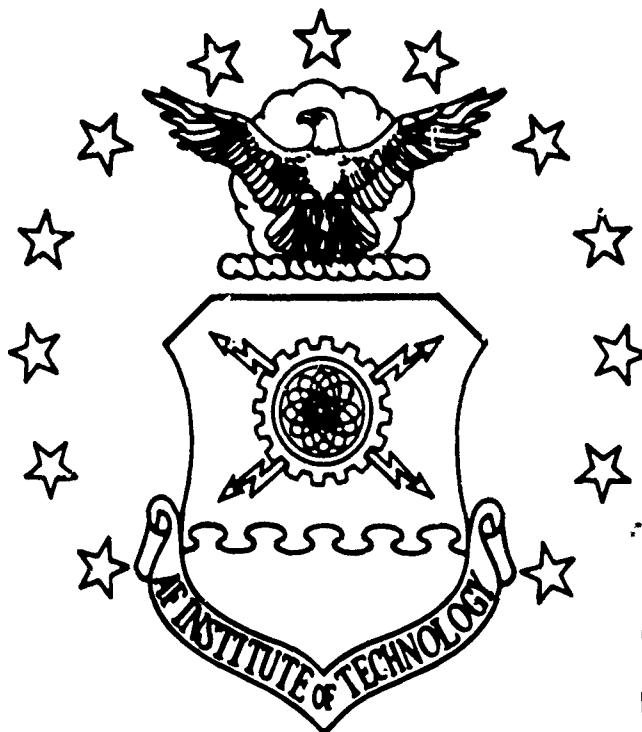


DTIC FILE 0021

①

AD-A216 042



DTIC
ELECTE
DEC 15 1989
S B D

VISUAL SERVOING FOR AUTONOMOUS
AIRCRAFT REFUELING

THESIS

Richard P. Shipman
Capt, USAF

AFIT/GE/ENG/89D-48

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

89 12 14 039

AFIT/GE/ENG/89D-48

**VISUAL SERVOING FOR AUTONOMOUS
AIRCRAFT REFUELING**

THESIS

**Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering**

**Richard P. Shipman, B.S.E.E
Capt, USAF**

December 14, 1989

Approved for public release; distribution unlimited

Preface

As with every endeavour undertaken in life, a person must have a driving force or purpose. The purpose of my research was to develop a robotic visual servoing system (RVSS) that would demonstrate the feasibility of visual servoing for autonomous aircraft refueling. The driving force behind this effort was the love and patience of my wife, Marie. Without her support and understanding, my work would have been fruitless and uninteresting.

Of course any research performed is never conducted on the knowledge and work of one person. I wish to thank Vern, Dave, Paul, Mark, and Sam for their help and for letting me "squeeze" in a picture or two between their runs with the PUMA. Their incredible ability to push the return key on "CYCLOP" was an immense aid in my research. I would also like to thank Mr. Dan Zambon for keeping all my systems running and for putting with my own imperfections in regard to operating the equipment. And of course a special thanks to Dr. Michael Leahy, my thesis advisor and resident PUMA expert. If it were not for his endless patience and help, I would still be at the beginning of my research trying to figure out how to communicate with the PUMA. In addition, without the support of my former supervisor and office mates, I would never have applied or been accepted to AFIT. Thank you Lt. Col. Ben Lucas, Sharon, Hector and everybody else at Space Division for your support. Thanks for all your help guys!

Finally, I would like to thank my wife for marrying me and putting up with me for the last 18 months; both were great challenges. And last, but not least, I would like to ask my son Derek's forgiveness for disappearing on him for the past year-and-a-half of his life; for all the times I said no whenever you asked for my help or to play with you; for all the weekends you had to stay home because I had to work on homework; and for all the times that I snapped at you because I was frustrated and angry with school. You are and always will be important to me. This thesis is dedicated to my son Derek, whom I hope will one day do it bigger and better than I did.

Richard P. Shipman



n For	
/&I	☒
ed	☐
tion	☐
tion/	
lity Codes	
l and/or	
Special	
Dist	
A-1	

Table of Contents

	Page
Preface	ii
Table of Contents	iii
List of Figures	vi
List of Tables	viii
Abstract	ix
 I. Introduction	 1-1
1.1 Motivation	1-1
1.2 Objective	1-3
1.3 Problem Statement	1-3
1.4 Contribution	1-5
1.5 Organization	1-6
 II. Review of Literature	 2-1
2.1 Introduction	2-1
2.2 Robotic Visual Servoing Research	2-1
2.3 Image Brightness Preprocessing	2-4
2.3.1 Thresholding	2-4
2.3.2 Brightness Normalization	2-6
2.4 Object Recognition	2-7
2.4.1 Decision-Theoretic	2-7
2.4.2 Structural	2-8
2.5 Visual Ranging Methods	2-10
2.6 Summary	2-11

	Page
III. Refueling Receptacle Recognition	3-1
3.1 Introduction	3-1
3.2 ITEX 100 Image Processor	3-2
3.3 Structured Environment Research	3-2
3.4 Unstructured Environment Research	3-6
3.5 Range	3-12
3.6 Modified Brightness Normalization	3-17
3.7 Summary	3-18
IV. Visual Servoing	4-1
4.1 Introduction	4-1
4.2 PUMA Communications	4-1
4.3 Search Pattern	4-2
4.4 Kinematics	4-5
4.4.1 Forward Kinematics	4-5
4.4.2 Inverse Kinematics	4-13
4.4.3 Self-Centering	4-17
4.5 Dynamic Servoing	4-17
4.6 Summary	4-18
V. Results and Discussion	5-1
5.1 Introduction	5-1
5.2 Refueling Receptacle Recognition	5-1
5.2.1 Object Recognition	5-2
5.2.2 Modified Brightness Normalization Preprocessing	5-5
5.3 Visual Servoing	5-6
5.3.1 Search	5-6
5.3.2 Static Look-and-Move	5-7

	Page
5.3.3 Dynamic Look-and-Move	5-8
5.4 Summary	5-8
VI. Conclusions and Recommendations	6-1
6.1 Summary	6-1
6.2 Conclusions	6-1
6.2.1 Object Recognition	6-1
6.2.2 Visual Servoing	6-2
6.3 Recommendations	6-2
6.3.1 Brightness Normalization Preprocessing	6-2
6.3.2 Camera Optimization	6-3
6.3.3 Partial Indicator Pattern Recognition	6-3
6.3.4 Vaccelerator Board Usage	6-3
6.3.5 Dynamic Look-and-Move	6-3
A. Equipment List	A-1
B. Robotic Visual Servoing System (RVSS) Software Flow	B-1
B.1 Introduction	B-1
B.2 RVSS Software Flow Discussion	B-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure	Page
1.1. Current Aerial Aircraft Refueling System	1-2
1.2. Proposed Ground-Based Robotic Aircraft Refueling System	1-3
1.3. Standard Configuration of Overall System During Testing	1-4
2.1. FIGARO at Work	2-2
2.2. Recognition of Gasket Based on Geometry	2-5
2.3. Structural Method Of Recognition	2-9
3.1. Video Coordinate System	3-3
3.2. Refueling Receptacle Test Stand with Mask	3-4
3.3. White Object Search	3-5
3.4. Relationship of Object Parameters	3-7
3.5. Placement of the Special Indicators	3-8
3.6. Indicator Identification	3-11
3.7. Relationship of Indicator Parameters Upon Identification	3-12
3.8. Indicator Width vs. Range	3-14
3.9. Logarithmic Plots	3-16
3.10. Example Isolation Mask (5 x 5)	3-19
4.1. Initial Search Position	4-3
4.2. Search Pattern	4-3
4.3. Search Logic Diagram	4-4
4.4. Joint Four Positioning	4-6
4.5. Representative PUMA Configuration	4-7
4.6. Denavit-Hartenburg Representation of the PUMA	4-8
4.7. Geometric View of Psuedolink Transformation	4-11

Figure	Page
4.8. Side View of Representative PUMA Configuration	4-14
4.9. Top View of the Representative PUMA Configuration	4-15
4.10. Side View of a Two-Link Manipulator	4-16
5.1. Identified Refueling Receptacle	5-2
5.2. Image Missing Indicator(s)	5-3
5.3. Top Indicator Partially Hidden	5-4
5.4. Bottom Indicators Partially Hidden	5-5
5.5. Partial Detection of Receptacle Indicators	5-6
5.6. Refueling Nozzle Placement Near Refueling Receptacle	5-7
A.1. PUMA-560	A-1
A.2. VAXstation III "CYCLOP"	A-2
A.3. Sony Compact CCD Video Camera	A-3
A.4. Refueling Receptacle Test Stand	A-4
A.5. Simulated Refueling Nozzle	A-5
B.1. Robotic Visual Servoing System (RVSS) Software Logic Diagram	B-2
B.2. RVSS Welcome Message	B-3
B.3. ITEX Welcome Message	B-4
B.4. PUMACAL Welcome Message	B-5

List of Tables

Table	Page
3.1. Indicator Width vs. Range	3-15

Abstract

The design and development of autonomous robotic systems for military applications is of major interest to the Air Force. The robotics research group at the Air Force Institute of Technology (AFIT) is actively involved in autonomous aircraft refueling research. The approach being undertaken at AFIT is the integration of vision and robot technologies allowing visual information, regarding identification and location of an aerial refueling receptacle on an aircraft, to be fed back to a robot controller which develops trajectory information needed for compliant robot motion.

The purpose of this thesis effort was to modify, develop, and evaluate existing visual processing algorithms to allow the AFIT PUMA-560 to visually acquire and track a half-scale mock-up UARRSI aerial refueling receptacle, and place a simulated refueling nozzle in the close proximity of the receptacle's slipway. This was accomplished by developing visual object recognition and robot servoing algorithms which: analyzed images, recognized and determined the position of the receptacle, and calculated proper PUMA joint angles for end-effector placement.

Based on this approach, a static look-and-move robotic visual servoing system (RVSS) was demonstrated which: identified, located, and servoed the PUMA's end-effector to the close proximity of the refueling receptacle. The RVSS provides the visual recognition and servoing necessary so that when combined with a compliant motion controller completes the insertion task, thereby completely demonstrating the AFIT concept for robot refueling. In addition, the initial architecture for a dynamic look-and-move RVSS was developed based on parallel processing. This system provides AFIT with the capability to meet Air Force initiatives for research on the further design, development, and application of visual servoing for autonomous aircraft refueling tasks.

VISUAL SERVOING FOR AUTONOMOUS AIRCRAFT REFUELING

I. Introduction

1.1 Motivation

Within the Air Force, the design and implementation of autonomous robotic systems capable of performing complex tasks will be a major initiative in the coming decade. One complex task that has been focused on over the past several years is autonomous aircraft refueling for both ground-based and aerial applications. With present technology, both tasks require intense human interaction and, in some environments, current techniques prove to be quite inadequate. Current aerial aircraft refueling systems are quite complex and require skilled personnel to manipulate the refueling boom and insert the fuel nozzle into the refueling receptacles of other aircraft, as shown in Figure 1.1. Ground-based refueling, on the other hand, requires several military members to accomplish the task, and in a hostile environment, refueling can be cumbersome and difficult [1:4].

Initial research in the area of autonomous ground-based refueling was conducted by the Flight Dynamics Laboratory at Wright-Patterson AFB. Their research determined that the air refueling receptacle would be the optimal place to refuel aircraft when utilizing a robotic refueling system. This decision was based on accessibility of the receptacle and minimal procedural requirements, unlike current refueling practice (i.e., removing fuel caps, connecting fuel hoses), for refueling at this location [1:5]. Figure 1.2 provides an illustration of this proposed robotic ground-based refueling scheme. For the case of the proposed ground-based refueling system, a fuel truck is driven into the close vicinity of the aircraft and a robotic refueling boom is placed over the aircraft. The operation of the robotic refueling manipulator in the vicinity of the target aircraft could be automated or left under manual control to locate the refueling receptacle, position, and then insert the refueling nozzle into the receptacle.

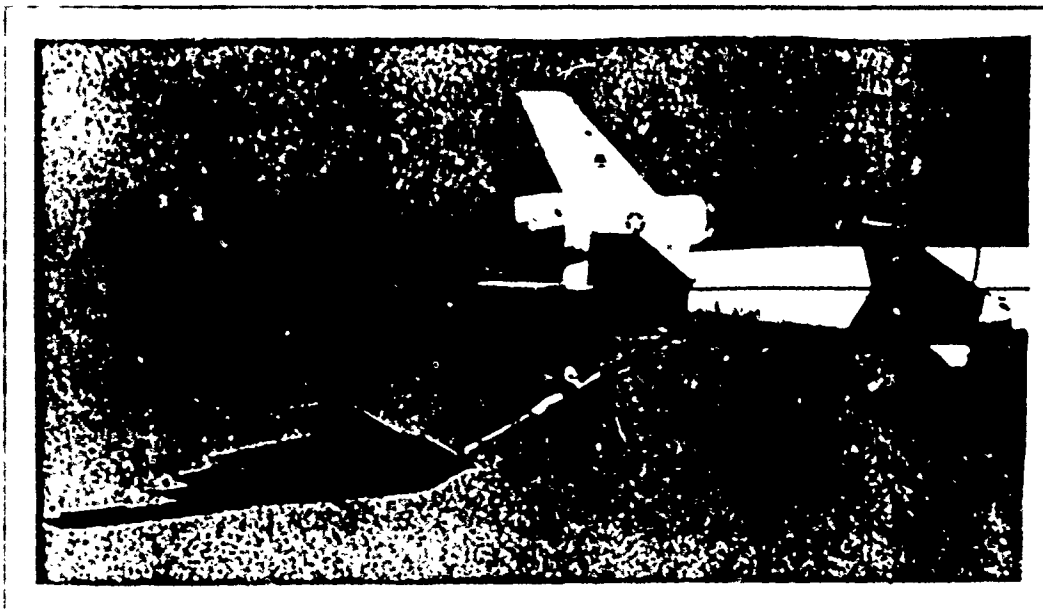


Figure 1.1. Current Aerial Aircraft Refueling System [2:30]

The robotics research group at the Air Force Institute of Technology (AFIT) is actively involved in aircraft refueling research. The approach being undertaken at AFIT is the integration of both vision and robot technologies in such a way that visual information regarding identification and location of the refueling receptacle is fed back to the robot controller which develops trajectory information needed for robot motion. In effect, the refueling system will use the vision system to close the loop around the motion control problem forming a visual servoing system. The job of the visual servoing system will be to place the manipulator in contact, or close proximity, with the refueling receptacle slipway from where a compliant motion algorithm would accomplish the final insertion [3]. The actual insertion will be performed using an impedance control scheme [3,4,5] which provides the necessary compliance for smooth motion along the refueling receptacle slipway and finally, insertion. Successful validation of this concept required advances in visual receptacle recognition, visual servoing, and compliant boom control.

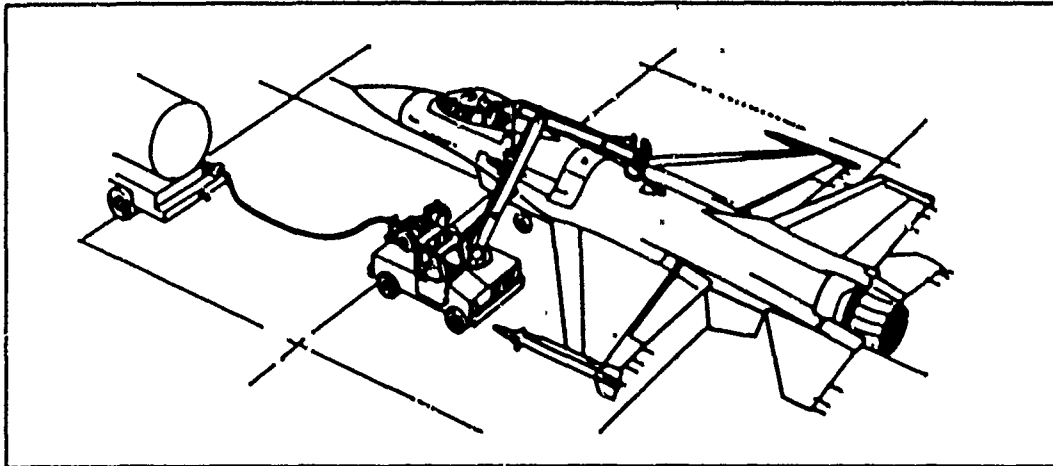


Figure 1.2. Proposed Ground-Based Robotic Aircraft Refueling System [1:3]

1.2 Objective

The objective of this thesis effort is to demonstrate the basic concepts of visual recognition and servoing for both ground-based and aerial refueling applications. Demonstrations are accomplished by using a PUMA-560 manipulator arm equipped with a CCD camera mounted on the third link, an ITEX 100 image processing board for preprocessing of vision data, and a half-scale mock-up of a standard Universal Aerial Refueling Receptacle Slipway Installation (UARRSI) (simulating the receiving aircraft). Figure 1.3 shows the normal configuration of the system in testing. The ultimate goal will be for the refueler (PUMA-560) to visually scan for the refueling receptacle, and upon identification, provide the correct joint positions (based on receptacle cartesian coordinates) necessary to place the end-effector (refueling nozzle) on the slipway. Visual servoing will be developed for the static (ground-based refueling) environment and, the initial architecture for the dynamic (aerial-based refueling) system will be developed.

1.3 Problem Statement

This thesis effort is a continuation of previous thesis research conducted at AFIT over the past several years. Capt. Mikel Miller, USAF, conducted experiments at AFIT in late 1987, using the PUMA-560 and a video camera (different than the one used for this

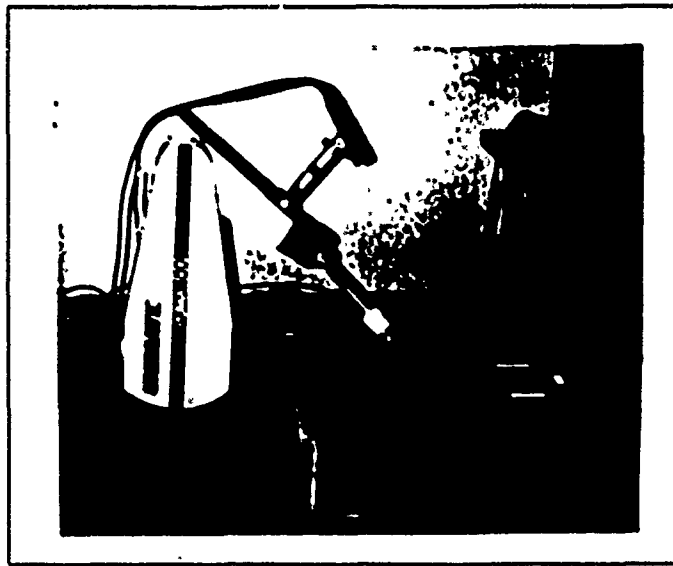


Figure 1.3. Standard Configuration of Overall System During Testing

effort), which successfully proved that the concept of visually guided robotic ground-based refueling was feasible. By representing the aircraft refueling receptacle as a white ball on a black background, the robot was able to locate, and move towards the ball [1]. This research demonstrated that robot vision could be used to locate and visually guide a robot end-effector to a specified object for both static and dynamic look-and-move systems. The system Miller developed was crude and only capable of operating in a completely structured environment (black background and white object) with static objects. The major drawbacks to Miller's research was the inability of the video system to operate in an unstructured (real-world) environment, track a moving object, and efficiently process vision data.

In 1988, Capt. Andrew Salisbury, USAF, conducted research also operating in a structured environment with a white object as the target. His goal was to use Kalman filtering to predict the trajectory of a moving object, thereby allowing the PUMA-560 to track and contact the object [3]. Unfortunately, his effort was never completed, but some working software was developed dealing with PUMA communications as well as camera start-up. These algorithms served as the basis for the software developed for this thesis effort.

During the same period of time that Miller was conducting his research, Capt. Lawrence Lambert was developing an Autonomous Face Recognition System. His efforts dealt mainly with visual recognition of faces present in an image. A key development produced by his research (besides a working face recognition system) was a brightness normalization algorithm which produced brightness invariant images [6]. In regard to this robotics research, Lambert's work in the area of image preprocessing may provide a possible "bridge" between robotic vision systems operating in structured lighting and unstructured lighting environments.

