and l_p must be found. Hence:

$$l_0 \begin{cases} P_{0_x} = t_{0_x} + kr_{0_{13}} \\ P_{0_y} = t_{0_y} + kr_{0_{23}} \qquad \infty^- < k < \infty^+ \\ P_{0_z} = t_{0_z} + kr_{0_{33}} \ , \end{cases} \tag{3.12}$$

and:

$$l_p \begin{cases} P_{0_x} = P_{1_x} + j\vec{\ell}_{p_x} \\ P_{0_y} = P_{1_y} + j\vec{\ell}_{p_y} \qquad \infty^- < j < \infty^+ \\ P_{0_z} = P_{1_z} + j\vec{\ell}_{p_z} \ , \end{cases} \tag{3.13}$$

must be solved simultaneously to find k and j. Thus:

$$P_{1_x} + j\vec{\ell}_{p_x} = t_{0_x} + kr_{0_{13}} \ , \tag{3.14}$$

$$P_{1_y} + j\vec{\ell}_{p_y} = t_{0_y} + kr_{0_{23}} \ , \tag{3.15}$$

$$P_{1_z} + j\vec{\ell}_{p_z} = t_{0_z} + kr_{0_{33}} \ . \tag{3.16}$$

Using Eq. 3.14,

$$j = \frac{t_{0_x} - P_{1_x} + \frac{r_{013}}{r_{023}}(P_{1_y} - t_{0_y})}{\vec{\ell}_{p_x} - \frac{r_{013}}{r_{023}}\vec{\ell}_{p_y}} . \tag{3.17}$$

The midpoint $P_E = (P_{E_x},\ P_{E_y},\ P_{E_z})$ along l_p between l_0 and l_1 is found by substituting j from Eq. 3.17 in Eq. 3.13:

$$P_E \begin{cases} P_{E_x} = P_{1_x} + \frac{j}{2}\vec{\ell}_{p_x} \\ P_{E_y} = P_{1_y} + \frac{j}{2}\vec{\ell}_{p_y} \\ P_{E_z} = P_{1_z} + \frac{j}{2}\vec{\ell}_{p_z} . \end{cases} \tag{3.18}$$

3.4 Camera Combinations

Using this method to find the estimate of the intersection of two lines in 3D, a multitude of pairs of cameras from a NOC could be used to find the location of the feature point in the WCS. For N cameras there are

$$1 + 2 + 3 + ... + (N - 1) \tag{3.19}$$

unique pairs of cameras. In this study, the feature location is calculated using all the pairs of cameras and the results are compared.

As the number of cameras in the network increases there is more redundant information. Although there are many ways this information can be used, it is beyond the scope of this work. For instance, instead of pairs of cameras, triplets could be used to improve the calculation. In fact, there may be ways of using any number of cameras simultaneously or even dynamically changing the number of cameras based on the situation.

Another possibility would be to choose the cameras that will most likely produce the best results. Two cameras with very little distance between them have a short baseline increasing the sensitivity of the feature location calculation to errors in pan

and tilt. Another situation, where the feature point passes directly in between two cameras, risks the optical axes becoming colinear. The algorithm would fail in this case. Careful selection of cameras from the network may produce better results.

3.5 Summary

In this section, the tracking system was described and a method to find the feature point in the WCS, based on intersecting the optical axes of two cameras, was outlined. The performance of this system was evaluated experimentally and is discussed in the following chapter.

Chapter 4

Experimental Results

4.1 Introduction

Four pan/tilt camera units were built to test the algorithms for tracking and feature point location. The resulting NOC was arranged in two configurations for evaluation. In one configuration (Table Top Test), the cameras view a small volume of about $2m^3$ while in the other configuration (Room Test), the network views a volume of approximately $16m^3$. In the tests, the tracking system tracks an infrared point source as described in the previous chapter. The results of the three tests are discussed in the Performance Section (§4.3).

4.2 Test Results

4.2.1 Table Top Test

In the first test, four pan/tilt camera units were arbitrarily placed within the WCS. Figure 4.1 shows the calibrated coordinates of the cameras. A precision XY-table was used to move the feature point around the XY-plane of the WCS in the shape of a

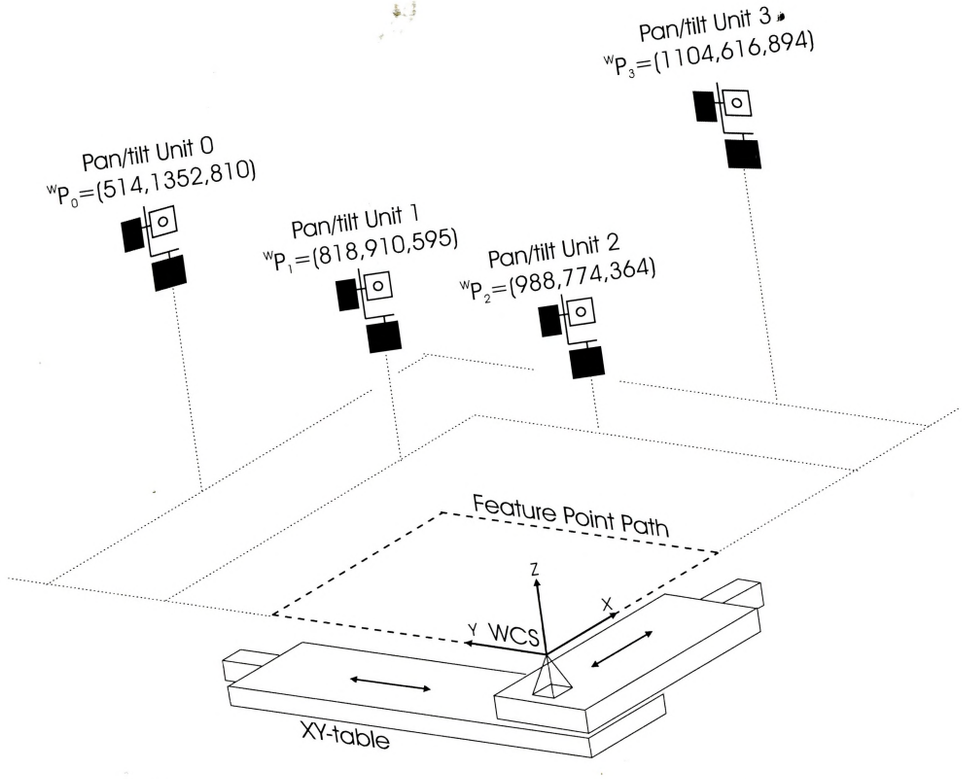

Figure 4.1: Table Top Test Setup

rectangle with length, 45cm, and width, 30cm. The feature point traveled at a rate of 25mm/sec while data was collected from the 6 possible pairs of cameras producing 283 data points per set. For clarity, the results from only 2 representative pairs of cameras, cameras 1-2 and cameras 2-3, are shown in Fig. 4.2.

Since this experimental setup did not have the ability to record the actual location of the feature point at the instant each image was captured, the absolute 3D error could not be determined. The error could only be determined in 2 dimensions. The rectangular test path lay in the XY-plane (z=0) and hence the error in the Z direction could easily be calculated. The ends of the rectangular test path should yield y=0 and the sides of the rectangular test path should yield x=0. Hence the error in Y can only be determined along the ends and the error in X along the sides. Summing the magnitude of the error in the Z direction and the X or Y direction gives the 2D error. The magnitude of the 2D error can be seen in Fig. 4.2b.