The purpose of this thesis research was to modify, develop, and evaluate existing visual processing algorithms to allow the AFIT robotic visual servoing system (RVSS) to acquire and track a half-scale aerial refueling receptacle, and to place the refueling nozzle in the close proximity of the receptacle's slipway. In order to fully succeed in this undertaking, the following areas were specifically addressed:

- the communications between the PUMA-560 and a VAXstation III to enable proper motion control,
- an "optimal" search pattern to ensure efficient area coverage and reduce refueling receptacle recognition error,
- the minimal subset of visual processing cycles necessary to accomplish the autonomous refueling task,
- the path planning techniques to guide the robot between visual updates,
- the effects of, and requirements for, brightness and critical threshold levels for adequate operation of the vision system,
- an object recognition algorithm providing minimal recognition error, and
- the development of a parallel processing architecture based on shared memory.

1.4 Contribution

A static look-and-move robotic visual servoing system (RVSS) was demonstrated, which identified, located, and servoed the PUMA-560's end-effector into close proximity

of the refueling receptacle. The initial architecture for a parallel processing system required for a dynamic look-and-move RVSS was developed. The RVSS provides AFIT with the capability to meet Air Force initiatives for research on the design, development and application of visual servoing for autonomous aircraft refueling tasks.

1.5 Organization

In order to better understand the detailed implementation of these research activities, this thesis is organized as follows. Chapter 2 provides a review of literature on key design issues required for successful demonstration of this refueling concept: robotic visual servoing, image brightness preprocessing, object recognition, and visual ranging methods. Chapter 3 provides a detailed analysis of the algorithms developed during this thesis effort for visual recognition. In Chapter 4 a detailed analysis of the visual servoing algorithms, also developed for this thesis effort, is presented. Chapter 5 discusses research accomplishments and problems encountered during the implementation of this refueling concept. Finally, Chapter 6 presents the conclusions and recommendations for future refueling research. All software is documented in an AFIT Robotic Systems Laboratory (ARSL) internal report (ARSL-89-14) [7].

In addition, two appendices are provided. The first presents the hardware used in completing and demonstrating this concept, and the second describes the software logic and flow during program execution for the static look-and-move demonstration.

II. Review of Literature

2.1 Introduction

The development and implementation of a robotic visual servoing system (RVSS), especially one for autonomous aircraft refueling, is a difficult task. Such a task requires the interfacing of both robotic and vision technologies. Therefore, a review of current experimentation in the area of robotic visual servoing was considered necessary, not only for possible solutions to problems encountered in this project, but to become familiar with the current focus of research being performed for both autonomous and visual servoing tasks. Several additional areas of review were determined to be critical research areas for this project. These areas included visual ranging methods, object recognition techniques, and image brightness preprocessing.

2.2 Robotic Visual Servoing Research

Over the past several years, there have been many research projects conducted using visually servoed robotic manipulators to perform tasks. The use of visual servoing robotic systems appears to satisfy a myriad of tasks of great breadth and depth. Although much research has been performed using stationary cameras mounted above a work area [8,9], only research dealing with robot mounted cameras was pertinent to this project due to the increased interaction between camera and robot. Therefore, both static and dynamic control methods for servoing were reviewed in regard to robot mounted camera systems.

There are two basic types of control schemes for visual servoing systems, static and dynamic feedback. Static feedback maintains the robot in a stationary position while the image is acquired and processed. Manipulator motion is performed after vision processing is complete, and no further information from the vision system is provided during this motion. In general, this form of control system operates open-loop. Dynamic feedback on the other hand allows for real-time robot motion based on informational updates acquired from periodic processed vision data. In general, this form of visual servoing control is closed-loop [1:12]. The use of a particular visual servo control scheme is based on the

task requirements. Static look-and-move may be appropriate for ground-based refueling, whereas aerial-based refueling requires dynamic look-and-move.

A unique area of robotic visual servoing research is that of the Flexible Intelligent Garment Assembly RObot (FIGARO) sewing system. FIGARO was developed to perform assembly and handling operations on cloth. In this automated case, FIGARO was required to control cloth panels during both handling and sewing operations. This system employed a technique for producing a seam parallel to an edge of arbitrary contour based on visual servoing [10]. A PUMA-560 with a mounted low resolution vision system was at the heart of FIGARO [10:1831-2]. Figure 2.1 shows FIGARO at work.



Figure 2.1. FIGARO at Work [10:1831]

The cameras used in this system provided very low resolution (30 x 32 pixels), but were deemed adequate for the task since they were located very close to the sewing table [10]. An IBM AT was used as the main system controller supplying trajectory updates for the system (at a very slow rate – one second range) [10:1831]. For refueling applications, high camera resolution is required since the location of the desired object is unknown. So,

although FIGARO provided relatively good results, the overall approach used would be inadequate.

In a different application of robotic visual servoing principles, Harrell et al. developed a vision guided robotic tree fruit harvester. The vision system, located in the harvester's end-effector, provided real-time two-dimensional fruit location information. The robotic harvester system was based on a 3-DOF manipulator with a prismatic terminating link and global machine vision. Fruit was identified and then its centroid location was calculated. Based on this location and the relative position between the camera and the end-effector, a trajectory was calculated and the third (prismatic) link was extended until touch sensors sensed contact with the fruit [11].

Although the object of interest (the fruit) can be considered a static target in contrast to tracking a moving object, two important conclusions were produced by this research. After the fruit was identified, the vision system attempted to maintain the fruit in the center of the image as the prismatic link was extended [11:543]. This self-centering ability allowed the manipulator to continually servo the two rotary joints keeping the end-effector in alignment with the fruit as it moved forward towards the fruit. The second important contribution was the derivation of the camera to fruit ranging methodology. Range was calculated using the horizontal diameter of the fruit as determined from its image. The derived equation showed that range increased as diameter distance decreased, and vice versa [11:542]. These two conclusions are key elements to be utilized in this thesis effort.

More recently, Feddema and Mitchell performed experiments in which a PUMA-600 robot visually tracked a carburetor gasket on a circular parts feeder. Their camera was mounted on the last link of the PUMA-600. Object recognition was based on the circles of the gaskets, as shown in Figure 2.2, and their relationship in reference to each other. Feddema and Mitchell's approach successfully tracked the gasket using vision feedback until the end-effector was positioned accurately enough to pick up the gasket [12:11]. Problems occurred when the speed of the parts feeder increased. Either the visual servo system could not keep up with the gasket and would lose it, or accuracy decreased due to sensor motion time lags. As before, two key conclusions of this research were produced. The first conclusion was that asynchronous sensory input allows two functions to operate in parallel

without introducing system degradation due to one process waiting for another process's output. For a system where time delays in data processing occur, such as is inherent in current vision processing, parallel processing is a requirement. The second conclusion was that geometric indicators (internal in this case) can be used to simplify object recognition. Again, both of these conclusions provide helpful information in the development of this thesis effort.

2.3 Image Brightness Preprocessing

A significant problem with vision systems used for recognition purposes is illumination effects (i.e., shadows, glare, and lighting unevenness) on a object. In an environment where controlled lighting intensity is maintained, such as a laboratory or industrial work area, the effects of lighting on an object are minimized, and simple, quick techniques can be used in preprocessing the image for further use. Unfortunately, some environments, such as factories or outdoor areas, do not provide the desired structured lighting and, therefore special, possibly time consuming, brightness preprocessing algorithms must be applied to the image before the image can be manipulated further. Two common types of image brightness preprocessing used today are thresholding and brightness normalization.

2.3.1 Thresholding Thresholding is probably one of the simplest types of preprocessing currently in use [13:357]. Thresholding in its simplest form is merely a binarized image with each pixel being in one of two states (0 or 1), or in this case, black or white. The generation of a binarized image from a more general gray-scale image requires the introduction of a thresholding level to partition the image into pixels with black or white values. The general approach to creating a binary image is based on the relationship where $f(x,y)$ is the intensity value associated with the pixel located at image coordinates (x,y) and T is the thresholding value [14:17]. By applying the following rule,

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) < T \\ 1 & \text{if } f(x,y) \geq T \end{cases} \quad (2.1)$$

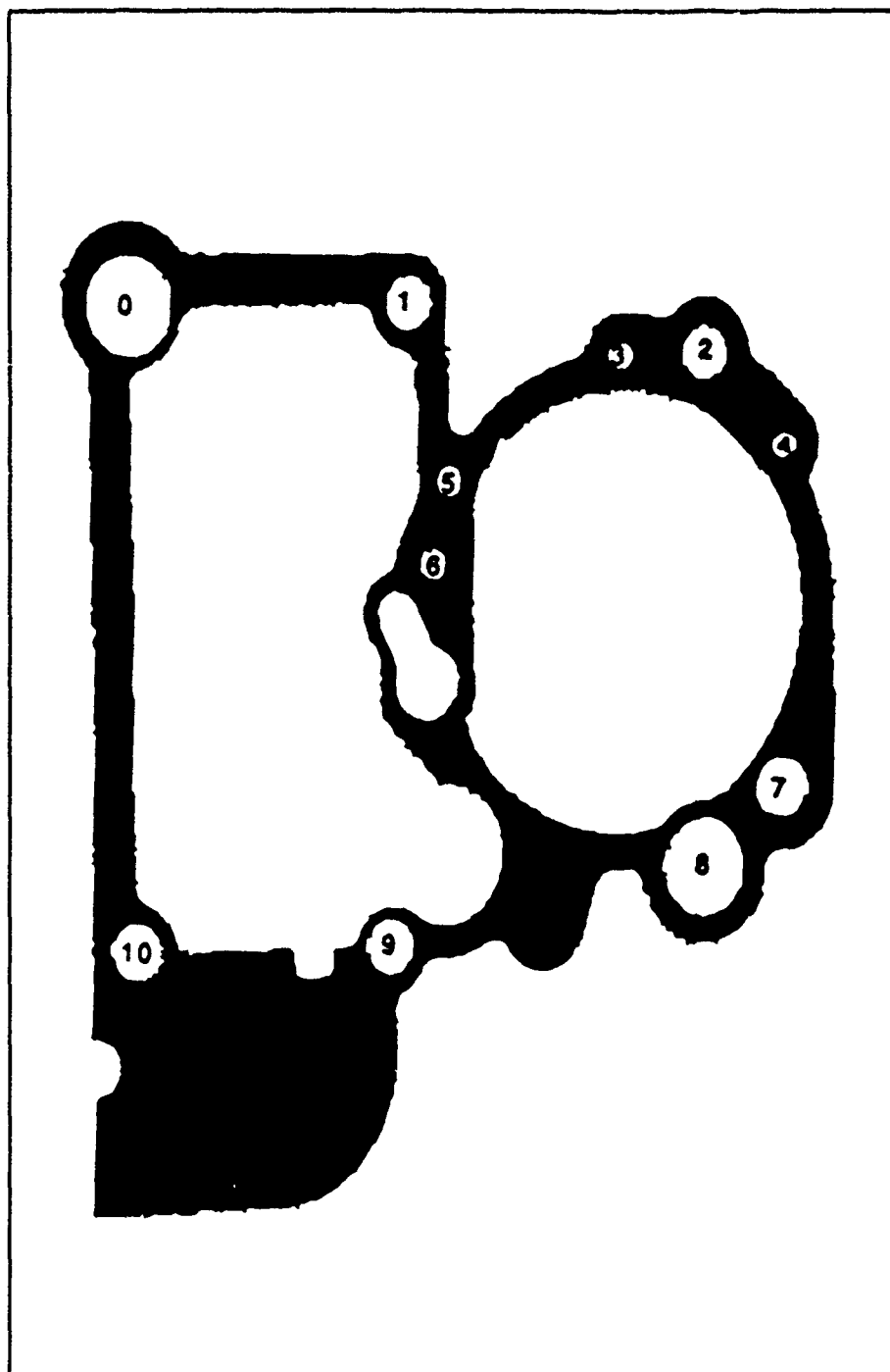


Figure 2.2. Recognition of Gasket Based on Geometry [12:696]

the image is binarized (where T is a nonnegative value and bound within the gray-scale values) [13:356]. Thus, based on $g(x,y)$, the object can be associated with pixels having a value of 1 and the background corresponds to pixel value 0 [13:358], or vice versa as the case may be.

Determination of the thresholding value can be quite complex. The threshold value can be determined either statically or dynamically. Static determination is simplest and consists of T remaining a constant value for all images being processed. This type of thresholding is useful for repetitive image analysis of the same scene where the environmental lighting changes are minor. Dynamic determination consists of constant reevaluation of T for each image processed. This is quite useful for analysis of nonrepetitive images or constantly changing environmental lighting conditions. A basic way to find a threshold value for an image is to use image histograms. Each time an image is acquired, a histogram is produced and evaluated based on the raw image pixel data. Information from histograms, such as maximum intensities or positions of intensity peaks, can be used to identify or calculate a threshold value [13:374]. The research of Fiedema and Mitchell, previously discussed in this chapter, is an example of an actual application of static thresholding in robotic visual servoing.

2.3.2 Brightness Normalization When lighting is unstructured, and shadows and glare are introduced in the image, simple thresholding does not suffice and other methods must be used to preprocess the image. Capt. Lawrence Lambert, while a student at AFIT, developed an algorithm which performed a brightness normalization on an image. This algorithm made the image invariant to the overall brightness of the input scene [6]. Lambert's algorithm calculated the average brightness of a square-shaped neighborhood centered around the pixel of interest, and then reset that pixel's value using the following equation:

$$pixelvalue = threshold + (pixelvalue - neighborhoodaverage) \quad (2.2)$$

This threshold value, quite different from the thresholding value previously discussed, is merely a referenced average value defined by the user. In a sense, the "darkness" of a

pixel is measured relative to a fixed average value (i.e., the threshold value). This process is performed on every pixel in the scene with each pixel having its own unique neighborhood of sorts [6:3-17]. After brightness normalization, further processing, such as thresholding, can be performed to again binarize the image. One side effect of this brightness normalization is that the algorithm acts as both a brightness normalization scheme and an edge detector. Therefore, although a binarized image is produced, complex recognition algorithms may be required to process and identify objects in the image. Another issue associated with this type of algorithm is one of processing time. For accurate visual servoing, processing time of an image should be as minute as possible, thereby allowing quick image feedback to the robot controller. Currently, Lambert's brightness normalization algorithm requires roughly three seconds processing time on a VAXstation III. Three seconds is unacceptably long for visual servoing needs. For actual implementations, CPU time associated with brightness normalization must be reduced.

2.4 Object Recognition

Recognition is mainly a labeling and interpretation process. The function of recognition algorithms is to identify each object in a scene and assign labels to them [13:424]. Inherent in recognition algorithms are assumed constraints that are to be applied to scene analysis. These constraints may deal with the known geometry or orientation of objects, or that each object is a unique entity and not part of a further whole [13]. Probably the most difficult aspect of object recognition is the ability to recognize a three-dimensional object from a two-dimensional image. Fortunately, two principle approaches exist for recognition: decision-theoretic (quantitative descriptors) and structural (symbolic descriptors and their relationships) [13:425].

2.4.1 Decision-Theoretic This approach makes use of quantitative descriptors for pattern recognition, and is based on decision functions. These decision functions are used to perform matching for object recognition. Each object to be found is provided a decision function in the form of an average vector. By comparison of regions within the image, a correlation coefficient is calculated and the closest match "wins". Problems with this

technique are object size and orientation in object comparison [13:425-7].

Another aspect of this type of recognition is the overall matching by a known template or silhouette to an extracted object. Work on template matching has been attempted in a variety of ways. Template matching is the process of determining the position and orientation of an object in an image. By using centroid properties of a region (or more precisely invariant moments), rotation analysis of the object compared to the template can be performed quite quickly [15:339]. This process involved shifting the template over the search area and computing the similarities of the area. If multiple regions exist in the image [15:340], again the "best" comparison is chosen.

The simplest form of template recognition is that of a binarized image, where the object consists of pixels with a value of 1 and the background consists of pixels with a value of 0 (values of 1 and 0 refer to the pixel values assigned during thresholding). If any pixels in the image have a value of 1, then the object is considered identified because the actual template is null and recognition is achieved by the existence of white pixels in the image. This technique was used by Feddema and Mitchell in their work.

Silhouettes are very useful in the recognition of three-dimensional objects in a known orientation. This technique also utilized regional recognition patterns to obtain the desired results, but most results are obtained through multiple views of the object of each of its three primary axes [16]. Based on some desired coordinate system, two measurements are made of the object: volume and mass distribution. These parameters are processed through Fourier transforms to develop contour coefficients. By comparing silhouettes to the contour coefficients of extracted regions, identification can be performed [17:513].

2.4.2 Structural This approach makes use of qualitative descriptors for pattern recognition, and distinctly ignores any geometric relationship that exists by representing the boundary of an object by a "string" of pattern primitives (structural decomposition of an object). The length and order of this string establishes the structure of the object. For recognition, the string is compared to coded strings in the recognition algorithms for possible matching [13:427]. Figure 2.3 provides an example of this technique.

Interpretation may be viewed as the process which endows a vision system with a

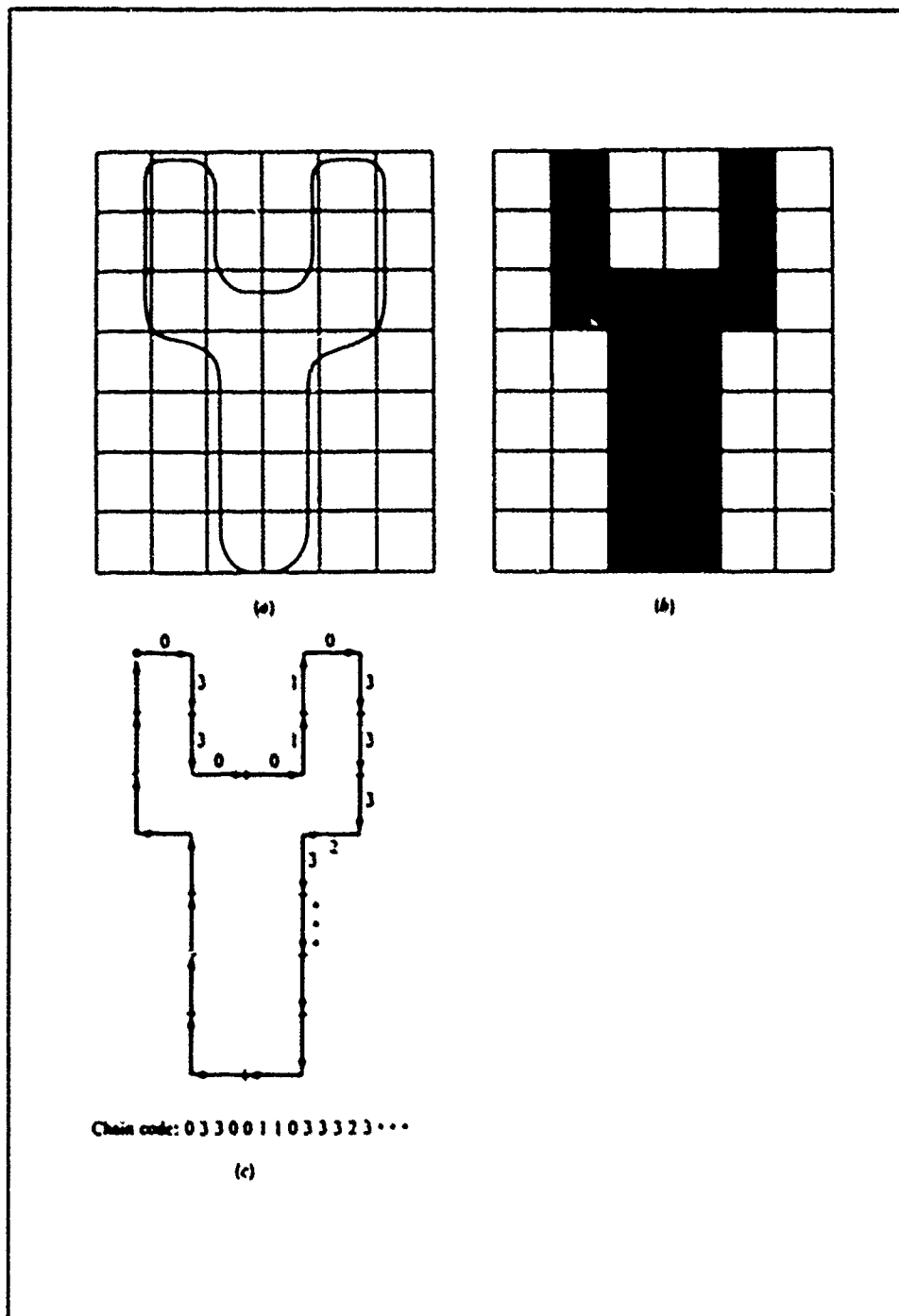


Figure 2.3. Structural Method of Recognition (a) Object, (b) Primitives, and (c) Object in Terms of Primitives [13:427]

higher level of cognition about its environment. The power of vision systems, image preprocessing, and recognition algorithms are determined by their ability to extract meaningful information from a scene under a broad range of viewing conditions while using minimal knowledge about the objects being viewed. There are a number of factors that make this type of processing a difficult task including: variable illumination (as discussed earlier), occluding bodies, and viewing geometry [13:439].

2.5 Visual Ranging Methods

Determining the range distance between a reference point and an object is an important problem in applications such as tracking objects in industrial or military environments. Active ranging devices can be fast and efficient, but they require projection signals (e.g., ultrasonics, laser beams) or patterns (e.g., grid of lights on the viewed object [18:80]). Unfortunately in the military environment, active devices tend to provide adversaries with indicators to the location of the source of signal generation. Therefore, the use of passive ranging devices, such as video cameras, is more desirable, yet more complex. In this case, techniques using object characteristics or multiple cameras can be used to determine range. Since this project is limited to single vision systems, stereo vision techniques were not reviewed.

Probably the simplest method of range determination is that based on prior knowledge of the object in question. In this instance, the dimensions or position of the object are known beforehand. Hung et al. performed just such an experiment to develop a procedure for determining the range and pose of a three-dimensional planar quadrangle referenced to the camera-centered coordinate system [18:80]. In addition, a technique for transforming from camera coordinate system to world coordinate system (based on some fixed reference point) was developed. Hung et al. used the known dimensions of the sides of the quadrangle and the image derived quadrangle vertices to determine, in a single iteration, the three-dimensional positions of the four vertices. Determining the vertices positions was accomplished by a simple transformation technique which transformed the coplanar image object vertices positions to the coplanar quadrangle vertices positions. Their results showed that prior knowledge of object parameters allowed for easy and quick range deter-

mination. Also, larger viewing angles (i.e., the image size) also provided more accurate range measurements [18].

Kim et al. used a different technique in determining range data for a visually servoed robot manipulator. The object of their experimentation was to locate and grasp a stationary object [16:417]. Unlike Hung et al., no prior knowledge of the objects dimensions, except those needed for recognition, were known. They devised a geometric technique which derived a ranging equation utilizing two images obtained from different positions lying in the same line-of-sight of the camera [16]. By maintaining a constant speed and attempting to keep the object centered in the image field, the derived equation produced fairly accurate results.

2.6 Summary

From this review, several concepts surfaced that may be of use in the development of this thesis effort: object self-centering, image thresholding, brightness normalization, and range determination based on known object parameters. Use of these concepts should reduce the research required to successfully develop a robotic visual servoing system for aircraft refueling. Therefore, the development of object recognition and visual servoing algorithms will be presented next.

III. Refueling Receptacle Recognition

3.1 Introduction

Recognition of the refueling receptacle in real-time is one of the key tasks of this thesis effort. Therefore, the criteria used to develop a refueling receptacle recognition algorithm was one of simplicity and speed. This chapter describes the development of the recognition algorithms and discusses their rationale and usage. This presentation will be divided into the following areas:

- *ITEX 100 Image Processor* details how vision data is stored, referenced, and manipulated by the ITEX 100, allowing for a better understanding of image analysis for recognition purposes.
- *Structured Environment Research* utilized simple black and white environment to visually servo the PUMA-560 for verification of search and kinematics algorithms, and development of a simple recognition algorithm.
- *Unstructured Environment Research* utilized the real-world environment to perform static look-and-move visual servoing based on the algorithms developed in the structured environment research.
- *Modified Brightness Normalization* modified Lambert's brightness normalization algorithm to preprocess real-world images in unstructured illumination areas.

One assumption required for the development of the recognition algorithms was that a structured lighting environment always be present. This preprocessing or structured lighting provided images that were easily and more accurately thresholded.

In order to clearly understand this topic, definitions for a scene and an image are necessary. A scene is the actual environment found in the field of view (FOV) of the camera. An image is the scene after it has been digitized by the ITEX image processor.

3.2 ITEX 100 Image Processor

In order to discuss how the recognition algorithms work, an understanding of the operations of the vision processing must first be established. All vision processing for this research was performed using an Image Technology Series 100 image processor board. Every time an image was acquired, the image was digitized into a two-dimensional array (512 x 512 pixels) and placed in a frame buffer memory area where the array could be manipulated [19:3]. Each pixel in the Series 100 Frame Buffer has a 12-bit value, but in this application only the first eight bits were actually used (the upper four bits were used for special ITEX functions not used for this application), thereby allowing pixel values ranging between 0 and 255. Each pixel value was determined by subsequent cascading of threshold values representing equal gray-level values between black (0) and white (255) [19:9]. In order to manipulate these pixel values (i.e., read, write, modify), the ITEX 100 comes with a standard library of image processing and graphics functions written in C. Through extensive use of these functions, image manipulation could be performed.

As stated previously, the pixels were stored in a two-dimensional array. When used or displayed on a monitor, the array of pixels has an origin at the uppermost left corner representing location (0,0) and the lowermost right corner is considered location (511,511). When viewing the image on a standard monitor, only the top 480 horizontal lines are displayed (the other 32 lines are considered below the bottom of the screen, but still useable) [19:6]. Figure 3.1 provides an illustration of the video coordinate system. For ease of use and testing purposes, only the top 480 lines were used for this thesis effort. Also, as with many monitors, there was an inherent problem with unequal x and y aspect ratios. Therefore, circles appeared ellipsoidal, and vice versa, on the monitor. For this system, the aspect ratio was 4:5 (y:x) [19:12-4].

3.3 Structured Environment Research

In order to verify the search routine and the kinematics algorithms developed (subjects to be discussed in Chapter 4), a simple structured environment (black and white) recognition algorithm was developed. Utilizing the circular nature of the refueling recep-

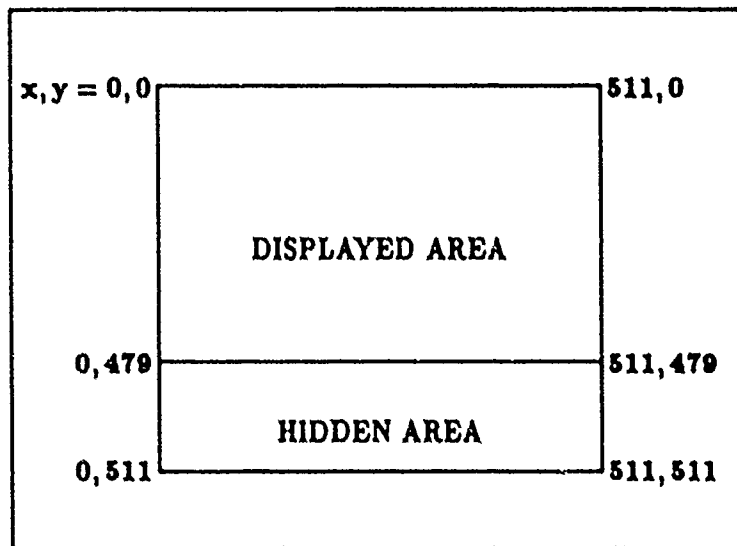


Figure 3.1. Video Coordinate System [19:6]

tacle as the object of interest (or target), a black mask, made of black construction paper, was overlayed onto the refueling receptacle test stand. A white paper circle (about seven centimeters in diameter) representing the receptacle was placed over the actual receptacle location of the test stand. Figure 3.2 shows the mask in place on the refueling receptacle test stand. Additional black construction paper surrounded the test stand to ensure that only white and black scenery was in the camera's FOV during the PUMA's search for the receptacle.

Sharp contrast between the background and the circular disk (which must be the only white object in the scene) representing the refueling receptacle was essential to the correct operation of this algorithm. This structured environment recognition algorithm used the concept of side parameters to locate and identify the refueling receptacle. That is to say, the refueling receptacle was recognized as being white and possessing a top and bottom, and left and right sides. Simple thresholding was used to differentiate between white and black pixel values. After extensive analysis of histograms taken of images possessing the white object, black was found to always be below pixel value 128 and white was always above 128. Therefore, based on a threshold value of 128, an image search for the refueling receptacle was performed.

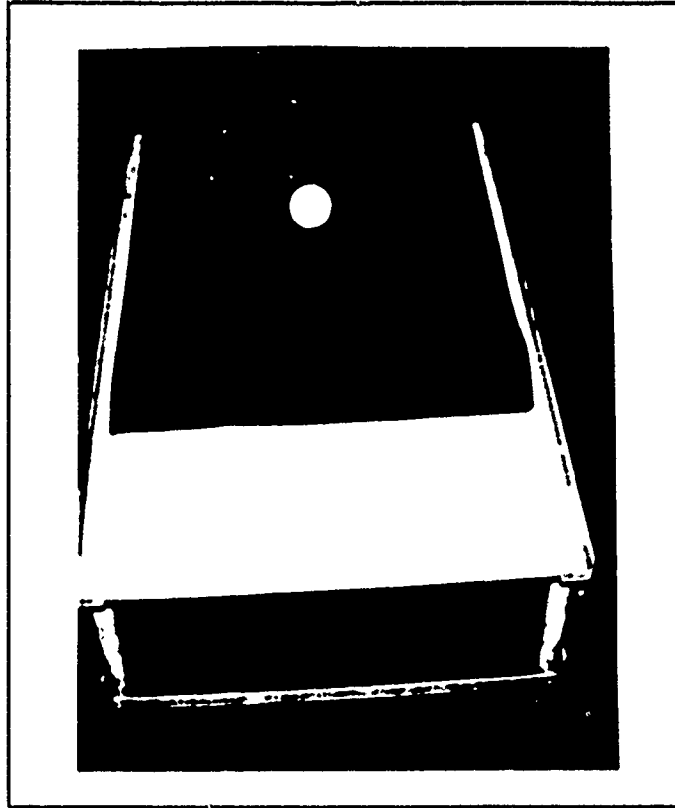


Figure 3.2. Refueling Receptacle Test Stand with Mask

The search first looked for the top of the object by reading in the first and subsequent horizontal row of pixels until a row with a white pixel was found. This horizontal row was considered the top of the object. Next, the search for the bottom of the object continued by reading subsequent horizontal rows of pixels from the top of the object. When a row was found that had no white pixels or the bottom of the screen was reached, this row was considered the bottom of the object.

The search for the sides began with the left side. Starting at the leftmost part of the image, the first and subsequent vertical columns of pixels were read until a vertical column with a white pixel was found. This vertical column was considered the left side. Continuing the search for the right side, subsequent vertical columns were read until a column with no white pixels was found or the right edge of the screen was reached. This vertical column was considered the right side. If all the sides were found, then the object

was considered found. Figure 3.3 illustrates the outcome from searching for the white disk in the image. In the event that the object was situated on one or more of the edges of the image, one or more sides would not be identified and the object would be considered not found. In this case, the PUMA would rotate joint one or three, 1° in the direction required to move the object towards the center of the image, and another frame would be acquired and passed through the recognition algorithm.

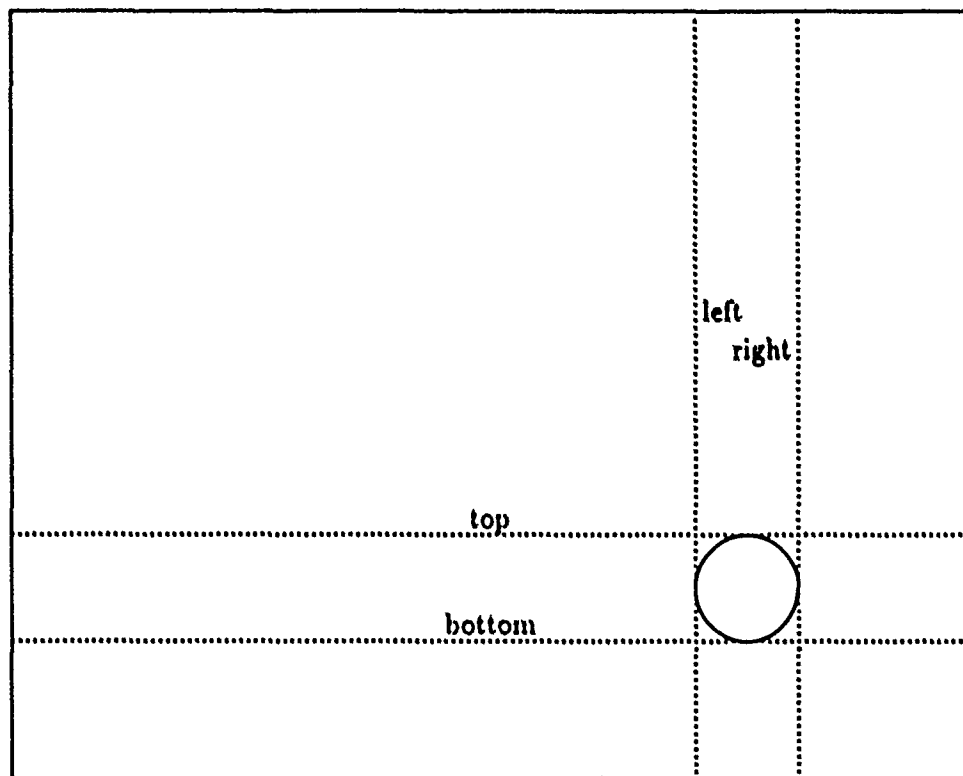


Figure 3.3. White Object Search

Once the object was found, the object's offset distances from the center of the image and the range of the object from the camera was calculated. These parameters were necessary to verify and validate the forward and inverse kinematics. The first step was to determine the location of the center of the object (referenced to the upper left of the image) based on the side locations:

$$centery = (bottom - top)/2 + top \quad (3.1)$$

and

$$center_z = (right - left)/2 + left \quad (3.2)$$

The center of the object was a key point for performing tracking and subsequent contacting of the refueling receptacle because this center point was the desired contact location of the refueling nozzle to the refueling receptacle test stand.

Next, the offset of the object from the center of the image (defined to be 255,255) was determined:

$$deltay = (255 - centery) * transy \quad (3.3)$$

and

$$delta_z = (255 - center_z) * trans_z \quad (3.4)$$

Here *transy* and *transz* were the conversion multipliers needed to convert pixels to meters. These delta values determined how far and in which direction joint one must be rotated to center the object in the image in order to align the end-effector with the refueling receptacle. Figure 3.4 provides an illustration of the relationship of the center of the object, the offsets of the object, and the center of the image to each other.

Finally, the range was determined using the following equation derived later in this chapter:

$$range = 77.846 / width^{0.1027} \quad (3.5)$$

allowing the kinematics to be applied, and the PUMA to move towards the location of the refueling receptacle (white circular disk).

3.4 Unstructured Environment Research

The next phase of recognition research was the development of an unstructured environment recognition algorithm. In this case, the the requirement for black and white structured scenery was deleted. In addition, tests were performed using different indicator structures, possibly aiding in the further recognition of the refuel. receptacle. The refueling receptacle test stand was painted gray to emulate the primer coating the skin of aircraft refueling receptacles. Also, special indicators were placed around the refueling receptacle slipway for use in the recognition algorithm.

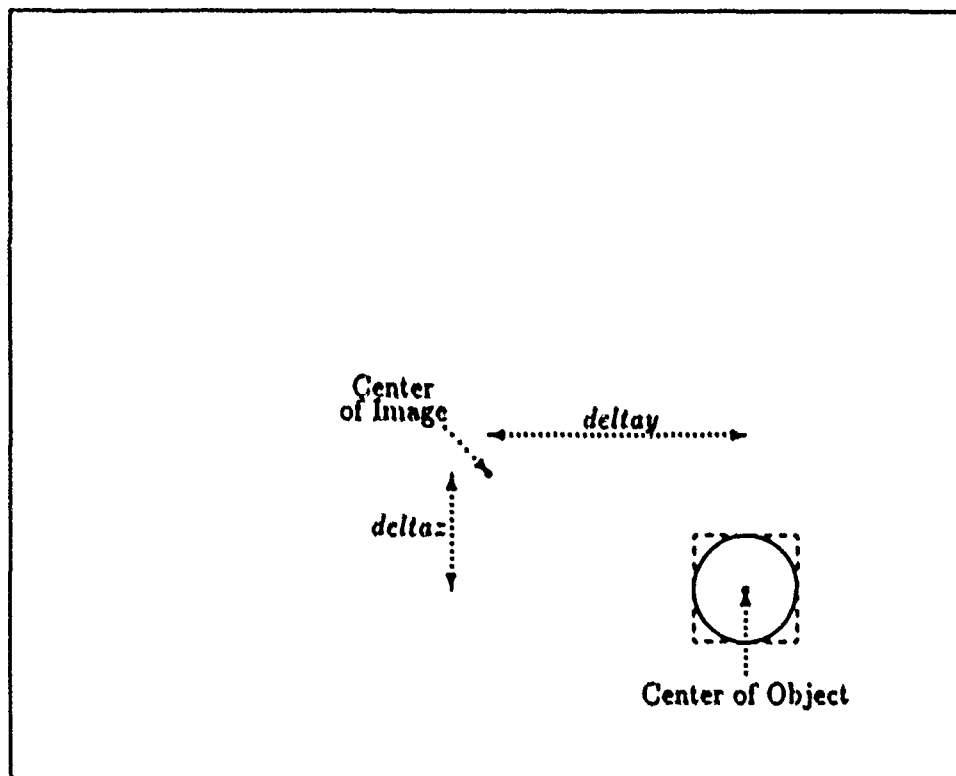


Figure 3.4. Relationship of Object Parameters

The use of indicators was deemed appropriate for several reasons:

- (a) these indicators are small and passive, therefore no emissions of any sort are generated and no increased risk to plane or pilot is introduced,
- (b) for a large fleet of aircraft, the placement of these indicators onto aircraft would require minimal cost and effort,
- (c) the use of indicators decreased the complexity of recognition, thereby decreased processing time, and
- (d) indicator design and placement are left to the discretion of the system development team.

The design and placement of indicators for this research effort was a representative case and was not considered "optimal."

The indicators chosen for this research were three rectangular strips made from white construction paper. White was chosen because white is one extreme of the gray-level spectrum used in image generation. Therefore, these indicators were the whitest objects in the area of the refueling receptacle (compared to the gray primer background), and this fact was used in developing a simpler recognition algorithm. White was not a requirement; its use merely simplified the recognition algorithm. In all actuality, any color could have been used, impacting only the thresholding level(s) evaluated. Two of the indicators were placed along the side of the slipway and the other was placed along the top of the slipway, as shown in Figure 3.5. In this way, the three indicators formed a three sided box around the refueling receptacle, thereby allowing the center of the box to be roughly the desired receptacle position of contact. Three indicators were deemed the minimal number required to provide center of mass, pose and pattern uniqueness data.