When the error plot of Fig. 4.2b is compared with the actual 3D locations in Fig. 4.2a, it is evident the error is not entirely due to noise in the system. The error plot for camera pair 1-2 reaches a maximum near the center of the chart. This corresponds to where the camera pair reaches a maximum in the Z direction in the 3D location plot. A second run of the same test showed similar results where points further away from the origin tended to have increased error. The results point toward inaccuracies in the servo calibration likely due to the assumption that the positioning is linear.

At points 38, 98, and 130 in the error plot Fig. 4.2b, there is a noticeable change in the magnitude of the error. This corresponds to a corner of the rectangular test path. Since the camera images are not captured synchronously it is impossible for more than one camera to have captured an image precisely at the time the feature point changed direction. This would produce the increased error as seen in the error plot at the points mentioned above.

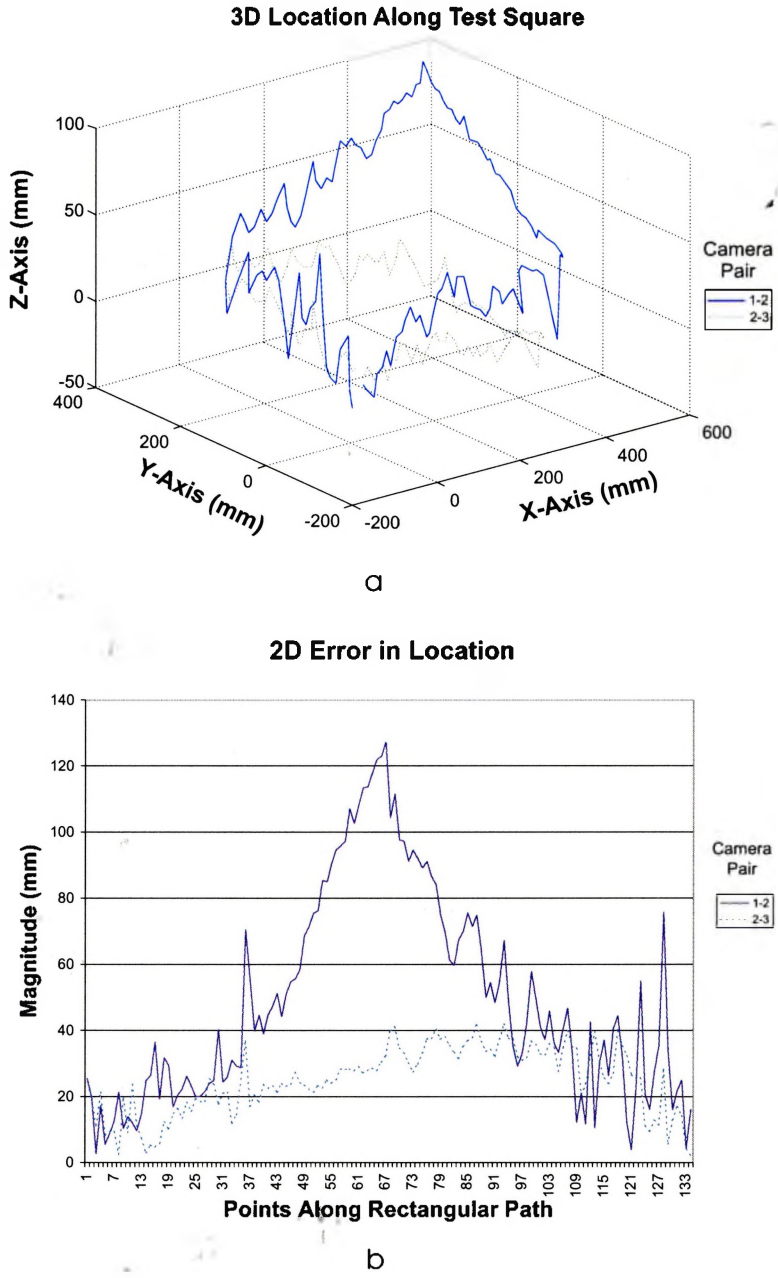

Figure 4.2: Results from Table Top Test

Figure 4.2b also shows increased noise in the error close to the origin which corresponds to either end of the plot. This was expected since the origin is furthest away from the cameras in this test. The calculated position of the feature point becomes increasingly sensitive to the error in the image analysis and servo positioning as the feature point travels further away from the cameras.

Table 4.3: Average Error (mm) from Table Top Test

Camera Pair	0-1	0-2	0-3	1-2	1-3	2-3
Average Error	29.8	20.5	42.3	46.9	88.2	24.1
Standard Deviation	11.4	12.0	29.2	30.8	59.1	10.1

The results from the four other pairs of cameras are not shown in Fig. 4.2 but produced similar results. The average error and standard deviation from each of the 6 pairs of cameras is shown in Table 4.3.

4.2.2 Room Test

In the second test, the cameras were distributed randomly around a volume that measured approximately 4m long, 2m wide and 2m high. The WCS origin was located near the middle of this volume. The locations of the cameras can be found in Table 4.4. To test the system, the feature point traveled 4m along the Y-axis of the WCS. Similar to the previous test, there was no method to determine the actual position of the feature point at the time each camera acquires an image. Hence, the results are formulated as a 2D error magnitude where the error in the Y direction is not factored in.

In Fig. 4.3, each line is an interpolation of the points collected from a camera pair. 37 points were collected per pair. Since the feature point traveled along a straight line parallel to the Y-axis, the lines on the Z_W *vs.* Y_W and X_W *vs.* Y_W coordinate

Table 4.4: Camera Position

Camera	(X,Y,Z) in mm
1	(888.3,-2497.2,524.8)
2	(-1349.1,-2537.4,610.8)
3	(-236.2,2248.1,366.1)
4	(2085.4,3894.6,906.8)

graphs should be horizontal straight lines.

The peak in the graphs at $Y > 2000$ was due to the feature point traveling outside the view of Camera 2. Thus, camera pairs 0-2, 1-2 and 2-3 produced erroneous values for $Y > 2000$. The average error and standard deviation in Table 4.5 clearly shows this. When the feature point was in the range of all the cameras ($Y = -1500$ to $Y = 1500$) camera pair 0-2 had the lowest error and standard deviation as shown in Table 4.6.