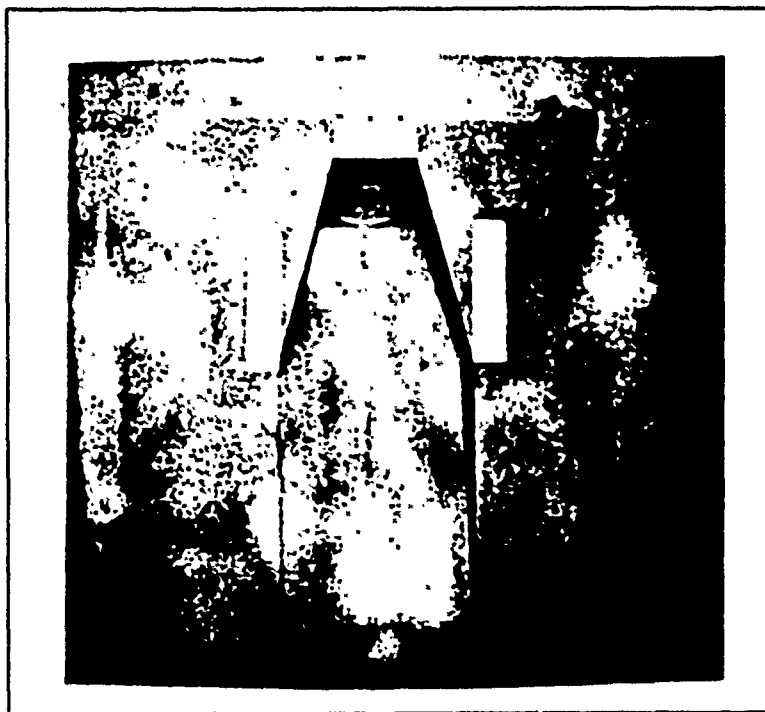


Figure 3.5. Placement of the Special Indicators

The unstructured environment recognition algorithm worked in roughly the same manner as the black and white algorithm, except that after an image was acquired, a

histogram was performed on the image and the subsequent data stored in an array. Starting at the white end of the spectrum (pixel value 255) and regressing, an element of the histogram array was compared to its two neighbors to detect a peak. In short, was the element in question larger than its two neighbors? If the number of pixels with this peak value or greater was larger than 25, then this pixel value was considered the white pixel value. In order to provide flexibility about this value, the white pixel value was decreased by 10 percent of its value; thereby, providing a new pixel value considered to be the white limit. The value 25 was chosen to eliminate the possibility of small peak detection; peaks too small to be the desired peak value. Any pixel whose value was equal to or greater than the white limit was considered white, everything else was considered black, and the image thresholded appropriately.

In a 480 x 512 pixel array, exactly 245,760 pixel elements exist. Through extensive analysis of images containing the three indicators but with the receptacle's distance from the PUMA varied, a range area of the number of white pixels capable of representing the three indicators was determined. If the number of white pixels in the image was less than 100 or greater than 100,000, then the refueling receptacle was considered not to be in the image and no further recognition processing would be conducted on this image. If the number of white pixels was within this range, then the center of mass of these white pixels was found:

$$\bar{x} = \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \quad (3.6)$$

and

$$\bar{y} = \frac{\sum_{i=1}^n y_i w_i}{\sum_{i=1}^n w_i} \quad (3.7)$$

where w_i is the weight associated with each pixel element i , and x_i and y_i are the coordinates of the pixel element i in the image. Assigning a weight of one to each white pixel element, these equations reduce further:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (3.8)$$

and

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (3.9)$$

since n (the number of white pixels in the image) constituted the total weight of white pixels in the image.

Finding the center of mass was critical for the recognition algorithm. If the receptacle was in the image, then the only white pixels would be those corresponding to the three indicators. Therefore, the center of mass should lie somewhere between the three indicators, as shown in Figure 3.6(a). By checking every horizontal row above the center of mass until one was found that had no white pixels in it, the image was divided in half leaving the top indicator in one region and the other two indicators in the other region, as shown in Figure 3.6(b). By using the same routine used in the structured environment recognition algorithm for finding the top, bottom, left and right sides of an object, but reducing the search area to the top half of image, the top indicator was found. Once the top indicator was found, the lower region of the image was divided vertically based on the horizontal center of the top object, as shown in Figure 3.6(c).

First the left object was found, using the same side parameter recognition algorithm developed for the structured environment case, and then the right object was found, but within the newly defined search areas enclosing each object. If any side parameter of any of the three objects was not found, then the objects were considered not to be the refueling receptacle indicators. If all the side parameters were found for all three indicators, then the following tests had to be satisfied to ensure that the objects found could be considered the refueling receptacle:

- Is the left side of the top object to the right of the left side of the left object?
- Is the right side of the top object to the left of the right side of the right object?
- Is the height of the top object less than the height of the left object?
- Is the height of the top object less than the height of the right object?
- Is the width of the top object greater than the width of the left object?
- Is the width of the top object greater than the width of the right object?
- Is the area of the right object within ± 10 percent of the area of the left object?

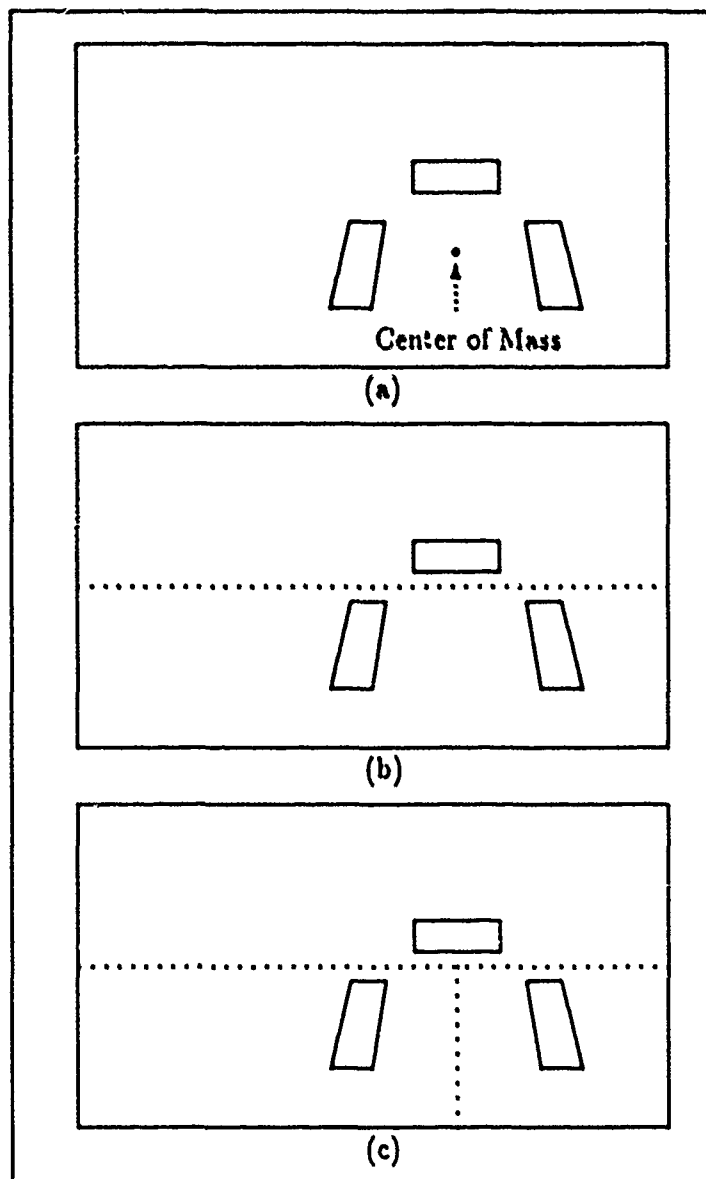


Figure 3.6. Indicator Identification (a) Center of Mass of White Pixels, (b) Image Division Separating Top Indicator From Lower Indicators, and (c) Division of Lower Area Separating the Left and Right Indicators

If any of these criteria were not met, then the object was considered not to be the refueling receptacle and the search continued into the next search area. Once the refueling receptacle was recognized, then the range, as well as, the location parameters were determined. These parameters were determined in the same manner as in structured environment with the center of the object and the offsets calculated based on location and area covered by the three indicators. Figure 3.7 illustrates the relationship of the object parameters for the unstructured environment (i.e., refueling receptacle indicators).

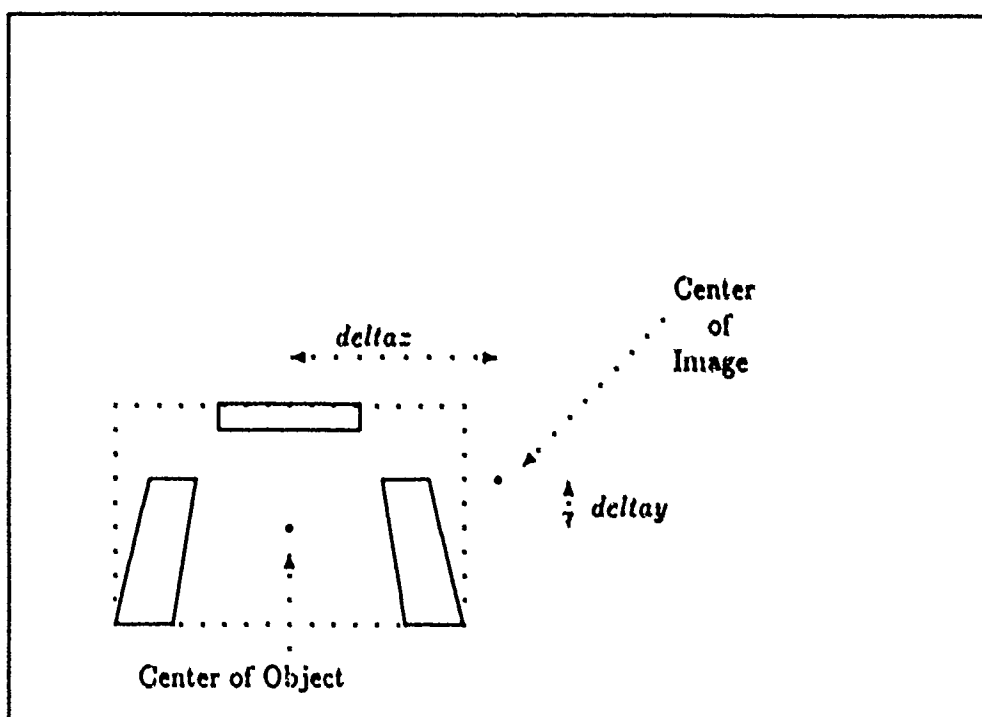


Figure 3.7. Relationship of Indicator Parameters Upon Identification

3.5 Range

The range was calculated based on the relationship of object width to object distance from the camera:

$$width = right - left \quad (3.10)$$

A basic assumption for demonstrating this refueling task was that the approach of

the refueling receptacle to the PUMA would be a forward, direct approach within a 10° horizontal arc. This arc was to emulate the approach pattern used during aerial refueling of a target aircraft. Through numerous image analyses, the width of the object was noted to change (± 2) pixels at the most from transition from side to side.

Due to the geometric nature of the range in regard to the width of the object, the proper range equation used to calculate the distance of the refueling receptacle from the camera was required. This range equation was applicable to both the structured and unstructured environments since the width of the circular disk and the top indicator (used for range determination) were identical, respectively. Figure 1.3 shows the general relationship between the camera and the refueling receptacle upon identification of the receptacle from an acquired image.

As stated above, the range of the refueling receptacle from the camera was based on the width of circle (for black and white) or the top indicator (for real-world scenes) in pixels as determined from an image. The derived range equation was based on the top indicator width of the real world scene. Indicator width and range were inversely proportional. As the receptacle drew closer to the camera, the width of the top indicator increased while range naturally decreased, and vice versa.

A numerical approach was used in the derivation of the range equation. Multiple data points (pixel width vs. range) were acquired experimentally and curve fitting attempted. Table 3.1 provided the sample points of pixel widths at different measured ranges. The data ranged from the closest position possible for the refueling receptacle test stand to the PUMA, out to about four meters from the PUMA base. The data has been rounded to the nearest five pixels to ease the derivation, since slight movement of the test stand revealed little or no changes in indicator width (in pixels). When plotted against cartesian coordinates, the semi-parabolic curve, shown in Figure 3.8, is generated.

Both manual and automated attempts were made to curve fit the data into linear, quadratic and cubic equation forms, but these attempts were fruitless and the equation constants did not provide consistent results. Therefore, analysis was performed on the assumption that the relationship between the two variables may be logarithmic due to

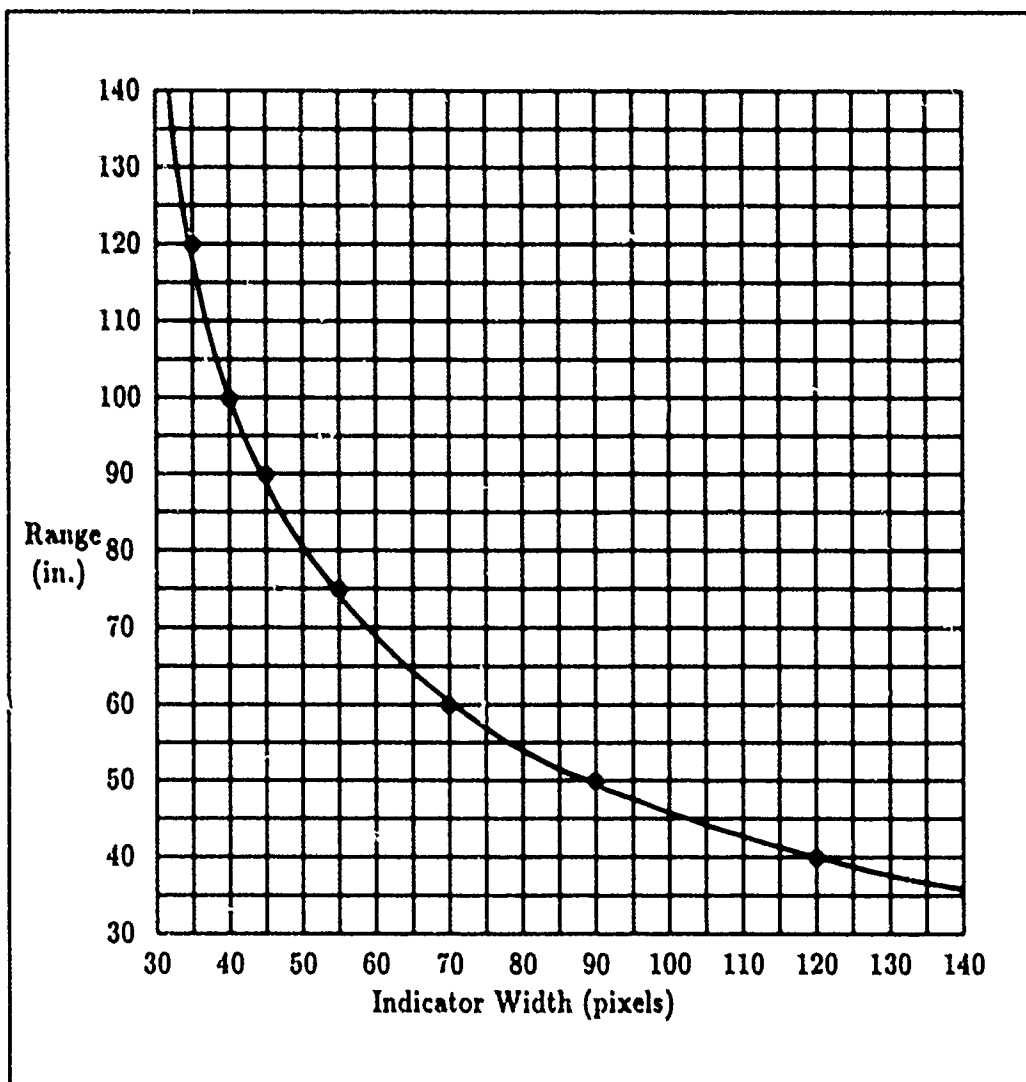


Figure 3.8. Indicator Width (pixels) vs. Range (in.)

Table 3.1. Indicator Width vs. Range

Width (pixels)	Range (in.)
35	120
40	100
45	90
55	75
70	60
90	50
120	40

distortion factors of optical lenses. Plots of the data using cartesian vs. logarithmic and logarithmic vs. logarithmic scaled are shown in Figure 3.9(a) and (b).

These plots, especially that of Figure 3.9(b), supported the assumption that the relationship between range and indicator width appeared to be a logarithmic function having the form:

$$\ln y = d - c \ln x \quad (3.11)$$

where

y is the range,

x is the width of the top indicator,

c is the inverse multiplier, and

d is the asymptotic approach base.

To determine the values of c and d , simple algebraic function subtraction was performed using the two end-points of the data obtained. Substituting the appropriate values of range and indicator width into Equation 3.11 for both end-points, the following equations for the first and last data points were derived:

$$\ln(35) = d - c \ln(120) \quad (3.12)$$

and

$$\ln(120) = d - c \ln(40) \quad (3.13)$$

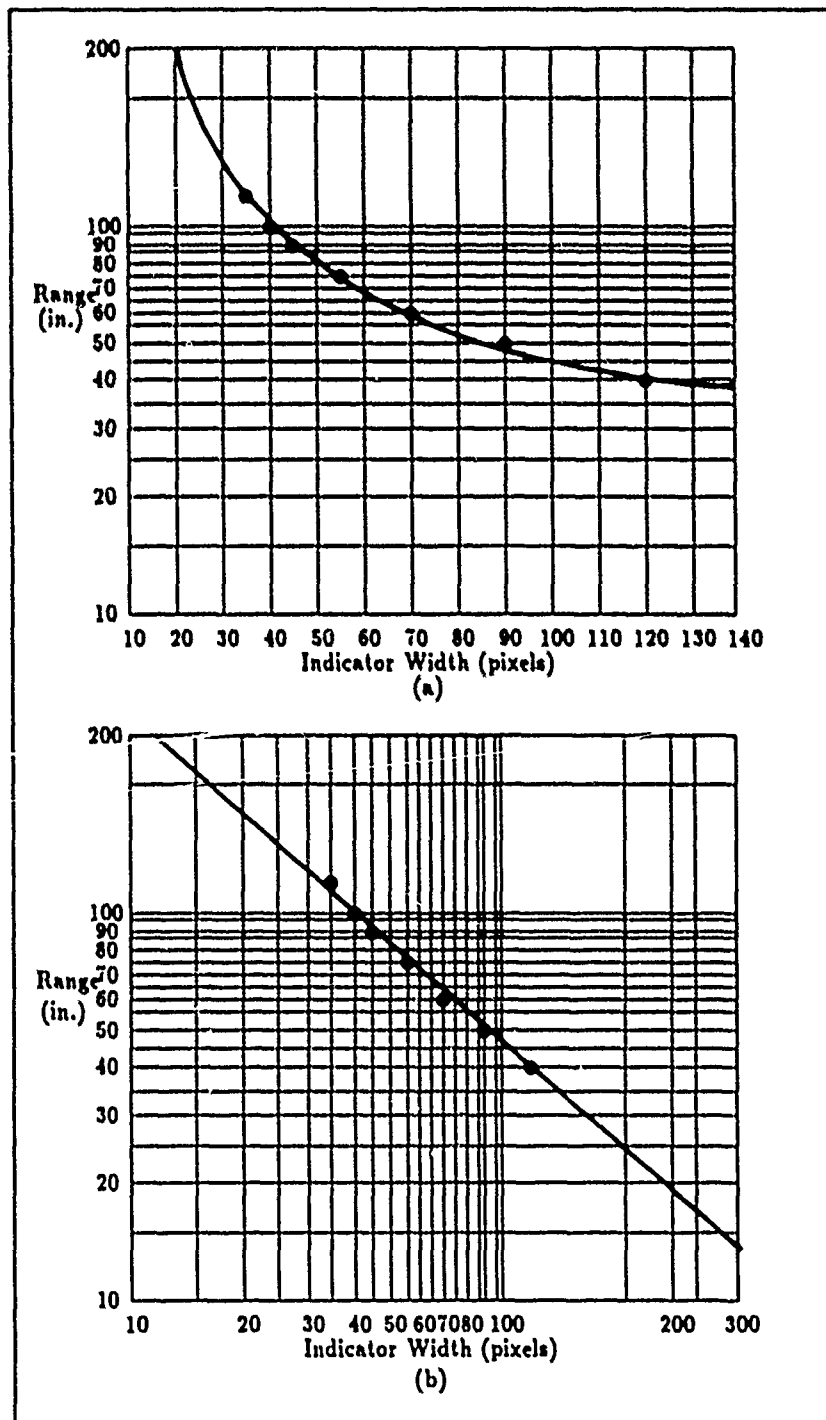


Figure 3.9. Logarithmic Plots (Indicator Width vs. Range) (a) Semilogarithmic and (b) Logarithmic

which when reduced became:

$$4.7875 = d - 3.5553c \quad (3.14)$$

and

$$3.6889 = d - 4.7622c \quad (3.15)$$

Subtracting Equation 3.15 from Equation 3.14 and solving for c , results in a c value of 0.91027. Substituting the value for c into Equation 3.14, d was found to be 8.0238. Therefore, the basic range equation becomes:

$$\ln(\text{range}) = 8.0238 - .91027 \ln(\text{width}) \quad (3.16)$$

Unfortunately, Equation 3.16 was in an unacceptable form since range was still a logarithmic function. Therefore, a range equation was produced by applying the exponential function to both sides of Equation 3.16 and simplifying:

$$\text{range} = 3052.8 / \text{width}^{.91027} \quad (3.17)$$

As used in this context, range calculations provided measurements in inches. For use in this thesis effort, range must be in meters. By multiplying the right side of Equation 3.17 by 0.0255 (inches to meters conversion factor), an equation that produced range measurements in meters was derived:

$$\text{range} = 77.846 / \text{width}^{.91027} \quad (3.18)$$

Based on this development procedure, the range equation was fairly accurate. For this research however, precise range calculations were not required since the object of the kinematics was to place the end-effector in the close proximity of the refueling receptacle's slipway, where an impedance controller would perform the actual insertion (an advantage of not using vision for the insertion task).

3.6 Modified Brightness Normalization

In an attempt to overcome the ever present illumination difficulties encountered, an image preprocessing algorithm was developed based on Capt. Lambert's brightness normalization research [6]. This preprocessing algorithm used the same neighborhood averaging

scheme employed by the Lambert's brightness normalization algorithm, but Equation 2.2 was modified by making the threshold value zero and multiplying the difference of the old pixel value and the neighborhood average by a scaling term:

$$pixelvalue = multiplier(pixelvalue - neighborhoodaverage) \quad (3.19)$$

In addition, if the value of the new pixel value was negative, then the pixel value was set to zero.

Since the image itself was not required to remain intact for recognition purposes, only the key object features needed for recognition were required in the image. Therefore, the preprocessed image was thresholded about zero (black) and only the edges of the objects found within the image would be displayed. The scaling multiplier was required to produce a spread of pixel intensities to better distinguish the edges of one object from another in the image.

To eliminate unwanted isolated patches of displayed pixels, an isolated pixel(s) removal technique was implemented. By passing a square mask of varying size over the entire image and ANDing the mask and current square, these patches could be eliminated. The size of the masks ranged from 3 x 3 to 10 x 10 pixel arrays. If all of the pixels on the mask's edges were zero (black), then all the pixel values in that square were changed to zero. The goal of this masking was to eliminate all isolated pixel areas leaving only the edges of continuous pixels in the image. This continuous pixel string would be the indicators or other objects which have a continuous connection of pixels enclosing an area. With the three indicators identified in the image, the refueling receptacle could be recognized using the algorithm discussed previously in the unstructured environment. Figure 3.10 provides an example of a 5 x 5 mask applied to the image.

3.7 Summary

This chapter detailed the development of the recognition algorithms used for recognition of the refueling receptacle, the range calculations, and receptacle offset parameters calculations. In addition, two different environmental algorithms were developed. The first for a white object on a black background (structured environment), and the second for the

0	0	0	0	0
0	X	X	X	0
0	X	X	X	0
0	X	X	X	0
0	0	0	0	0

Figure 3.10. Example Isolation Mask (5 x 5)

real-world unstructured environment using indicators. Finally, initial development of an algorithm using a variant of Lambert's brightness normalization was presented.

Now that the refueling receptacle can be identified and located (i.e., range and offsets), the next step is to discuss the search and kinematics algorithms used to visually servo the PUMA.

IV. Visual Servoing

4.1 Introduction

The visual servoing aspect for this thesis effort was divided into four separate areas:

- communications between the VAXstation III and the PUMA-560,
- development of the search algorithm,
- derivation and development of the forward and inverse kinematics, and
- integration of vision feedback and PUMA motion control.

All four areas will be discussed in turn.

4.2 PUMA Communications

Prior to any other task being accomplished, verification of communications between the PUMA-560 and a VAXstation III (suitably called "CYCLOP") computer, which executed the application software, was required. This communications path was designed to be an open-loop, static look-and-move system, and was serial with a baud rate of 19200. Previously, all communications software had been implemented and/or called by FORTRAN subroutines.

The VAL II operating system originally supplied with the PUMA was previously replaced by a custom designed environment, the AFIT Robotic Algorithm Development and Evaluation Environment (ARCADE) [3], for control algorithm evaluation. The inhouse system retained the original PUMA microprocessor structure and is thus able to issue VAL motion primitives. Since the goal in this case was positioning rather than low-level control research, all motion was controlled through the original PUMA servo loop. In order to use these primitives, the connection between the host computer and the PUMA must be serial. Additional information regarding PUMA level programming can be found in RAL 67 [20].

Since the use of C appears to be the robotics language of choice and all the vision software supporting the ITEX board was written in C, C was chosen for the visual servoing

programming language. Rather than rehost the FORTRAN communications subroutines, this thesis effort was the first attempt for calling these functions from a C main program. For communications test purposes, a simple C program was written (utilizing these FORTRAN subroutines) which allowed the user to automatically calibrate the PUMA, open a communications path between the PUMA and "CYCLOP", select from one to six joints to move and the new joint position(s) to move to, and upon request, terminate communications with the PUMA. All the code and user's manual for this communications test code can be found in internal report ASRL-89-14 [7] in the ASRL at AFIT.

4.3 Search Pattern

In order to efficiently locate the refueling receptacle, a reasonable search pattern had to be developed that would make the best use of the PUMA's motion abilities. Naturally a progressively inward spiral search pattern would appear to be best, but the PUMA, due to its rotational motion, is more adept for vertical search and move patterns with horizontal incrementation. Therefore, this type of search pattern was implemented with an initial search position, as shown in Figure 4.1, starting at the current joint one angle, and joints two and three placed at -90° and -35° , respectively. The search pattern consisted of sequential movements of joints one and three performing the motion shown in Figure 4.2.

At each search position, an image was acquired by the ITEX imaging board and recognition algorithms attempted to ascertain if the refueling receptacle was in the field of view (FOV). If the proper image was not found, then the PUMA was moved to its next search position and another image was acquired. If the refueling receptacle was found in the image, then the range and offsets parameters were determined. If the refueling receptacle was in range, then the search was stopped, the kinematics were performed, and the PUMA was directed to move towards the refueling receptacle. If the refueling receptacle was lost while the PUMA was in motion towards it, then the PUMA would return to the initial search position (at the current joint one angle) and the search would continue in a normal manner. If the refueling receptacle was found but was out of range of the PUMA's reach, then joints one and three were rotated slightly to maintain the refueling receptacle in the center of image and the search was stopped. On the other hand, if the refueling receptacle

was out of range and then became lost, the PUMA would again return to its initial search position and continue with the search. Figure 4.3 provides a logic diagram for this process.

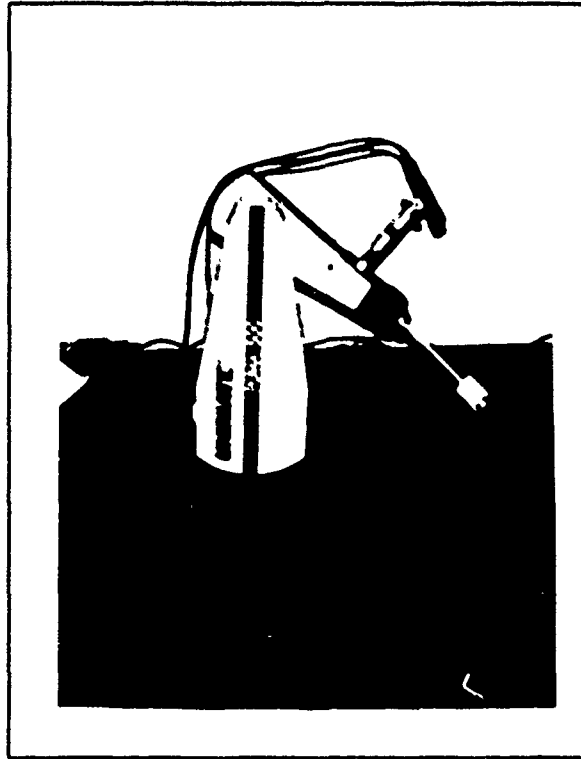


Figure 4.1. Initial Search Position ($\theta_2 = -90^\circ$, $\theta_3 = -35^\circ$)

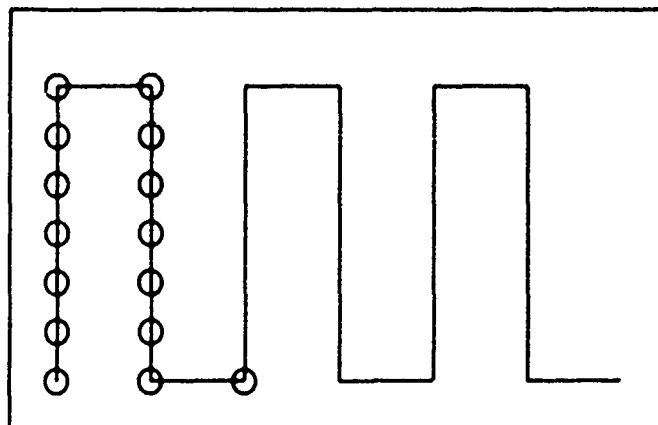


Figure 4.2. Search Pattern (circles represent image acquisition points)

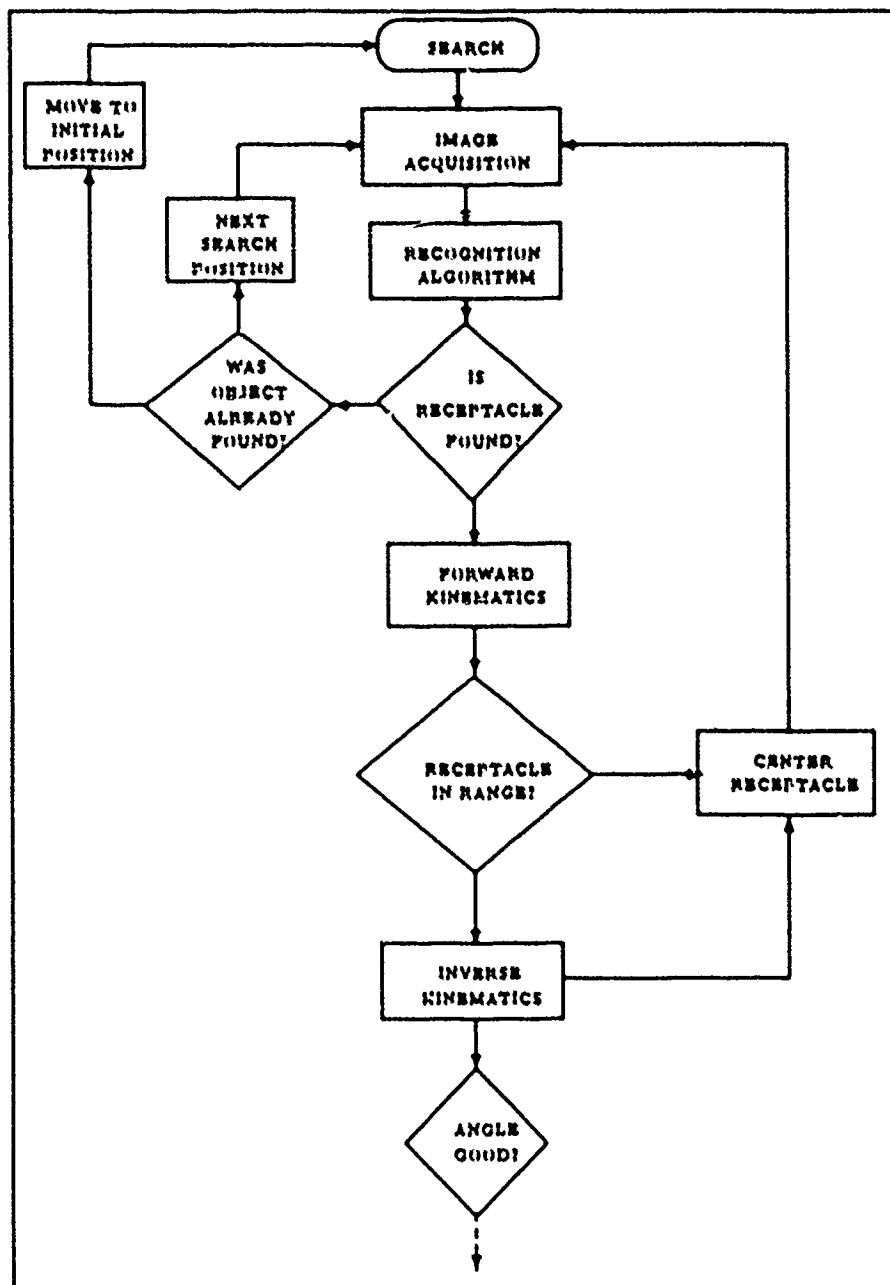


Figure 4.3. Search Logic Diagram

In order to ensure that during the search phase the receptacle was not "overlooked" (i.e., partially in the image, skipped entirely from one search position to the next), incremental search positions for the search pattern sufficiently overlapped each other. The lowest search angle was derived based on the fact that the closest position that the refueling receptacle could come to the robot was when the refueling receptacle test stand was butted up against the PUMA base. In this case, the refueling receptacle was in the camera's FOV when joint three was at -35° . The highest search angle (0°) was derived from the fact that when the PUMA was placed in this position, the camera was in a look-down position capable of maximum desired scene depth. Based on these two extremes and numerous tests, 7° increments were found to provide sufficient vertical overlap of search areas. Rotational increments of 3° provided sufficient horizontal overlapping of the search area to prevent "overlooking."

4.4 Kinematics

Kinematics were required to determine the world position of the refueling receptacle, in cartesian coordinates, and then transform them into the PUMA joint angles required for moving the end-effector (refueling nozzle) to the receptacle when found. The camera was mounted on link 3 of the PUMA to simulate that of a real refueling boom, which would have the camera mounted near the end-effector, if not on the last link. In the case of the PUMA, this end-point was considered at the end of link 3 with links 4-6 maintained rigid and aligned along the direction of link 3. The refueling nozzle was considered an extension of link 3. Since only 3-DOF motion was required, not the full 6-DOF capability of the PUMA, kinematics complexity was reduced. Joint four motion was only used to place the refueling nozzle off to the side during image acquisition, and as a safety feature in avoiding actual contact with the refueling receptacle test stand, as shown in Figure 4.4.

4.4.1 Forward Kinematics The forward kinematics were based on the 3-DOF position equation developed by Fu et al. for the first three links of the PUMA, and provided the cartesian position of the receptacle relative to the base of the PUMA (considered the origin of the world coordinate system) [13]. Figure 4.5 illustrates a representative PUMA

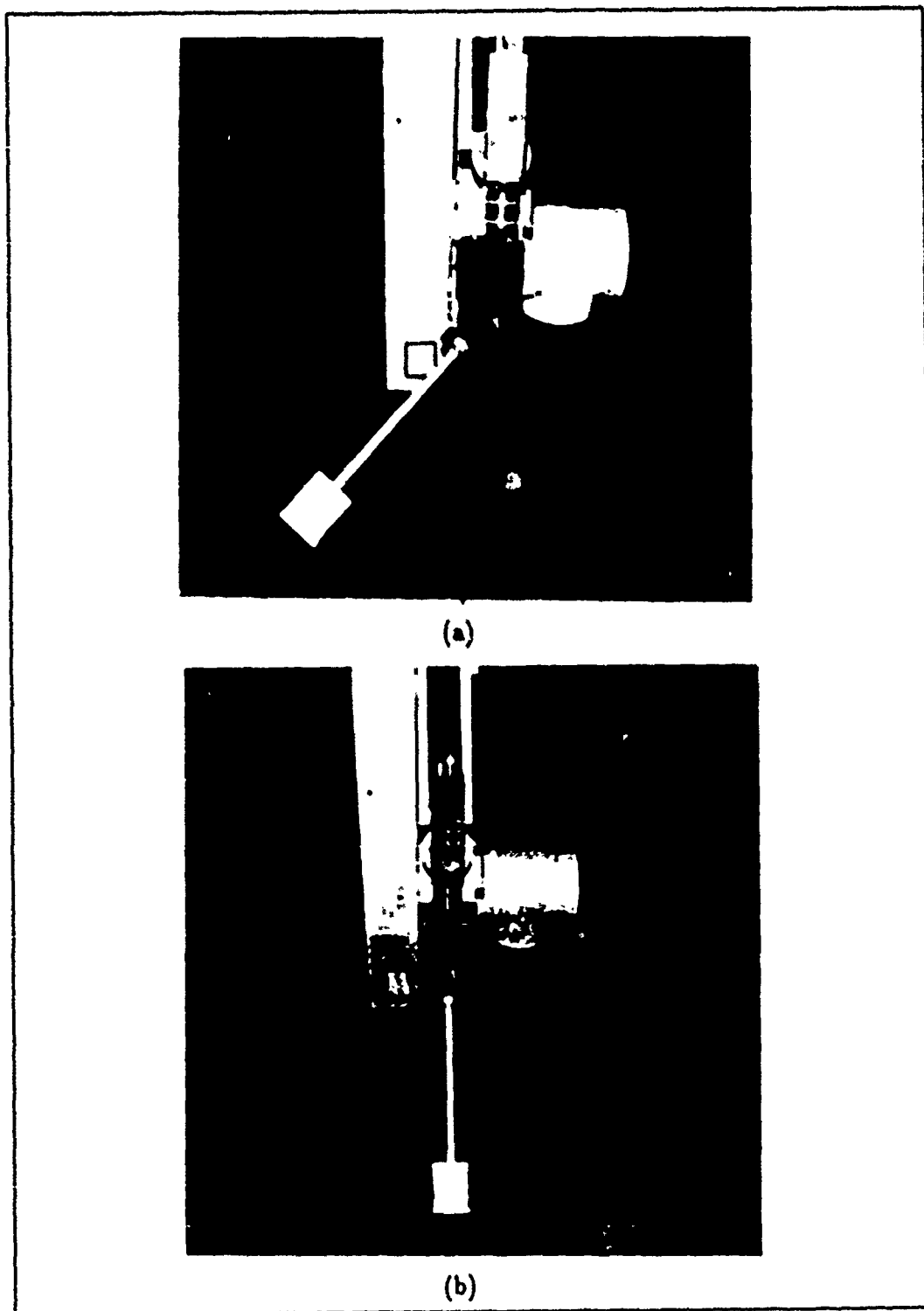


Figure 4.4. Joint Four Positioning (a) Rotated to Side During Image Acquisition, and (b) Rotated Up During Movement to Receptacle

configuration in reference to the refueling receptacle.