Table 4.5: Average Error and Standard Deviation (mm) from Room Test

Camera Pair	0-1	0-2	0-3	1-2	1-3	2-3
Average Error	39.95	38.72	62.85	49.32	89.81	86.68
Standard Deviation	18.08	43.88	14.33	48.26	27.33	73.41

Table 4.6: Average Error and Standard Deviation (mm) from Y=-1500 to Y=1500

Camera Pair	0-1	0-2	0-3	1-2	1-3	2-3
Average Error	40.30	25.01	62.77	33.78	92.14	64.19
Standard Deviation	18.08	12.28	14.84	18.16	27.99	23.20

In Fig. 4.4, the magnitude of the 2D error in location is plotted for each camera pair with respect to the Y coordinates. Camera pair 1-3 shows a widely varying error between $Y = -1500$ to $Y = 100$ which comes from the widely varying estimation of

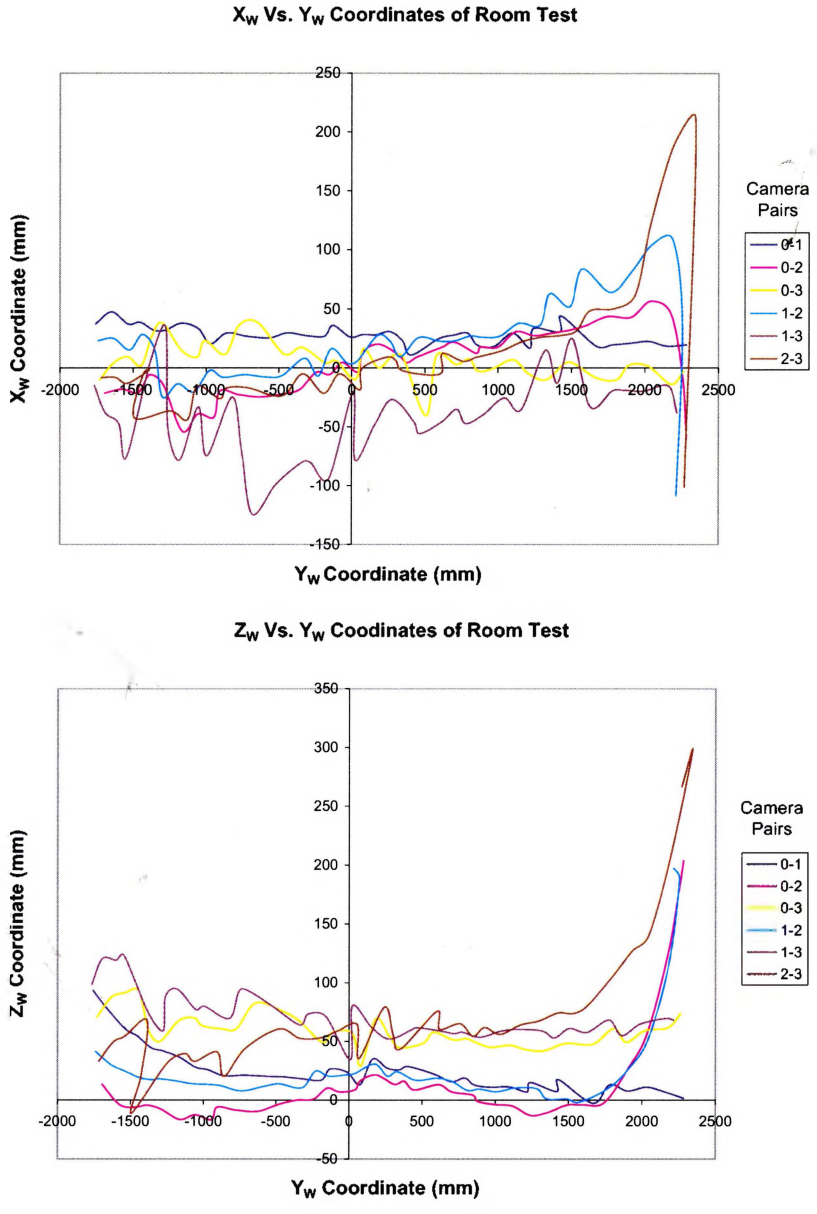

Figure 4.3: Results from Room Test

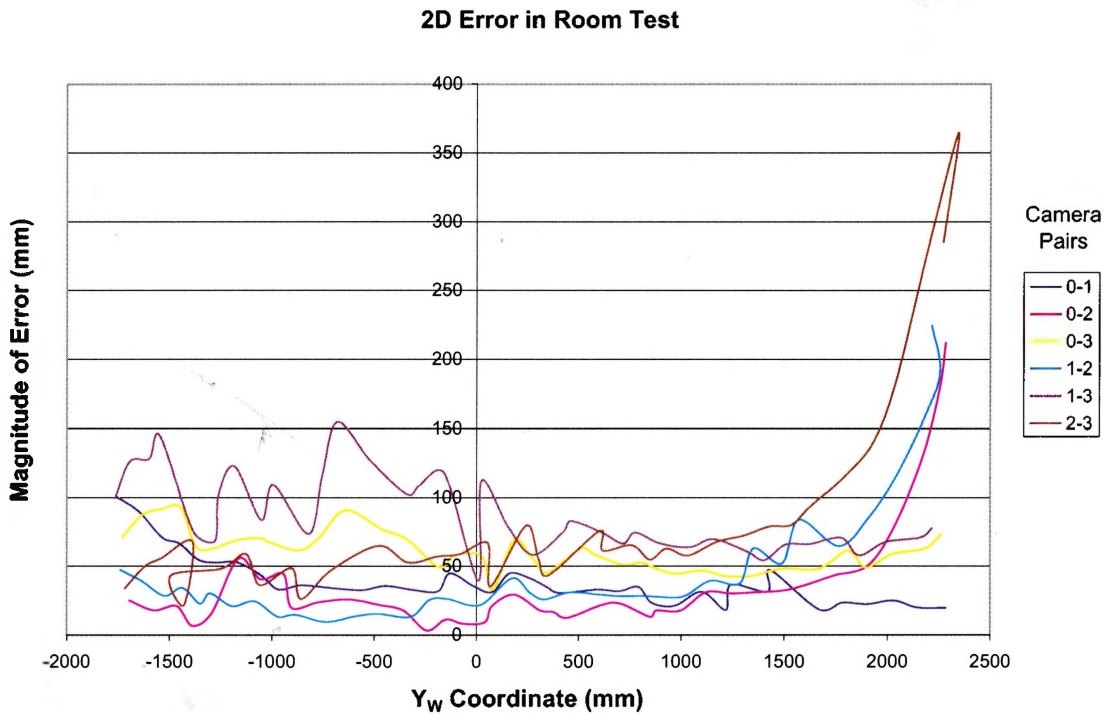

Figure 4.4: 2D Error in Room Test

X_W as seen in Fig. 4.3, X_W *vs.* Y_W. However, this pair shows a low error in the latter half of the plot. This indicates that there are many factors effecting the accuracy of the 3D location of the feature point. These factors are discussed in §4.3. The 2D error plot also suggests that some pairs of cameras produce better results than others. Therefore, there should be ways to analyze the data to determine how the redundant information may be used to determine the best estimate of the 3D location. This idea is discussed further in the last chapter as a future direction of work.

4.3 Performance

4.3.1 Accuracy

The best average error was 25.01mm while the worst average error for $Y = -1500$ to $Y = 1500$ was 92.14mm. The simplest method of using the 6 results from the 4 cameras is to compute the average. Figure 4.5 shows plots for the average of the 6 camera pairs as well as the plots for the 3 best pairs. The 3 best pairs were selected based on the sum of squared differences (SSD) between one camera pair and all the others. A small SSD implies that the other camera pairs produced similar results. A large SSD thus implies that it is an outlier. Hence, the 3 best pairs were the 3 pairs with the smallest SSD. For example, this method would eliminate from the average the X values from camera pair 1-3 from $Y = -750$ to $Y = 0$ as seen in Fig. 4.3. However, as Fig. 4.5 indicates, there is very little improvement over averaging all 6 pairs. Without any a priori knowledge it would be difficult to determine the accuracy of the data and to determine a method for the use of the 6 results. However, often there are assumptions that can be made about the data based on the application and the nature of the objects being tracked. Table 4.7 shows the averages and standard deviation of the graphs in Fig. 4.5.