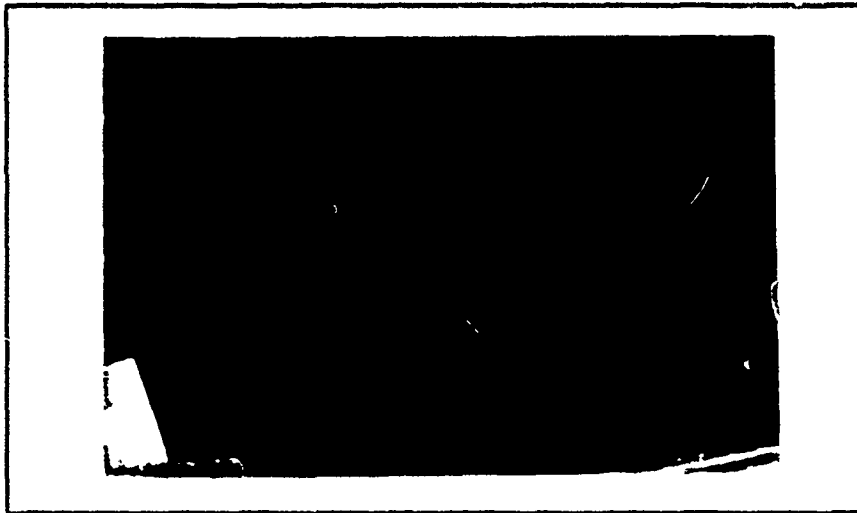


Figure 4.5. Representative PUMA Configuration

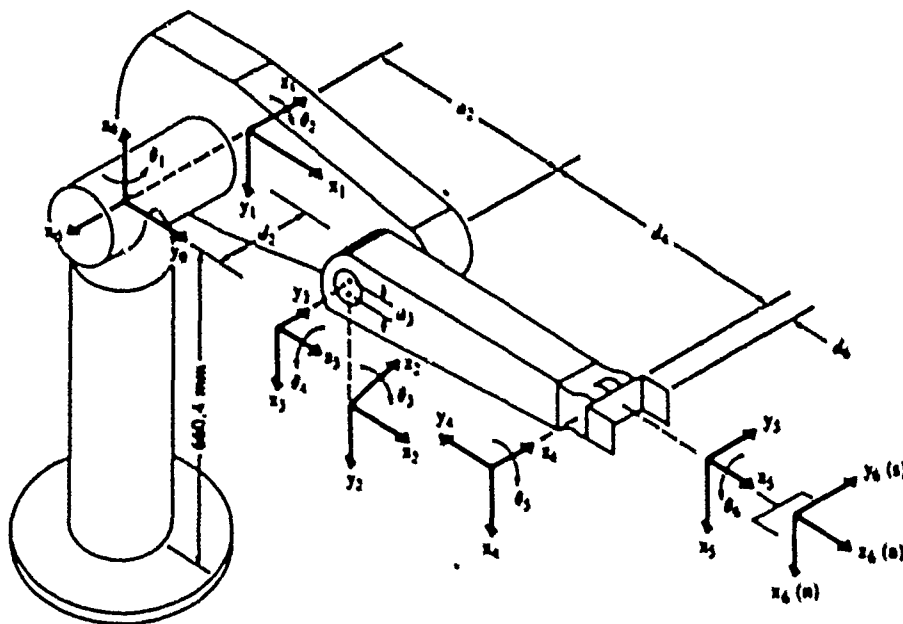
The algorithm developed for determining the position of the receptacle relative to the base is called the Psuedolink Transformation. This transformation finds the cartesian coordinates of the receptacle by making use of the link parameters d_4 and a_2 associated with links 2 and 3 of the PUMA as defined by Denavit-Hartenburg representation which states:

"To describe the translational and rotational relationship between adjacent links ... a matrix method of systematically establishing coordinate systems (body-attached frame) to each link of an articulated chain. This ... representation results in a 4×4 homogeneous transformation matrix representing each link's coordinate system at the joint with respect to the previous link's coordinate system. Thus, through sequential transformations, the end-effector expressed in the 'hand coordinates' can be transformed and expressed in the 'base coordinates' which make up the frame of this dynamic system [13:36]."

Figure 4.6 provides an illustration of the Denavit-Hartenburg representation for the PUMA. Based on this figure, d_4 and a_2 were considered the lengths of link 2, and links 3, 4, 5, and 6, respectively.

The Psuedolink Transformation transformed the link distances, d_4 and a_2 , into parallel but modified lengths, d'_4 and a'_2 , which place the end-effector at the receptacle's

- θ_i is the joint angle from the x_{i-1} axis to the x_i axis about the z_{i-1} axis (using the right-hand rule).
- d_i is the distance from the origin of the $(i-1)$ th coordinate frame to the intersection of the z_{i-1} axis with the x_i axis along the z_{i-1} axis.
- a_i is the offset distance from the intersection of the z_{i-1} axis with the x_i axis to the origin of the i th frame along the x_i axis (or the shortest distance between the z_{i-1} and z_i axes).
- α_i is the offset angle from the z_{i-1} axis to the z_i axis about the x_i axis (using the right-hand rule).



PUMA robot arm link coordinate parameters					
Joint i	θ_i	α_i	a_i	d_i	Joint range
1	90	-90	0	0	-160 to +160
2	θ	0	431.8 mm	149.09 mm	-225 to 45
3	90	90	-20.32 mm	0	-45 to 225
4	0	-90	0	433.07 mm	-110 to 170
5	0	90	0	0	-100 to 100
6	0	0	0	56.25 mm	-266 to 266

Figure 4.6. Denavit-Hartenburg Representation of the PUMA [13:37]

calculated position, but without motion (i.e., links 2 and 3 were lengthened or reduced as required to place the end-effector at the receptacle while maintaining the same angles of joints one, two and three). In this way, the PUMA acted like a 3-DOF manipulator and the position equation (developed by Fu et al. for the first three links PUMA [13:63]) was utilized to find the cartesian position of the receptacle when using these new lengths along with the current angle values for joints one, two, and three:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} C1(a_2 * C2 + a_3 * C23 + d_4 * S23) - d_2 * S1 \\ S1(a_2 * C2 + a_3 * C23 + d_4 * S23) + d_2 * C1 \\ d_4 * C23 - a_3 * S23 - a_2 * S2 \end{bmatrix} \quad (4.1)$$

where

$$C1 = \cos \theta_1$$

$$C2 = \cos \theta_2$$

$$C23 = \cos \theta_2 + \theta_3$$

$$S1 = \sin \theta_1$$

$$S2 = \sin \theta_2$$

$$S23 = \sin \theta_2 + \theta_3$$

A note should be added about the use of the parameter a_3 . a_3 is the distance from the end of link 2 to the center of joint three. In effect, links 2 and 3 cannot be considered, by themselves, as an ideal two-link manipulator. Due to the magnitude of this parameter and the fact that the receptacle's calculated position did not have to be that accurate, this value was still used for calculating position, but links 2 and 3 were considered to have a common rotation point for deriving the Pseudolink Transformation. Figure 4.7 illustrates a graphical representation of this transformation.

A physical requirement for this transformation was that the camera, the tip of the refueling nozzle when extended forward, and the center of the image must lie on a straight line. This requirement was necessary since when performing the transformation, the center of the image was considered the placement point of the end-effector when moving towards the receptacle. An additional benefit of this requirement was that it fixed the camera

geometry in respect to the PUMA, thereby deleting the requirement of a formal transformation from the camera space to the robot space, since the angle ϕ_0 (which is the camera to robot transformation) is determined only by the height of the camera above link 3. In addition, the PUMA was always maintained in a right and above arm configuration, as defined by Fu et al. [13:56], to avoid any singularity occurrences.

Using the Psuedolink Transformation concept, both d_4 and a_2 were transformed into lengths that placed the end-effector at the receptacle position while maintaining the angles for joints one, two, and three. First, a_2 was transformed into a'_2 as follows:

$$a'_2 = a_2 - F \quad (4.2)$$

The F term was found by first determining the distance from the receptacle to the end-effector, $portnoz$:

$$portnoz = range - camnoz \quad (4.3)$$

where $camnoz$ is the distance from the camera to the tip of the extended end-effector.

Once the distance $portnoz$ was found, the angle of approach, that is the angle that the receptacle position and end-effector position make between d_4 and d'_4 , was determined. If the receptacle was centered in the image, then this value would be ϕ_0 . ϕ_0 was the angle that the camera made with its horizontal component (parallel with d_4). If the receptacle had a vertical offset from the center of the image, then that offset distance, $deltaz$, was used to determine the new angle of approach, ϕ_2 :

$$\phi_2 = \phi_0 - \phi_1 \quad (4.4)$$

ϕ_1 was found by the geometric relationship of right angles and was determined as:

$$\phi_1 = \arctan(deltaz/portnoz) \quad (4.5)$$

This change in approach angles also changed the actual distance from the end-effector to the center of the refueling receptacle, thereby creating a new distance, $portnozzle$, as shown:

$$portnozzle = \sqrt{(portnoz^2 + deltaz^2)} \quad (4.6)$$

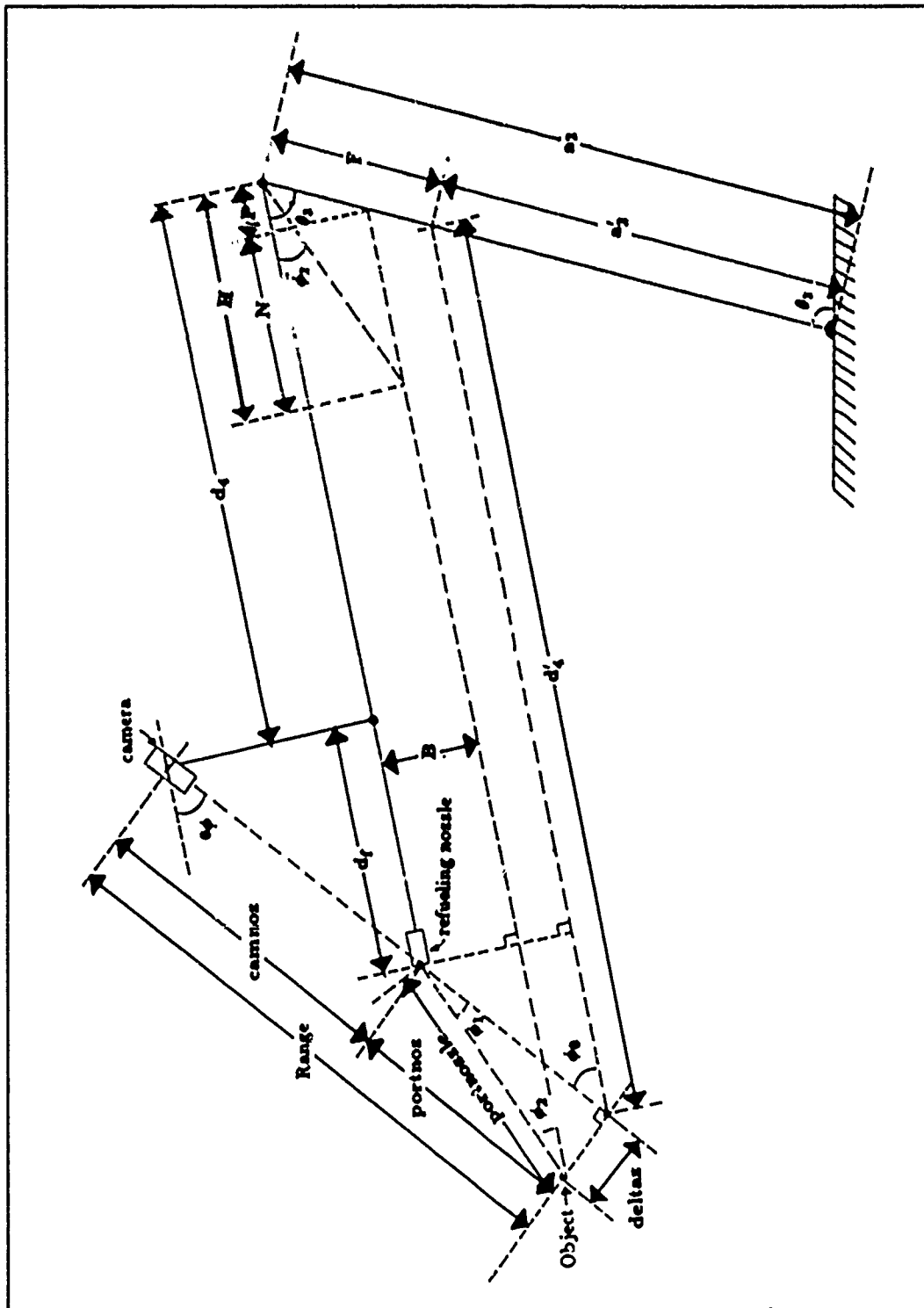


Figure 4.7. Geometric View of Psuedolink Transformation

Based on this distance, the separation distance between d_4 and d'_4 , B , was found:

$$B = portnozzle * \sin(\phi_2) \quad (4.7)$$

and therefore, F was found through the relationship:

$$F = B / \sin(\theta_2) \quad (4.8)$$

With the appropriate substitutions, F became:

$$F = portnozzle * \sin(\phi_2) / \sin(\theta_2) \quad (4.9)$$

and a'_2 became:

$$a'_2 = a_2 - portnozzle * \sin(\phi_2) / \sin(\theta_2) \quad (4.10)$$

With a'_2 derived, d'_4 was determined by the following equation:

$$d'_4 = d_4 + d_f + N \quad (4.11)$$

where d_f was the distance from the end of link 3 to the end of the refueling nozzle, and N was the delta length of the transformation. Both d_4 and d_f were known values, therefore only N had to be determined. N was based on the relationship:

$$N = H - P \quad (4.12)$$

where P was represented simply as:

$$P = F * \cos(\theta_2) \quad (4.13)$$

By making the appropriate substitution for F , P became:

$$P = portnozzle * \sin(\phi_2) / \tan(\theta_2) \quad (4.14)$$

and H was expressed simply as:

$$H = portnozzle * \cos(\phi_2) \quad (4.15)$$

By making the appropriate substitutions for H and P , N became:

$$N = portnozzle * (\cos(\phi_2) - \sin(\phi_2) / \tan(\theta_2)) \quad (4.16)$$

Since link 1 was actually a rotational joint (it has no link length), this separation was considered valid. Joint one rotation provides the centering of the receptacle in the image as well as aligning the end-effector with the receptacle, and joints two and three provided the planar motion required to place the aligned end-effector at the refueling receptacle. As with the forward kinematics, the inverse kinematics were also implemented in C.

4.4.2.1 Joint One Figure 4.8 provides a side view of the representative PUMA configuration used to develop the forward kinematics. From this figure, the horizontal component of the distance from the base to the end-effector, *outdis*, was determined as:

$$\text{outdis} = d'_4 * \cos(\theta_1 - \theta_2) - a'_2 * \cos(\theta_1) \quad (4.19)$$

From the PUMA, this distance represented the horizontal component of the distance from the origin of the world coordinate system to the location of the refueling receptacle. This distance was located in the X-Y plane of the world coordinate system.

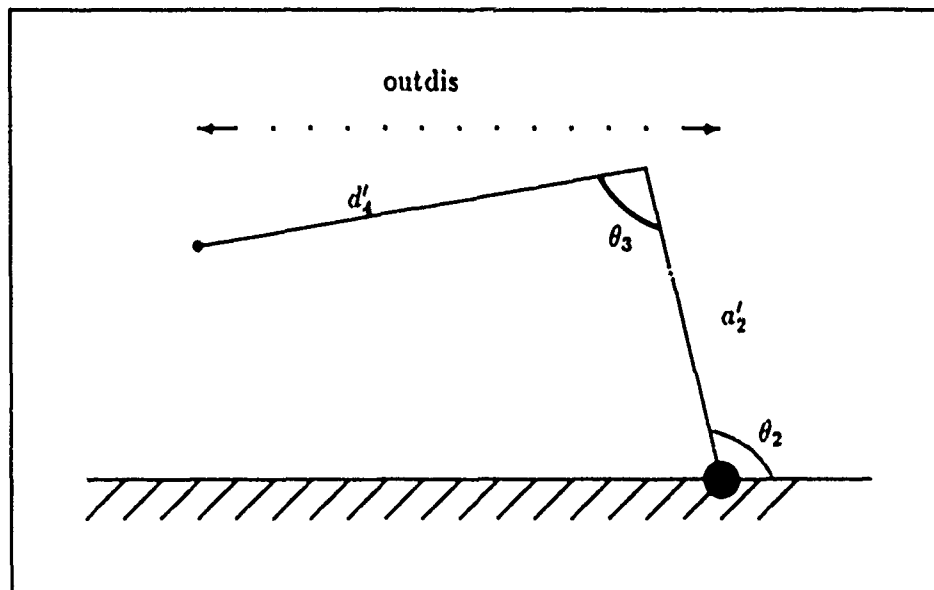


Figure 4.8. Side View of Representative PUMA Configuration (shown as two-link system)

A top view of the representative PUMA configuration is shown in Figure 4.9. If the refueling receptacle was identified and located a perpendicular distance *G* from the

centerline, then the rotation increment required of joint one to align the end-effector with the receptacle was determined by the following equation:

$$\alpha_2 = \alpha_0 - \alpha_1 \quad (4.20)$$

where α_0 was the angle between the centerline and the location of the receptacle, and α_1 was the angle between the centerline and the end-effector.

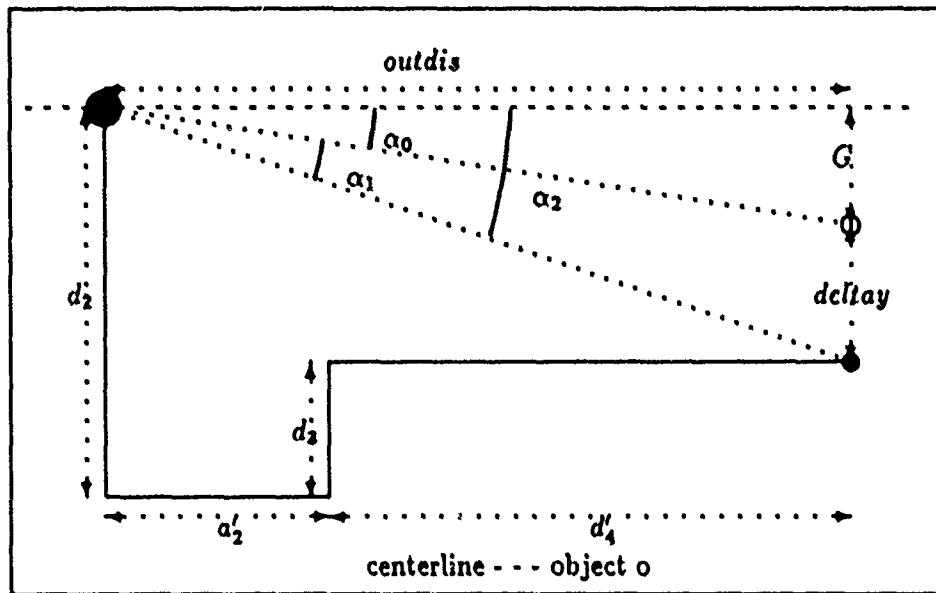


Figure 4.9. Top View of the Representative PUMA Configuration

The equations for α_0 and α_1 were simply:

$$\alpha_0 = \arctan(G/outdis) \quad (4.21)$$

and

$$\alpha_1 = \arctan((d_3 - d_2)/outdis) \quad (4.22)$$

respectively. Substituting Equation 4.21 and Equation 4.22 into Equation 4.20 yielded:

$$\alpha_2 = \arctan(G/outdis) - \arctan((d_3 - d_2)/outdis) \quad (4.23)$$

Therefore, joint one's value, θ_1 was changed by an incremental amount, α_2 :

$$\theta_1 = \theta_1 + \alpha_2 \quad (4.24)$$

to align the end-effector with the refueling receptacle.

4.4.2.2 *Joints Two and Three* The inverse kinematics for joints two and three were derived by considering links 2 and 3 as a two-link planar manipulator and knowing the location of the end-effector from the base of the manipulator (the origin of the PUMA). In addition, since the lengths of the sides of a triangle were known (link 2, link 3, and the distance from the origin to the end-effector), the Law of Cosines was used to determine the three unknown angles within that triangle, and those angles were subsequently used to determine joint angles two and three, θ_2 and θ_3 , respectively. Figure 4.10 provides a side view of a two-link manipulator where the refueling receptacle is a distance, position, away, based on the position coordinates derived in the forward kinematics.