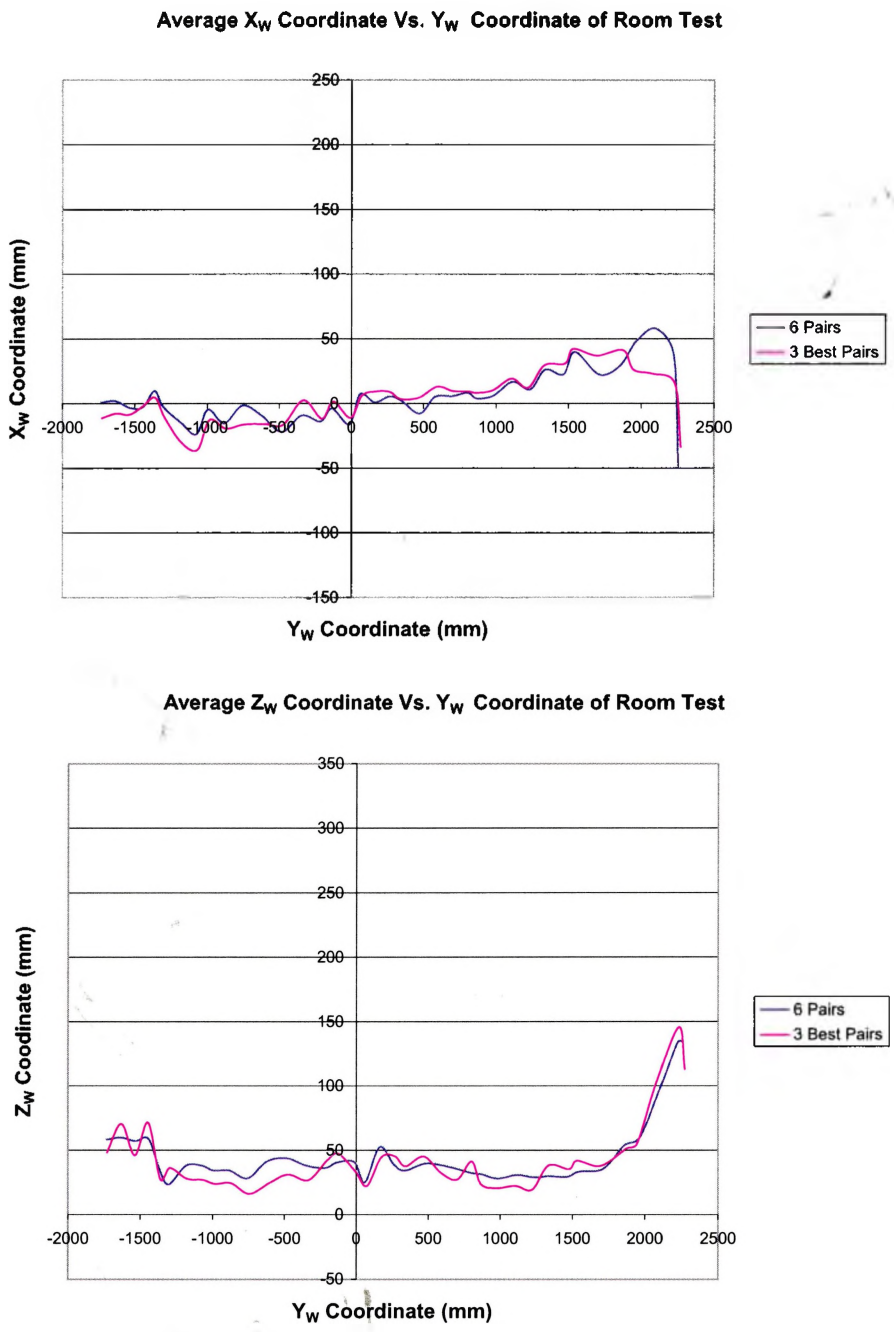

Figure 4.5: Average Error from Room Test

Table 4.7: Average and Standard Deviation (mm) From Averaged Results

	Average X	Average Z	3 Best X	3 Best Z
Average	4.9	47.0	-2.5	53.3
Standard Deviation	20.5	25.0	13.0	16.2

The results demonstrate that each pair of cameras will produce different results depending on factors such as:

- camera pair location,
- baseline between camera pair, and
- angle of triangulation.

Tracking should therefore not be limited to one pair. Camera location, with respect to the target, changes with target movement. The same is true with the angle of triangulation. These criteria may be used to establish the best pair of cameras to use at any given time to track the target. This would also allow the target to move out of the view of one camera and into the view of another.

Servo and Image Resolution

Fixed pan/tilt servo resolution implies a decrease in angular accuracy as distance from the camera increases. Furthermore, fixed image resolution also implies a decrease in angular accuracy proportional to the distance from the camera. These two effects are tied together since the image is used as visual feedback in the control of the pan/tilt servos. Hence accuracy is lost as objects travel further away from the cameras.

The combined effect of image and pan/tilt resolution was tested with the results shown in Fig. 4.6. The experiment geometry was set in such a way to allow a depth measurement to be calculated from the pan servo angle of one camera. The depth became directly proportional to the pan servo angle. The intersection algorithm was

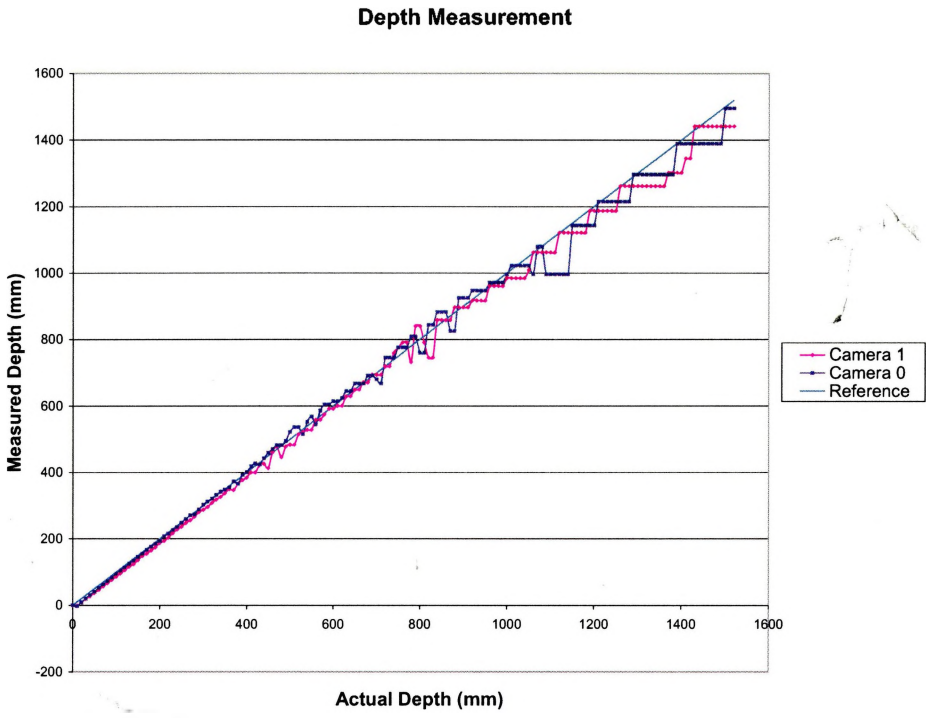

Figure 4.6: Depth Measurement Based on Servo Angle

not required and not used. The depth measurement was therefore dependent on the image and servo resolution. The reference line in Fig. 4.6 represents the correct depth measurement. Two cameras were tested. Both cameras showed that as depth increased the error in the measured depth also increased. Near the end of the plot a stair case effect can be seen. The flat sections represent movements of the feature point that could not be detected in the image.

Camera Placement

Figure 4.7 demonstrates the effect of the angular error discussed in the previous section. In this figure, angle α has an angular error of $\pm\frac{\beta}{2}$. This would cause a depth

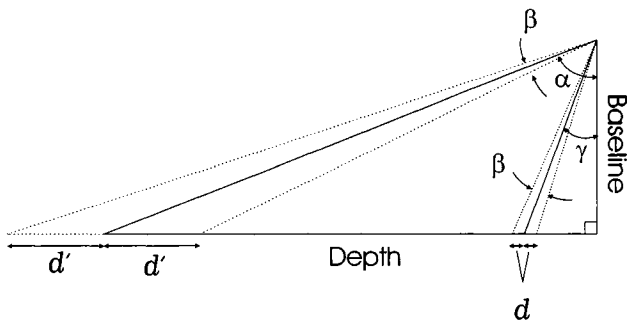

Figure 4.7: Depth Error From Angle Error

error of $\pm d'$. Whereas for a smaller angle γ, the same angular error would only cause a depth error of $\pm d$. Hence it would be advantageous to maintain a small angle. One way of accomplishing this would be to extend the base line. Thus, the location of the cameras in the network plays a significant role in the accuracy of the 3D triangulation. Having long base lines would improve the accuracy of the 3D location measurement by reducing the effects of the image and servo resolution.

4.3.2 Speed

The centroid calculation of §3.2, the pose calculation of §3.2.1 and the intersection calculation of §3.3 are the three calculations required to track and produce the 3D location of the target. Table 4.8 lists the average time it takes for a Intel PII 450MHz (NT4.0) to perform these calculations. The table also lists the frame rate achieved with the configuration shown in Fig. 2.4. A Matrox Meteor-II Multi-channel frame grabber was used to acquire the images from the 4 cameras. The frame grabber image acquisition time was found to be much larger than the time to perform the centroid and intersection calculations and hence is largely responsible for the frame rate listed in Table 4.8. The expected frame rate is based only on the time for the calculations and assumes that image acquisition is instantaneous. Thus, as frame

acquisition time approaches zero, the actual frame rate will approach the expected frame rate. The Meteor-II frame acquisition would have been significantly faster had the camera image streams been synchronized. In the testing, the camera video signals were not synchronized forcing the frame grabber to synchronize with each signal as it switched between cameras.

Table 4.8: Performance

Frame Capture:	66ms
Centroid Calculation:	4.82ms
Intersection Calculation:	5.75μs
Servo Update:	26ms
Frame Rate:	10f/sec
Expected Frame Rate:	33f/sec

4.3.3 Occlusion

Although Fig. 4.3 shows the dramatic effect when one camera can no longer track the target, it is not necessarily a simple task to determine when this occurs solely from the data. As Fig. 4.5 indicates, the average of the 3 pairs with the lowest SSD will not eliminate undesired results. The SSD method did eliminate the effects in the X coordinate but did not eliminate the effects in the Z coordinate when camera 2 could not track the target. However, in a NOC, the tracking system could report a loss of target when the target goes out of view. This would be more logical since the tracking system inherently has the information required to track the target and thus should be able to identify instances when it fails.

A common loss of tracking occurs when the target is occluded. Occlusion occurs when the view of the target is blocked by an object closer to the camera. A NOC can continue to track the target by using views from other cameras. Since each

camera provides redundant information, a process could be developed to select the appropriate view to provide the correct information or the best information. Concave objects may have areas that are occluded from most viewing angles. In these circumstances, not much can be done. However, in many other cases, arbitrary views can be generated using existing multiple views. With the decreasing cost of cameras and networking, implementing multiple cameras can be an effective way to solve occlusion problems.

4.4 Summary

The results demonstrate the need to develop methods to effectively use the redundant information provided by the 6 pairs of cameras. Averaging the 6 results is a simple method to accomplish this. Calculating the SSD and averaging the lowest 3 does not improve the results significantly. A priori knowledge of the target being tracked could be used to improve the tracking algorithm and to improve methods to analyze the redundant information. A NOC would allow the addition of more cameras to solve occlusion problems. The next chapter outlines directions for future work.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

The experiments demonstrate the feasibility of this system to track moving objects in 3D. The pan/tilt camera units could be placed arbitrarily and calibrated. It was shown that the 3D location of the moving object could be found through triangulation using the servo positions of the pan/tilt camera units.

There is an obvious need to increase the accuracy of the calibration primarily in the servo positioning. However, the experiments clearly showed the advantage of using a NOC. It was shown that at any given time one pair of cameras will have a better view point and produce better results. This would include the event where a camera's view is occluded.

Active vision with a NOC can reduce the computational requirements of the application by distributing the load amongst a number of smart cameras. The pan/tilt capability effectively increases the image size without having to increase the field of view of the lens. It also allows the camera to be directed to features of special interest. It was shown that a NOC could be adapted to various volumes.

5.2 Future Directions

This study has shown many of the practical difficulties commonly encountered in vision applications. Future work may be directed towards calibration, servo control and development of strategies to fully take advantage of a NOC.

Calibration

A NOC could span large areas where one smart camera may not be viewing the same area of another. For tracking using the system described in this study, a minimum of two cameras must have views of the same area. The calibration method for the extrinsic parameters, outlined in Chapter 2, requires all the cameras to be able to view the calibration points. Hence there is a need for a calibration method that does not require the cameras to have a common viewing area.

Another avenue of future work in calibration would be a self calibrating system. For instance, if a calibration feature such as 4 light emitting diodes were put on each camera, a smart camera may be able to calibrate extrinsic parameters by finding other cameras within view.

Servo Control

The accuracy of this system is dependent on the positioning of the servos. Futher work could be directed towards enhancing the control of the servos. Path planning and prediction could be used to smooth the movements of the camera.

Strategies for a Network of Cameras

To implement a NOC, three requirements are necesary:

- Smart Camera (filtering, zoom, auto focus...),
- Transmission Medium (copper wire, wireless, fiber optics...), and

- Communication Protocol (IEEE 1394, ethernet...).