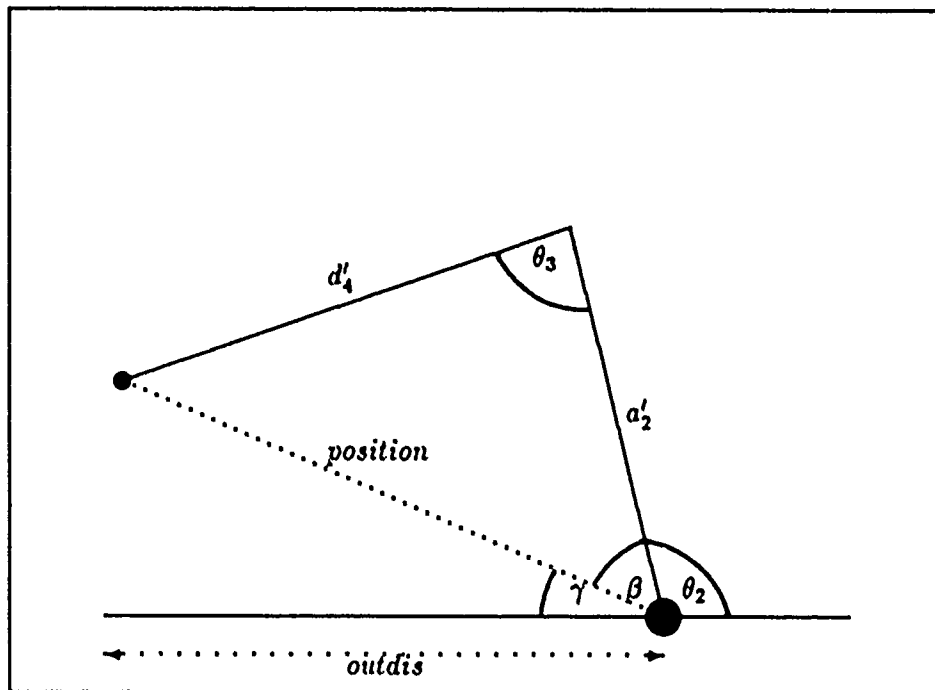


Figure 4.10. Side View of a Two-Link Manipulator

θ_3 was determined by substituting the appropriate side lengths into the Law of Cosines equation producing:

$$position^2 = a_2'^2 + d_4'^2 - 2 * a_2' * d_4' * \cos(\theta_3) \quad (4.25)$$

and solving for θ_3 ,

$$\theta_3 = \arccos((position^2 - a_2'^2 - d_4'^2)/(2 * a_2' * d_4')) \quad (4.26)$$

θ_2 was determined from the angular equation:

$$\theta_2 = 180 - \beta - \gamma \quad (4.27)$$

β was determined by applying the Law of Cosines:

$$\beta = \arccos((d_4'^2 - a_2'^2 - position^2)/(2 * a_2' * position)) \quad (4.28)$$

and γ was determined simply:

$$\gamma = \arctan(P_z / \sqrt{(P_x^2 + P_y^2)}) \quad (4.29)$$

Substituting Equation 4.28 and Equation 4.29 into Equation 4.27, θ_2 became:

$$\begin{aligned} \theta_2 = 180 - \arccos((d_4'^2 - a_2'^2 - position^2)/(2 * a_2' * position)) - \\ \arctan(P_z / \sqrt{(P_x^2 + P_y^2)}) \end{aligned} \quad (4.30)$$

With θ_1 , θ_2 and θ_3 found, the end-effector could now be placed at the location of the refueling receptacle anywhere within the actual limits of the PUMA's area of motion.

4.4.3 Self-Centering If the refueling receptacle was identified, but its location was outside the reach of the manipulator, then joints one and three were incremented slightly, as required, so that the receptacle was centered in the image plane. Therefore, as the receptacle moved in any direction, it was maintained in the camera's FOV. This self-centering task was performed quite simply. Joint one was servoed using Equation 4.23. Joint one's movement centered the receptacle in the horizontal plane, whereas, centering of the receptacle in the vertical plane was based on moving joint three an angular increment of ϕ_1 (derived in Equation 4.5).

4.5 Dynamic Servoing

In order to perform dynamic servoing, parallel processing of the vision and PUMA motion control algorithms was required. Using a Vaccelerator coprocessor board, as described in Appendix A, located within "CYCLOP" and utilizing the Vaccelerator's ability to create shared memory (general memory whose location is known and accessible by

more than one process), the initial parallel processing architecture for dynamic look-and-move was developed. The dynamic look-and-move servoing system separated the vision and PUMA motion control algorithms, and placed them on the VAXstation III's and the Vaccelerator board's CPU, respectively, allowing the increased processing speed of the Vaccelerator board to reduce the massive analysis and computational time required by the vision algorithms.

As stated previously, shared memory was created by the Vaccelerator CPU and initialized to zero. Shared memory was the preferred method for data transfer, because pipelining, message transfer and file transfer involve complicated communications protocol and are time consuming (i.e., disk transfer time). Shared memory on the other hand required very little communications protocol and transfer time was dependent only on the speed of memory access for the system. Although, for the shared memory technique, a software semaphore system (utilized by most modern operating systems) was implemented to control entry into and ensure data integrity within the shared memory. A semaphore system allows only one process to access (either to read or write data) the memory, while the other process, if requesting access, must wait. Access is granted when the other process leaves the semaphore system. The shared memory was set-up as a twenty element array holding new and current PUMA joint angle data, error code, task code, and current shared memory status (is this new or old data?).

All the search, recognition, and kinematics algorithms developed for the static look-and-move remained unchanged, the only change in software was the addition of data transfer modules (written in C and FORTRAN) within the software structure of both processes. In addition, the searching portion of this dynamic system remained static look-and-move motion, only the tracking and forward motion of the PUMA was converted to dynamic look-and-move, thereby allowing simpler testing of this system.

4.6 Summary

Within this chapter, the visual servoing algorithms were presented which deals with movement of the PUMA for searching, and for motion towards the refueling receptacle. The task used to reduce the kinematic complexity of the forward kinematics was based on

the Psuedolink Transformation concept. The inverse kinematics were derived in two separate calculations: joint one's required rotation angle for receptacle/end-effector alignment, and joint's two and three determination based on the Law of Cosines relationship. In addition, the framework for a dynamic look-and-move system was described. Now that the recognition and visual servoing algorithms have been presented, the next chapter discusses the results obtained from these algorithms.

V. Results and Discussion

5.1 Introduction

The primary goal of this thesis effort was to develop and demonstrate a robotic visual servoing system (RVSS) for autonomous aircraft refueling. This task was accomplished by combining current vision technology with existing AFIT robot control systems and algorithms. Development of this RVSS was divided into two separate tasks:

- refueling receptacle recognition and,
- visual servoing of the PUMA.

For each task, extensive testing was performed attempting to produce an effective and efficient system. Based on the development scheme used for this thesis effort, results are presented and discussed dealing with the following distinct areas:

For the refueling receptacle recognition task,

- (a) the success of the object recognition system, and
- (b) the success of the modified brightness normalization algorithm.

And for the visual servoing task,

- (a) the success of the search algorithm,
- (b) the success of static look-and-move servoing, and
- (c) the initial evaluation of dynamic look-and-move servoing.

5.2 Refueling Receptacle Recognition

Chapter 3 provides a detailed description of how the recognition algorithm operates. Only the results dealing with the unstructured environment (real-world) recognition system are presented, since the black and white research was developed for search and kinematics algorithm verification only. In addition, special tests using a modified version of Lambert's brightness normalization algorithm were performed to determine its benefits for image preprocessing for variable scene illumination.

5.2.1 Object Recognition This section of the results discusses the reliability and efficiency of the recognition algorithm. Figure 5.1 provides an illustration of an identified refueling receptacle. In this figure, all three indicators have been identified and differentiated from the background portion of the image.

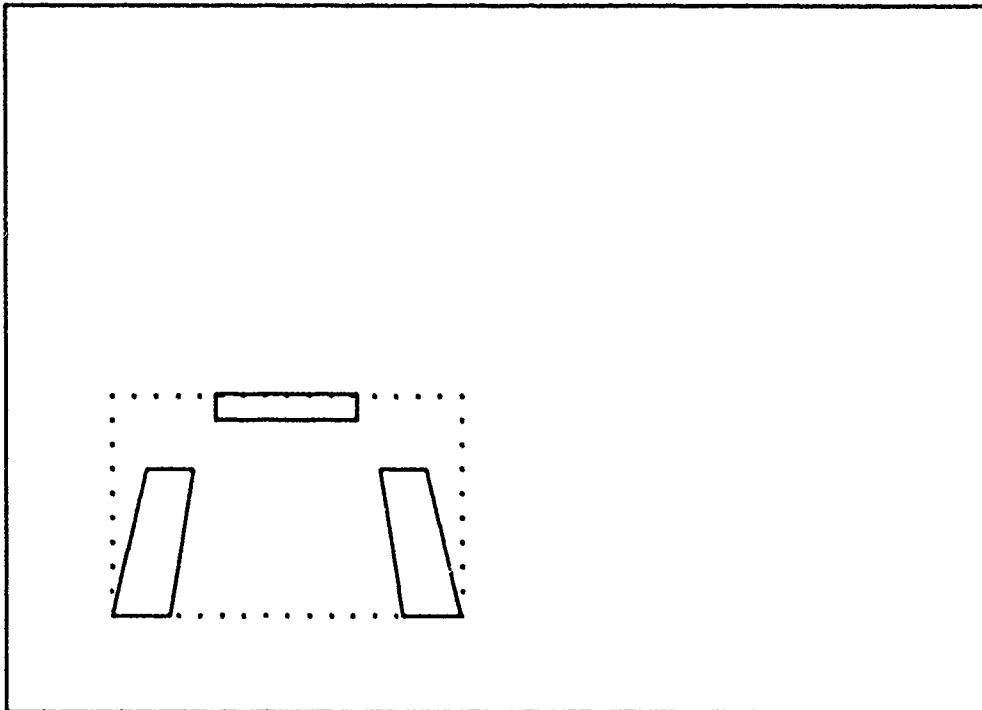


Figure 5.1. Identified Refueling Receptacle (dotted box represents identified object)

Nonrecognition of the refueling receptacle occurred under the following circumstances:

- (a) refueling receptacle not in the image,
- (b) missing indicator from image,
- (c) indicator(s) partially in the image (indicator lies on edge), and
- (d) only partial or no indicator(s) displayed in the image.

Case (a) was expected, if the receptacle was not in the camera's field-of-view (FOV), then recognition of a nonexistent entity was impossible. Case (b) was also expected, since the recognition algorithm required all three indicators to be present for recognition purposes.

Encountering this case meant that all the indicators would appear in subsequent images. Figure 5.2 provides an example of this case.

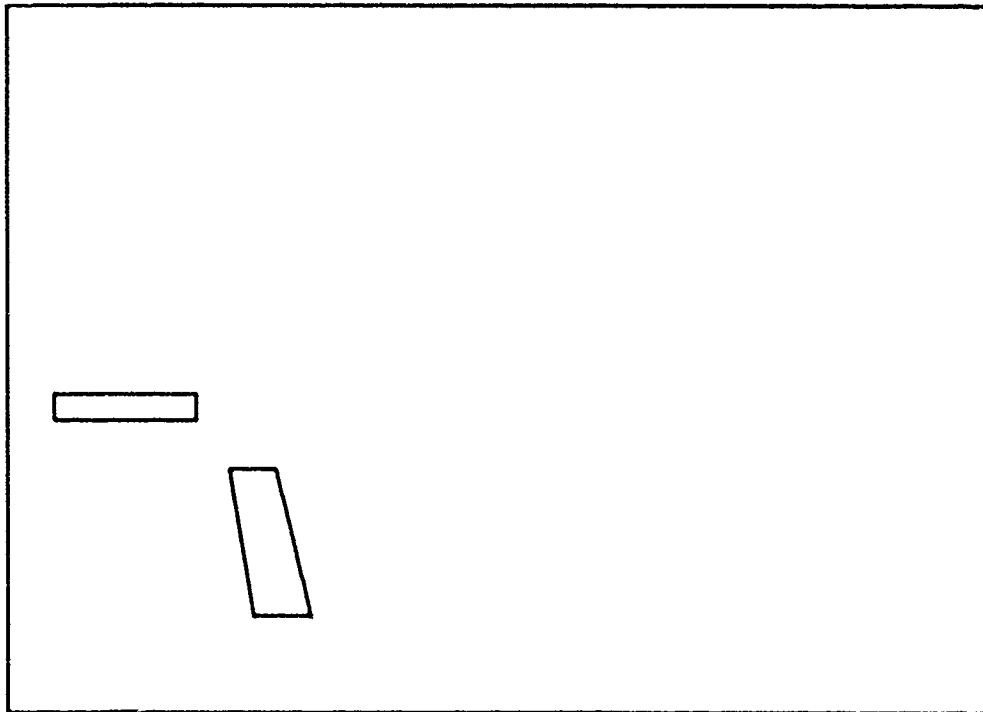


Figure 5.2. Image Missing Indicator(s)

Case (c) was also expected, but dependent on the size of the displayed indicator(s) in the image, recognition may or may not occur during this search area analysis. This irregularity occurred due to the criteria used for recognition developed in Chapter 3. For example, if the top indicator was partially hidden at the top of the image, as shown in Figure 5.3, then the refueling receptacle would be identified because the height of the top indicator was not one of the criteria used in receptacle recognition. On the other hand, if the two bottom indicators were partially hidden at the bottom of the image, as shown in Figure 5.4, then the refueling receptacle would not be recognized because the height of the bottom two indicators must be greater than the height of the top indicator. If the receptacle is not identified due to this problem, recognition will occur in subsequent images.

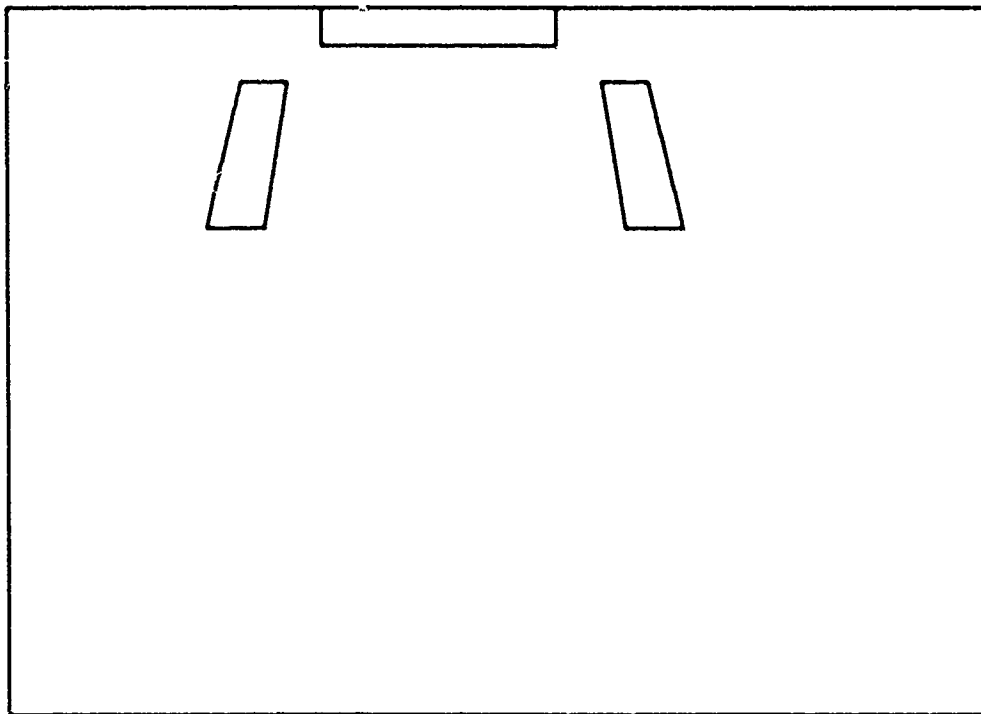


Figure 5.3. Top Indicator Partially Hidden

Case (d) on the other hand was not expected and degraded the reliability of the recognition system. This occurrence was caused for two different reasons. The first occurrence was caused by the dynamic threshold determination algorithm. This threshold is based on the pixel value of the brightest peak of the image histogram. In some instances, some indicator pixels may fall below this threshold, thereby making these pixels appear as part of the background instead of as part of the object, as shown in Figure 5.5. This seldom happened when lighting was uniform upon the refueling receptacle test stand. The second occurrence was caused due to nonstructured lighting effects. Nonrecognition in this case was caused by lighting effects such as glare, shadows, illumination unevenness, etc., in which case the dynamic thresholding algorithm would produce an erroneous threshold value. In both cases, however, recognition would not occur as required, thereby decreasing the reliability of the recognition system.

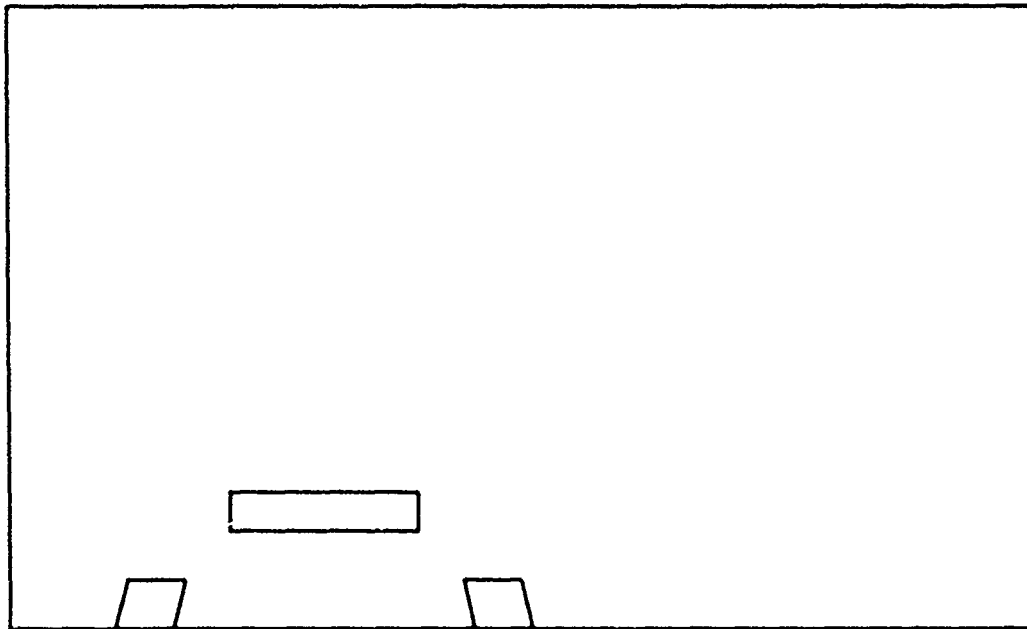


Figure 5.4. Bottom Indicators Partially Hidden

5.2.2 Modified Brightness Normalization Preprocessing In order to compensate for the effects of lighting on the receptacle, which caused recognition problems as shown in case (d) above, testing was performed using a variant of Lambert's brightness normalization algorithm (as described in Chapter 3). After the algorithm was applied to an image containing the three indicators, the edges of the indicators were present, but all other edges were present as well. Unfortunately, applying the isolation mask to the preprocessed image failed to eliminate all patches of isolated pixels. Further research in this area proved fruitless.

System efficiency for the static look-and-move RVSS was poor. Overall area search was quite long, since the PUMA had to analyze six separate scenes for each search cycle until the receptacle was found. On the average, PUMA search motion to image acquisition to PUMA motion was about four seconds. Most of this time was spent analyzing the scene, especially creating an image histogram. All other calculations were computed quite fast (under the one second time frame). Since the search pattern provided scene overlapping during the search phase, acquiring one scene which possessed entirely all three receptacle's

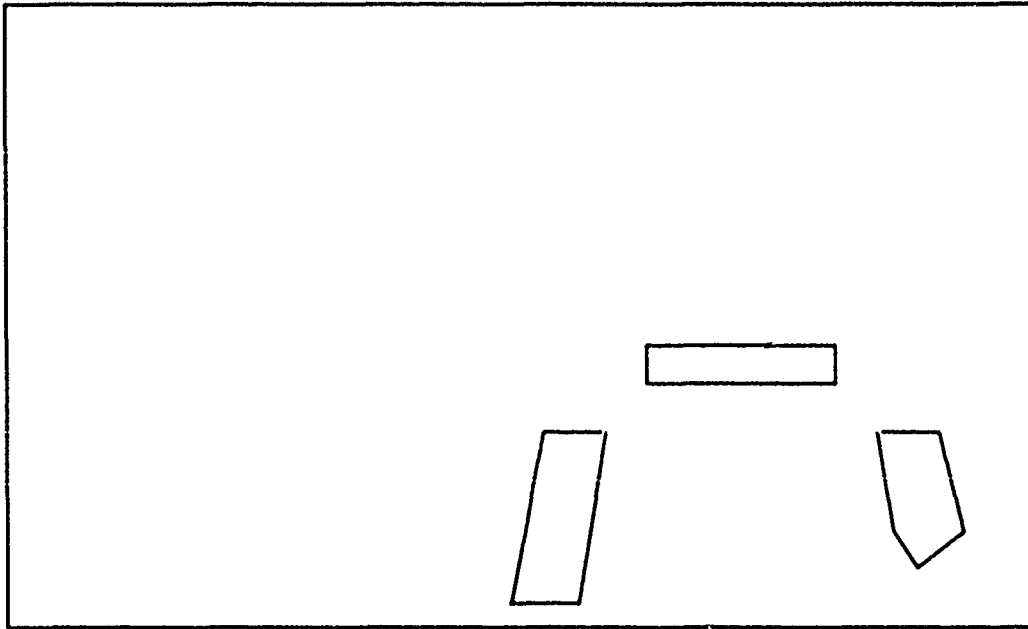


Figure 5.5. Partial Detection of Receptacle Indicators

indicators was always met.