A NOC opens many possibilities for efficient use of computational resources and the multiple views from the arbitrarily placed cameras. Some suggestions for future work with a NOC include developing:

- a smart camera unit capable of communicating via ethernet,
- a smart camera unit with zoom, iris and focus control,
- a smart camera unit capable of doing online computations such as filtering or image compression,
- a protocol for communicating with smart cameras and the inter communication of smart cameras for cooperative processing,
- a system to pass the tracking of a target from one smart camera to another, and
- strategies for using the redundant information from more than two cameras viewing the same target.

Appendix A

Spatial Descriptions and Transformations

It is common practice in robotics and computer vision to use coordinate systems to describe the spatial relationship between objects. Although in robotics there are other types of coordinate systems, Cartesian coordinates are most widely used in vision systems. Typically, the notion of a world coordinate system (WCS) is adopted whereby any object's position and orientation can be referenced. However, a point cannot describe the position of an object which spans a finite volume. Therefore, a second coordinate system or object frame is generally attached to the object. The object, with respect to the object frame, does not move allowing the position and orientation of the object to be referenced with respect to the WCS via the object frame.

A.1 Position

The 3D location ${}^{W}P$ of a point in the WCS can be described through a 3 X 1 *position vector*:

$$ {}^{W}P = \begin{bmatrix} {}^{W}P_x \\ {}^{W}P_y \\ {}^{W}P_z \end{bmatrix} \tag{A.1} $$

The superscript "W" denotes the frame (WCS) in which the coordinates are represented.

A.2 Orientation

One way to denote the orientation of an object is to describe the object frame's unit vectors of its principal axes in terms of the WCS. The unit direction vectors of the object frame are denoted by $\vec{X}_O$, $\vec{Y}_O$, $\vec{Z}_O$. When written in terms of the WCS, they are denoted by ${}^{W}\vec{X}_O$, ${}^{W}\vec{Y}_O$, ${}^{W}\vec{Z}_O$. For convenience, the three vectors are put together to form a *rotation matrix*:

$$ {}^{W}_{O}R = \begin{bmatrix} {}^{W}\vec{X}_O & {}^{W}\vec{Y}_O & {}^{W}\vec{Z}_O \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{A.2} $$

where r_{ij}, $i, j \in \{1, 2, 3\}$ are the elements of ${}^{W}\vec{X}_O$, ${}^{W}\vec{Y}_O$, ${}^{W}\vec{Z}_O$. The scalars r_{ij} are the components of the projections of the object frame direction vectors onto the WCS. Hence:

$$ {}^{W}_{O}R = \begin{bmatrix} {}^{W}\vec{X}_O \cdot {}^{W}\vec{X}_W & {}^{W}\vec{Y}_O \cdot {}^{W}\vec{X}_W & {}^{W}\vec{Z}_O \cdot {}^{W}\vec{X}_W \\ {}^{W}\vec{X}_O \cdot {}^{W}\vec{Y}_W & {}^{W}\vec{Y}_O \cdot {}^{W}\vec{Y}_W & {}^{W}\vec{Z}_O \cdot {}^{W}\vec{Y}_W \\ {}^{W}\vec{X}_O \cdot {}^{W}\vec{Z}_W & {}^{W}\vec{Y}_O \cdot {}^{W}\vec{Z}_W & {}^{W}\vec{Z}_O \cdot {}^{W}\vec{Z}_W \end{bmatrix}. \tag{A.3} $$

Since all the vectors in this matrix are described with respect to the WCS, the superscript "W" can be omitted on the elements of the matrix allowing the superscript "W" on the ${}^{W}_{O}R$ to imply that the vectors are in the WCS unless otherwise shown. Therefore, ${}^{W}_{O}R$ can be written as:

$$ {}^{W}_{O}R = \begin{bmatrix} \vec{X}_O \cdot \vec{X}_W & \vec{Y}_O \cdot \vec{X}_W & \vec{Z}_O \cdot \vec{X}_W \\ \vec{X}_O \cdot \vec{Y}_W & \vec{Y}_O \cdot \vec{Y}_W & \vec{Z}_O \cdot \vec{Y}_W \\ \vec{X}_O \cdot \vec{Z}_W & \vec{Y}_O \cdot \vec{Z}_W & \vec{Z}_O \cdot \vec{Z}_W \end{bmatrix} . \tag{A.4} $$

As the dot product of two unit vectors yields the cosine of the angle between them, the components of the rotation matices are often referred to as *direction cosines.* Further inspection of A.4 shows that the rows of the matrix are the unit vectors of the WCS described in terms of the object frame. This implies that:

$$ {}^{O}_{W}R = {}^{W}_{O}R^{T}. \tag{A.5} $$

This also suggest that the inverse of a rotation matrix is equal to its transpose, which can be easily verified as:

$$ {}^{W}_{O}R^{T}\ {}^{W}_{O}R = \begin{bmatrix} {}^{W}\vec{X}^{T}_{O} \\ {}^{W}\vec{Y}^{T}_{O} \\ {}^{W}\vec{Z}^{T}_{O} \end{bmatrix} \begin{bmatrix} {}^{W}\vec{X}_O & {}^{W}\vec{Y}_O & {}^{W}\vec{Z}_O \end{bmatrix} = I \ , \tag{A.6} $$

where I is a 3×3 identity matrix. Hence:

$$ {}^{W}_{O}R = {}^{O}_{W}R^{-1} = {}^{O}_{W}R^{T} \ . \tag{A.7} $$

Finally, with the object coordinate system rigidly attached to the object, the object's position and orientation can be described through the object frame in the WCS. The position ${}^{W}P_{O_{org}}$ of the object frame origin and the orientation with respect to the WCS, ${}^{W}_{O}R$, fully describes the object frame in the WCS.

A.3 X-Y-Z Fixed Angles

As mentioned in the previous section, the elements of a rotation matrix are the direction cosines. This leads to the *X-Y-Z Fixed Angle* interpretation of a rotation matrix. For a rotation γ about the X-axis, β about the Y-axis and α about the Z-axis of a fixed coordinate system, the rotation matrix is:

$$R_{XYZ}(\gamma, \beta, \alpha) = R_Z(\alpha) R_Y(\beta) R_X(\gamma)$$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

$$= \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}. \tag{A.8}$$

The word "Fixed" refers to the fact that the rotations are specified about the fixed (i.e., non-moving) reference frame. This convention is referred to as *roll, pitch, yaw* angles.

A.4 Z-Y-X Euler Angles

For a rotation α about the Z-axis, *then* β about the Y-axis and *then* γ about the X-axis of a *moving frame*, the rotation matrix is:

$$R_{ZYX}(\alpha,\beta,\gamma) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)$$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

$$= \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \tag{A.9}$$

In the moving frame reference, the rotation about the Y-axis is applied to the resulting new orientation from the rotation about the X-axis and the rotation about the Z-axis is applied after the Y-axis rotation. Hence each rotation is applied to the frame in a different orientation sequentially. This also implies there are many different Euler angle representations. Note in this case, Eq. A.8 and Eq. A.9 are the same. X-Y-Z fixed angle rotation is the same as Z-Y-X Euler angle rotation. A X-Y-Z Euler rotation would be completely different.

A.5 Mappings From One Frame To Another

For a given point in the object frame, the point can be described in the WCS knowing the description of the object frame in the WCS by:

$${}^{W}P = {}^{W}_{O}R\,{}^{O}P + {}^{W}P_{O_{org}}. \tag{A.10}$$

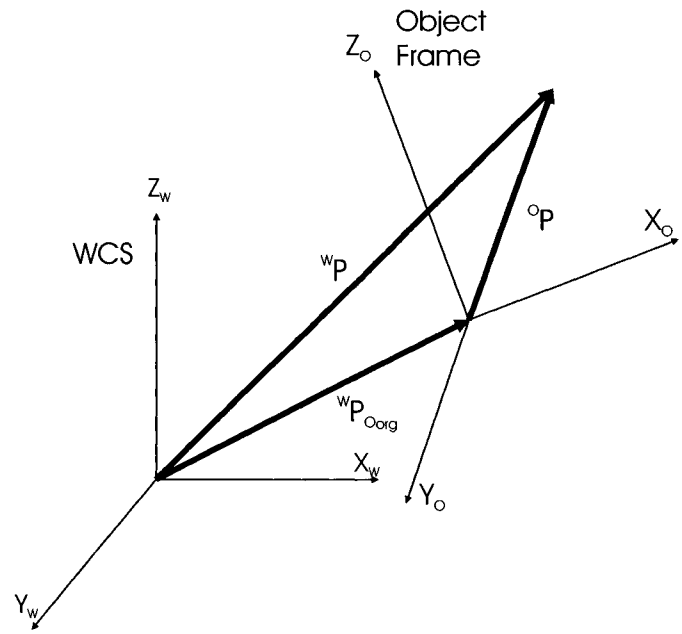

Figure A.1: Mapping a Vector from One Frame to Another

Equation A.10 describes a general transformation mapping of a vector from its description in one frame to a description in a second frame. The mapping is graphically shown in Fig. A.1.

A.6 The Homogeneous Transform

Equation A.10 can be simplified if the rotational description ${}^W_O R$ and the position of the origin ${}^W P_{O_{org}}$ are combined to form a *homogeneous transform*:

$$ {}^W_O T = \begin{bmatrix} & & & \vdots & \\ & {}^W_O R & & \vdots & {}^W P_{O_{org}} \\ & & & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} . \tag{A.11} $$

Then Eq. A.10 becomes:

$$\begin{bmatrix} \\ {}^{W}P \\ \\ \cdots \\ 1 \end{bmatrix} = \begin{bmatrix} & & & \vdots & \\ & {}^{W}_{O}R & & \vdots & {}^{W}P_{O_{org}} \\ & & & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \begin{bmatrix} \\ {}^{O}P \\ \\ \cdots \\ 1 \end{bmatrix}, \tag{A.12}$$

or

$${}^{W}P = {}^{W}_{O}T \; {}^{O}P \; . \tag{A.13}$$

Note that while the homogeneous transform is derived in terms of mappings, it also serves as a description of frames. Just as rotation matrices are used to specify an orientation, the homogeneous transform is used to specify a frame. The description of frame B relative to A is ${}^{A}_{B}T$.

A.7 Compound Transforms

If frame C is known relative to frame B, and frame B is known relative to frame A, then:

$${}^{A}_{C}T = \begin{bmatrix} & & & \vdots & \\ & {}^{A}_{B}R \; {}^{B}_{C}R & & \vdots & {}^{A}_{B}R \; {}^{B}P_{C_{org}} + {}^{A}P_{B_{org}} \\ & & & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix}, \tag{A.14}$$

which is the same as:

$${}^{A}_{C}T = {}^{A}_{B}T \; {}^{B}_{C}T. \tag{A.15}$$

A.8 Inverting a Transform

If the description of frame B with respect to frame A is ${}^{A}_{B}T$, then the inverse of ${}^{A}_{B}T$ is the description of frame A with respect to frame B, i.e.:

$$ {}^{B}_{A}T = {}^{A}_{B}T^{-1}. \tag{A.16} $$

The homogeneous transform matrix has unique properties that simplify the inversion to just a few calculations. Recall Eq. A.5 reproduced here:

$$ {}^{O}_{W}R = {}^{W}_{O}R^{T}. \tag{A.17} $$

To find ${}^{B}_{A}T$ and ${}^{B}_{A}R$, ${}^{B}P_{A_{org}}$ must be computed from ${}^{A}_{B}R$ and ${}^{A}P_{B_{org}}$. The description of ${}^{A}P_{B_{org}}$ into frame B is:

$$ {}^{B}({}^{A}P_{B_{org}}) = {}^{B}_{A}R\ {}^{A}P_{B_{org}} + {}^{B}P_{A_{org}}\ . \tag{A.18} $$

Since the left-hand side of Eq. A.18 must be zero, this simplifies to:

$$ {}^{B}P_{A_{org}} = -{}^{B}_{A}R\ {}^{A}P_{B_{org}} = -{}^{A}_{B}R^{T}\ {}^{A}P_{B_{org}}. \tag{A.19} $$

Using Eq. A.17 and Eq. A.18,

$$ {}^{B}_{A}T = \begin{bmatrix} & & & \vdots & \\ & {}^{A}_{B}R^{T} & & \vdots & -{}^{A}_{B}R^{T}\ {}^{A}P_{B_{org}} \\ & & & \vdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} = {}^{A}_{B}T^{-1}\ . \tag{A.20} $$

Appendix B

Pinhole Camera Model

The pinhole camera model is widely used to represent the projection of light rays onto the imaging surface of a camera. The model assumes that the perspective rays pass through an infinitesimal aperture at the front of the camera. In reality, this is not the case since the aperture must be a diameter to allow sufficient light to pass. A lens gathers more light, allowing the camera to work with less ambient illumination or with a faster shutter speed. However, the use of a lens limits the depth of field. Lenses do not introduce any effects that violate the assumption of ideal perspective projection defined by the pinhole camera model.

Figure B.1 shows the pinhole camera model. According to the *law of collinearity*, or the *fundamental perspective projection*, a point on the image, the pinhole and the scene point corresponding to the image point all lie along the same line. Given the location of a scene point $P = (P_x, P_y, P_z)$, the resulting point on the image plane is (u, v, f) where:

$$\begin{aligned} u &= \frac{f}{P_z} P_x \\ v &= \frac{f}{P_z} P_y \end{aligned} \tag{B.1}$$

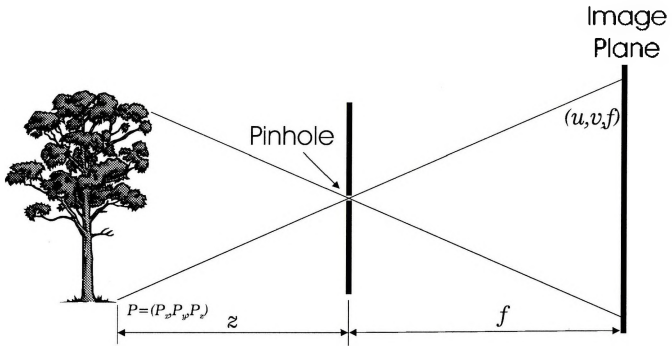

Figure B.1: The Pinhole Camera Model

Equation B.1 is referred to as the *perspective projection* equation. In the photogrammetric literature, f is referred to as the *camera constant*. It is of course related to the focal length of the lens. However, as the lens must be moved closer to or farther from the image projection plane to focus, the focal length of the lens must be considered as only the nominal value of f. To complicate matters, most vision systems use a digitizing system to allow a computer to interpret the image and perform calculations (computer vision). The image sampling process and conversion to a digital format usually scales the image which further alters the effective value of f. The actual value of f must be determined by a calibration process that involves all parts of the imaging system from the lens to the digitizer.

In most cases the scene point coordinate will be known or at least defined in the WCS while the image will be defined in the camera coordinate system. Equation A.10 can be used to map points from one coordinate system to another.

References

[1] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane Image Analysis: An Approach to Determining Structure from Motion. *International Journal on Computer Vision*, 1(1), 1987.

[2] J. Brooker, A Plooy, P. M. Sharkey, and J. P. Wann. A Dynamic Vergence Telepresence System Incorporating Active Gaze Control. *11th IS&T/SPIE Ann. Symp. on Electronics Imaging '99, Photonics West '99, San Jose, CA*, pages 23–29, January 1999.

[3] D. C. Brown. Close-range Camera Calibration. *Photogrammetric Engineering*, 37:855–866, 1971.

[4] Homer H. Chen. Pose Determination from Line-to-plane Correspondences: Existence Condition and Closed-form Solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), June 1991.

[5] Zen Chen, Din-Chang Tseng, and Jenn-Yee Lin. A Simple Vision Algorithm for 3-D Position Determination Using a Single Calibration Object. *Pattern Recognition*, 22(2):173–187, 1989.

[6] H. L. Chou and W. H. Tsai. A New Approach to Robot Location by House Corners. *Pattern Recognition*, 19:439–451, 1986.

[7] H. Collewijn, C. J. Erkelens, and R. M. Steinman. Binocular Coordination of Horizontal Saccadic Eye Movements. *Journal of Physiology*, pages 157–182, 1988.

[8] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. Kenyon, and J. C. Hart. The CAVE - Audio Visual Experience Automatic Virtual Environment. *Communications of the ACM*, pages 64–72, 1992.

[9] Olivier Faugeras and Luc Robert. What Can Two Images Tell Us About a Third One? *Internatinonal Journal of Computer Vision*, 18:5–19, 1996.

[10] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[11] I. Fukui. TV Image Processing to Determine the Position of Robot Vehicle. *Pattern Recognition*, 14:101–109, 1981.

[12] W. E. L. Grimson. A Theory of Human Stereo Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, January 1985.

[13] R. M. Haralick and Y. H. Chu. Solving Camera Parameters from the Perspective Projection of a Parameterized Curve. *Pattern Recognition*, 17:637–645, 1984.

[14] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 2. Addison-Wesley Publishing Company, Inc., 1986.

[15] E. E. Hemayed, A. Sandbek, A. G. Wassal, and A. A. Farag. Investigation of Stereo-based 3D Surface Reconstruction. *Proceedings of SPIE*, 3023:191–202, February 1997.

[16] Radu Horaud, Bernard Conio, Oliver Leboulleux, and Bernard Lacolle. An Analytic Solution for the Perspective 4-Point Problem. *Computer Vision, Graphics, and Image Processing*, 47:33–44, 1989.

[17] Takeo Kanade, Hideo Saito, and Sundar Vendula. The 3D Room: Digitizing Time-varying 3D Events by Synchronized Multiple Video Streams. Technical Report CMU-RI-TR-98-34, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890 USA, December 1998.

[18] Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano, and Masaya Tanaka. A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications. *IEEE Conference on Computer Vision and Pattern Recognition*, June 1996.

[19] K. Kanatani. Camera Rotation Invariance of Image Characteristics. *Computer Vision, Graphics, and Image Processing*, 39(3):328–354, September 1987.

[20] William Klarquist and Alan Bovik. FOVEA: A Foveated Vergent Active Stereo System for Dynamic Three-Dimensional Scene Recovery. *International Conference on Robotics and Automation*, June 1998.

[21] Eileen Kowler. *Exploratory Vision, The Active Eye*, chapter Cogito Ergo Moveo: Cognitive Control of Eye Movement, pages 51–74. Springer, 1996.

[22] E. P. Krotkov. *Active Computer Vision by Coorperative Focus and Stereo.* Springer-Verlas, New York, 1989.

[23] Peter Lehel, Elsayed E. Hemayed, and Aly A. Farag. Sensor Planning for a Trinocular Active Vision System. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:306–312, June 1999.

[24] Seppo Linnainmaa, David Harwood, and Larry S. Davis. Pose Determination of a Three-Dimensional Object Using Triangle Pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):634–647, 1988.

[25] Yuncai Liu, Thomas S. Huang, and Oliver D. Faugeras. Determination of Camera Location from 2-D to 3-D Line and Point Correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), January 1990.

[26] David Marr. *Vision.* Freeman, 1982.

[27] L. Matthies, R. Szeliski, and T. Kanade. Kalman Filter-based Algorithms for Estimating Depth from Image Sequences. *International Journal on Computer Vision*, 3:209–236, 1989.

[28] D. W. Murray, K. J. Bradshaw, P. F. McLauchlan I. D. Reid, and P. M. Sharkey. Driving Saccade to Pursuit Using Image Motion. *International Journal on Computer Vision*, 16:205–238, 1995.

[29] Don Murray and Anup Basu. Motion Tracking with an Active Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5), May 1994.

[30] Masatoshi Okutomi and Takeo Kanade. A Multiple-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4), April 1993.

[31] Long Quan and Zhongdan Lan. Linear N-Point Camera Pose Determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), August 1999.

[32] Peter Rander. A Multi-Camera Method for 3D Digitization of Dymanic, Real-World Events. Technical Report CMU-RI-TR-98-12, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890 USA, May 1998.

[33] P. M. Sharkey, D. W. Murray, P. F. McLauchlan, and J. P. Brooker. Hardware Development of the Yorick Series of Active Vision Systems. *Invited Paper for Special Issues on Mobile Robotics, Microprocessors and Microsystems Journal*, 21:363–375, 1998.

[34] P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A Modular Head/eye Platform for Real-time Reactive Vision. *Mechatronics Journal*, 3(4):517–535, 1993.

[35] Jay Stavnitzky and David Capson. Multiple-Camera Uncalibrated 3D Visual Servo. *International Symposium on Robotics, Montreal*, 2000.

[36] I. Sutherland. Three-dimensional Data Input by Tablet. *Proceedings of the IEEE*, 66:453–461, April 1974.

[37] Roger Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.

[38] Todd Williamson and Charles Thorpe. A Trinocular Stereo System for Highway Obstacle Detection. *IEEE International Conference on Robotics and Automation*, pages 2267–2273, May 1999.

[39] P. R. Wolf. *Elements of Photogrammetry.* McGraw-Hill Book Company, Inc., New York, 1983.

[40] Y. Yakimovsky and R. Cunningham. A System for Extracting Three-dimensional Measurements from a Stereo Pair of TV Cameras. *Computer Graphics Image Processing*, 7:195–210, 1978.