5.3 Visual Servoing

Chapter 4 provides a detailed description on how the search, and servoing algorithms operate. For this thesis effort, only the static look-and-move RVSS algorithms were actually demonstrated due to communications problems between CPUs during dynamic look-and-move testing.

5.3.1 Search After extensive testing, a search pattern with enough overlap to ensure receptacle recognition was developed. Unfortunately, one complete search cycle consisted of six separate search areas. This requirement increased the time required to search the overall world space. If one or all three indicators were in the image, but some portion of an indicator was missing due to the FOV of the camera at the time, the RVSS had no way of determining if these identified objects were part of the refueling receptacle or just white noise. Therefore, several other search areas had to be searched before the PUMA cycled

into an area where the entire refueling receptacle (i.e., the indicators) was fully in the FOV of the camera. Overall, the search pattern proved itself to work as desired, but the search was time consuming. In addition, the search logic that was discussed in Chapter 4 also worked as expected.

5.3.2 Static Look-and-Move For the static look-and-move RVSS, the overall result was a successful visual servoing system with some recognition problems as discussed before. When the receptacle was recognized, the kinematics placed the end-effector within ± 10 cm. of the refueling receptacle, as shown in Figure 5.6.

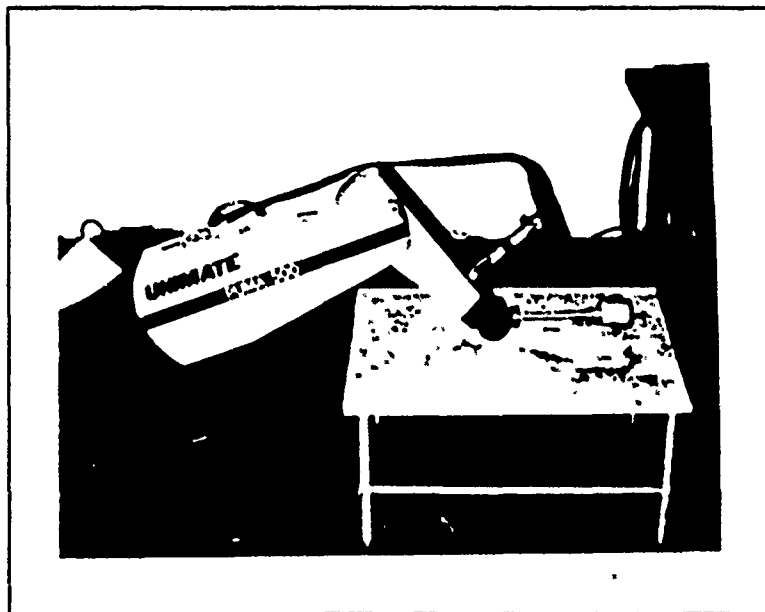


Figure 5.6. Refueling Nozzle Placement Near Refueling Receptacle

The self-centering and out-of-range tracking performed as expected, but in some cases, the receptacle was not truly centered, although still fully in the camera's FOV. This error was probably due to the inaccuracy of the kinematics equations (as discussed in Chapter 4), but since accuracy was not a goal this error was acceptable and was not investigated further.

Of even greater importance, the development of this static look-and-move RVSS allows further research to be performed using the baseline created by this working system.

5.3.3 Dynamic Look-and-Move Although the architecture for the dynamic look-and-move RVSS was developed, it was never fully tested, and demonstrated, due to communications difficulties between the Vaccelerator and "CYCLOP". Since the source code for the ITEX 100 function libraries was unavailable, compilation of this software on the Vaccelerator board was impossible. Therefore, based on the Vaccelerator Operator's Manual [21], all the ITEX 100 function names were placed in an access table, and linked to the Vaccelerator operating system. During testing of the vision algorithms, an "access violation" error was encountered for an undetermined reason. However, the shared memory ability of the Vaccelerator board was fully utilized for data transferral between CPUs and between multiple processes executing on "CYCLOP". In an attempt to run the modified dynamic RVSS software as two separate processes on "CYCLOP", problems occurred with data not being displayed on the terminal screen and code testing was unable to be performed. Debugging of these problems is still being actively pursued and should be overcome, but due to lack of time, no further work is being performed for this thesis effort.

An additional benefit of the shared memory and process interface research developed during this thesis effort will be realized for the Dexterous Motion Control research currently being conducted at AFIT. This research also requires the use of parallel processing which will be supported by an identical Vaccelerator board.

5.4 Summary

These results support the concept of the integration of vision feedback and robotic systems to perform complex tasks. For this thesis effort, an operational static look-and-move RVSS was developed, which now provides a foundation for further research in the areas of better object recognition algorithms and, more importantly, a dynamic look-and-move RVSS. Based on these results, the conclusions and recommendations will be presented.

VI. Conclusions and Recommendations

6.1 Summary

In this thesis research, a robotic visual servoing system (RVSS) was developed and demonstrated for an aircraft refueling application. This system was based on a PUMA-560 robot manipulator with a small Sony CCD camera rigidly attached to the third link. The vision system identified and provided location information regarding the refueling receptacle to kinematics algorithms, where PUMA motion control algorithms positioned the PUMA's end-effector (a simulated refueling nozzle) into the receptacle's slipway. Based on the testing and demonstration of the RVSS, several conclusions can be drawn.

6.2 Conclusions

The most general conclusion that can be drawn is that this study demonstrates the feasibility of autonomous aircraft refueling using vision as the means of recognition and receptacle gross location. In addition, the concept of autonomous refueling through the aircraft refueling receptacle developed by the Flight Dynamics Laboratory is definitely valid, since recognition and location information can be derived from this receptacle. But as stated in Chapter 1, vision should not be the principle means of nozzle insertion, since the accuracy of the visual feedback information was not that sufficiently precise.

6.2.1 Object Recognition In the area of object recognition, the results confirm that the use of indicators as recognition aids is valid and should be the preferred recognition technique. By creating a unique pattern out of simple white posterboard, both recognition and range information was derived based on the known parameters of the indicator patterns. For this research effort, the development of an "optimal" was not investigated, and the pattern used for this effort cannot be considered an "optimal" design. In the case of recognition time, by using a simple pattern, recognition occurred within one second of image preprocessing. As indicator patterns become more complex or simpler, recognition times should increase or decrease, respectively. As for range measurements, by using the width of the top indicator, a fairly accurate range equation was produced allowing close placement of the refueling nozzle to the refueling receptacle slipway.

Another conclusion was that the existing camera was inadequate for the task, since when the area illumination was decreased or increased, the aperture of the camera had to be adjusted manually to compensate for the change in lighting intensity. In regard to the illumination effects, structured lighting was a requirement for this research. Special image preprocessing will be required if this task is every to be implemented in uncontrolled lighting environments. Finally, simpler recognition algorithms and/or faster hardware will be needed to decrease the image acquisition to object recognition time, currently around three seconds. In order to increase the speed of the existing vision algorithms, work was begun utilizing the increased speed capability of the Vaccelerator board.

6.2.2 Visual Servoing In the area of visual servoing and PUMA control, a static look-and-move visual servoing system works fine for a ground-based aircraft refueling application. In order to perform dynamic look-and-move visual servoing, parallel processing will probably need to be developed in order to accomplish the aerial aircraft refueling task.

As for the use of the PUMA-560 as a test base, the PUMA performed as required for this thesis effort. But for further research, a more realistic laboratory model of the refueling hardware should be developed and utilized.

In addition, the static look-and-move RVSS provides a foundation for further research in this area, such as advanced object recognition algorithms or dynamic look-and-move visual servoing.

6.3 Recommendations

Based on the research accomplished for this thesis effort, a solid base has been established for continued work in the area of autonomous aircraft refueling using visual servoing techniques. The following areas should be considered as continuation efforts related to this thesis:

6.3.1 Brightness Normalization Preprocessing Conduct further research into scene illumination preprocessing, possibly based on Lambert's brightness normalization algo-

rithm, to overcome the effects of uneven image lighting, shadows, glare, etc. and enhancing object recognition, while minimizing processing time for overall object recognition.

6.3.2 Camera Optimization Replace the existing camera with one capable of autoadjusting aperture sizing in order to compensate for the overall area illumination level.

6.3.3 Partial Indicator Pattern Recognition Develop the necessary software required to perform recognition of partially displayed indicators in order to deviate from the strict search pattern path.

6.3.4 Vaccelerator Board Usage Modify the developed vision algorithms for use on the Vaccelerator board in order to decrease the processing time of the image acquisition and recognition algorithms, which currently is the system time sink.

6.3.5 Dynamic Look-and-Move If debugging the communications problems encountered is accomplished, a follow-on effort can test the current parallel processing dynamic look-and-move software configuration: vision on the Vaccelerator CPU and PUMA control on the VAXstation III's CPU. Parallel processing provides independent vision processing and PUMA motion control allowing continual vision updates on receptacle position while the PUMA is in motion towards the refueling receptacle (allowing tracking of a moving target). In a related area, the search mode developed for this thesis effort should be modified as well in order to perform the search in a dynamic look-and-move fashion.

Appendix A. *Equipment List*

This thesis effort and task demonstration would never been realized if not for the hardware resources available and used in the AFIT Robotics System Laboratory (ARSL). All the laboratory equipment used for this research effort are listed and described below:

PUMA-560 The six DOF robot manipulator manufactured by Unimation, Inc., as shown in Figure A.1. Used as servoing platform for this research task.

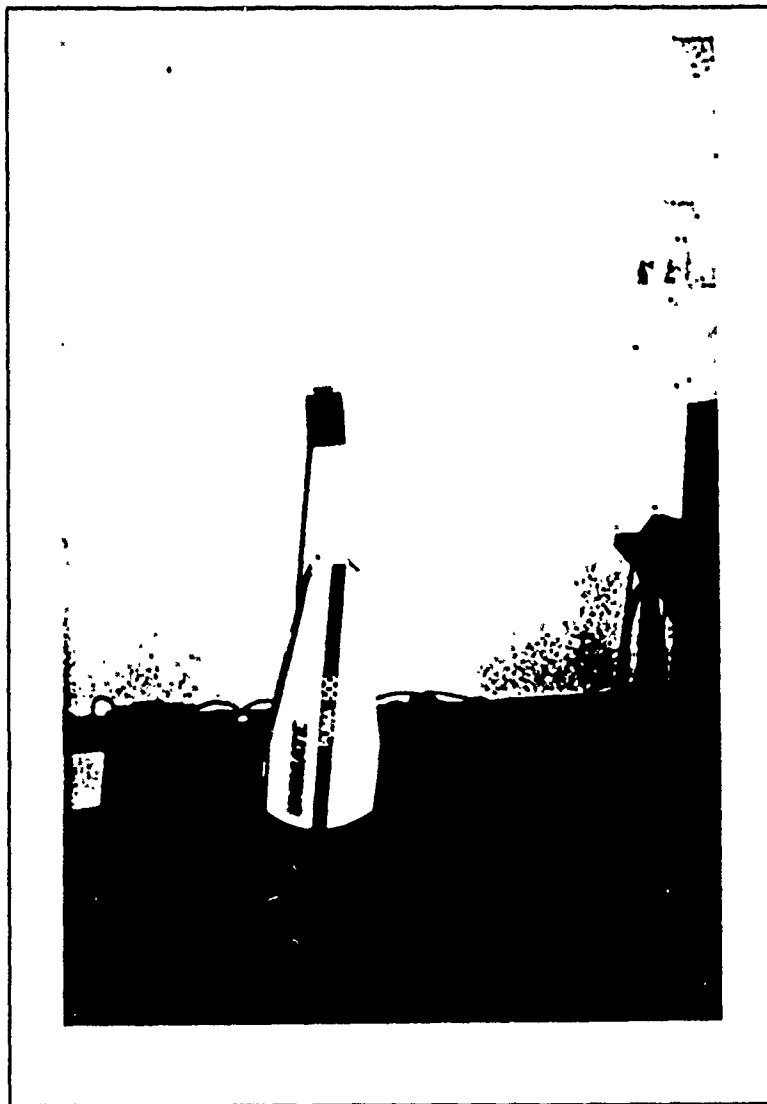


Figure A.1. PUMA-560

VAXstation III A microcomputer manufactured by Digital Equipment Corporation (DEC), as shown in Figure A.2. Housed the ITEX 100 Image Processing Board and executed all the software for performing the refueling task.



Figure A.2. VAXstation III "CYCLOP"

ITEX 100 An image processing system manufactured by Imaging Technology, Inc., for use in the VAXstation III. Performed the acquisition and pixel processing of an image used for receptacle recognition.

Vaccelerator A Vaccelerator AP-30 is a Q-BUS DMA peripheral containing its own processor, floating point hardware, and 4-megabyte memory, manufactured by Avalon

Computer Systems, Inc. for use in the VAXstation III. Used to create shared memory for data passing between the VAXstation III CPU and Vaccelerator CPU when attempting parallel processing. Vision software was processed by the Vaccelerator and PUMA motion control software was processed on the VAXstation III.

Video Camera A small, compact CCD video camera manufactured by Sony, Inc., as shown in Figure A.3. Attached to the third link of the PUMA-560 and used in image capturing.

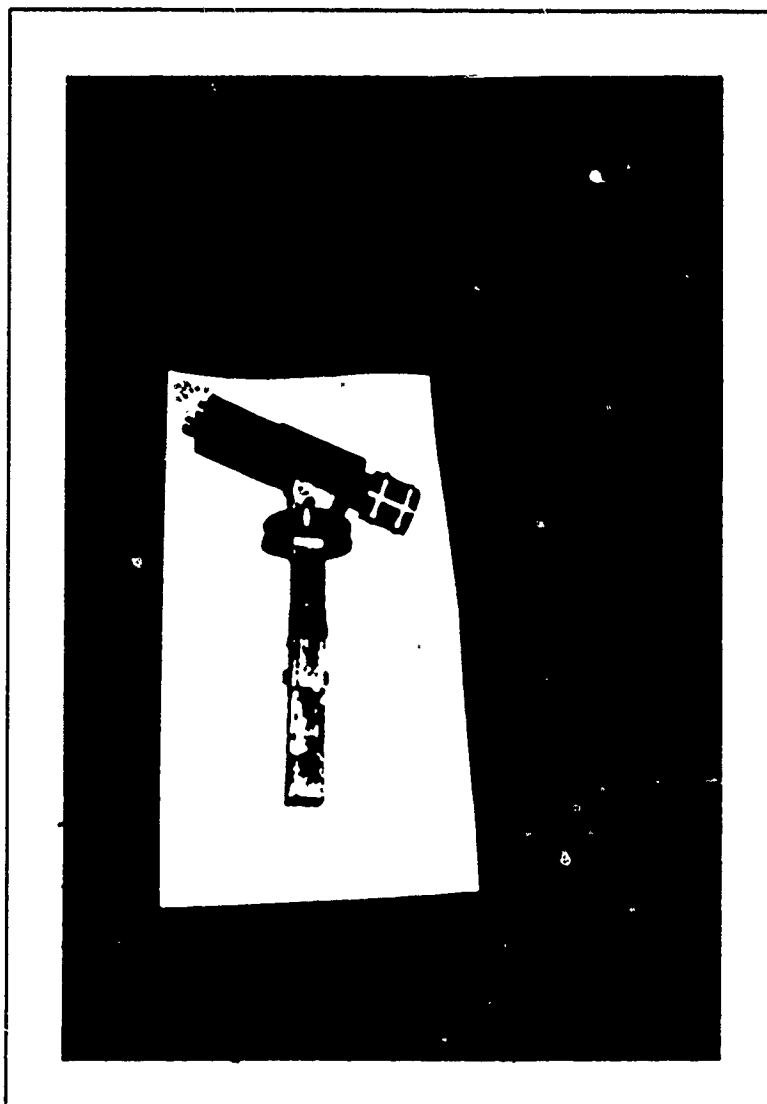


Figure A.3. Sony Compact CCD Video Camera

Refueling Receptacle Test Stand A stainless steel half-scale mock-up of a standard UARSSI aircraft refueling receptacle, as shown in Figure A.4. Manufactured on site at AFIT, WPAFB, OH. at the AFIT Model Shop for Duvall's research [4].

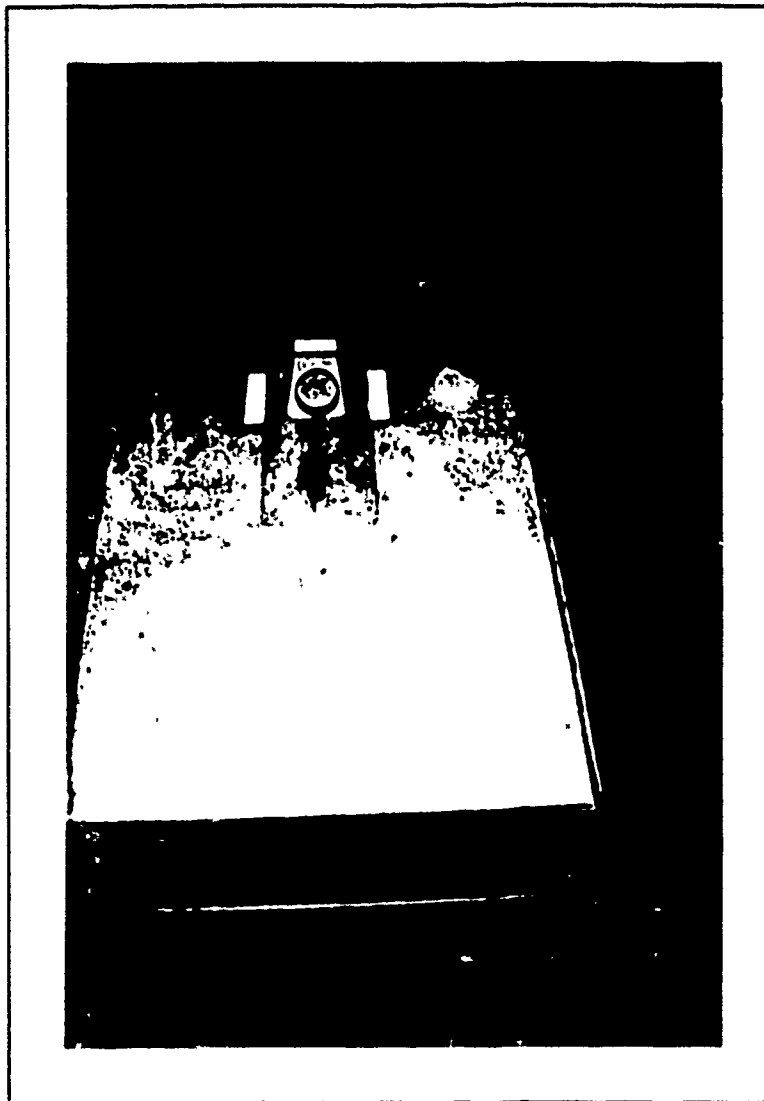


Figure A.4. Refueling Receptacle Test Stand

Refueling Nozzle A simulated aluminum refueling nozzle capable of being screwed into an attachment point at the end of link 6 on the PUMA, as shown in Figure A.5. Manufactured on site at AFIT, WPAFB, OH. at the AFIT Model Shop for Duvall's research [4].

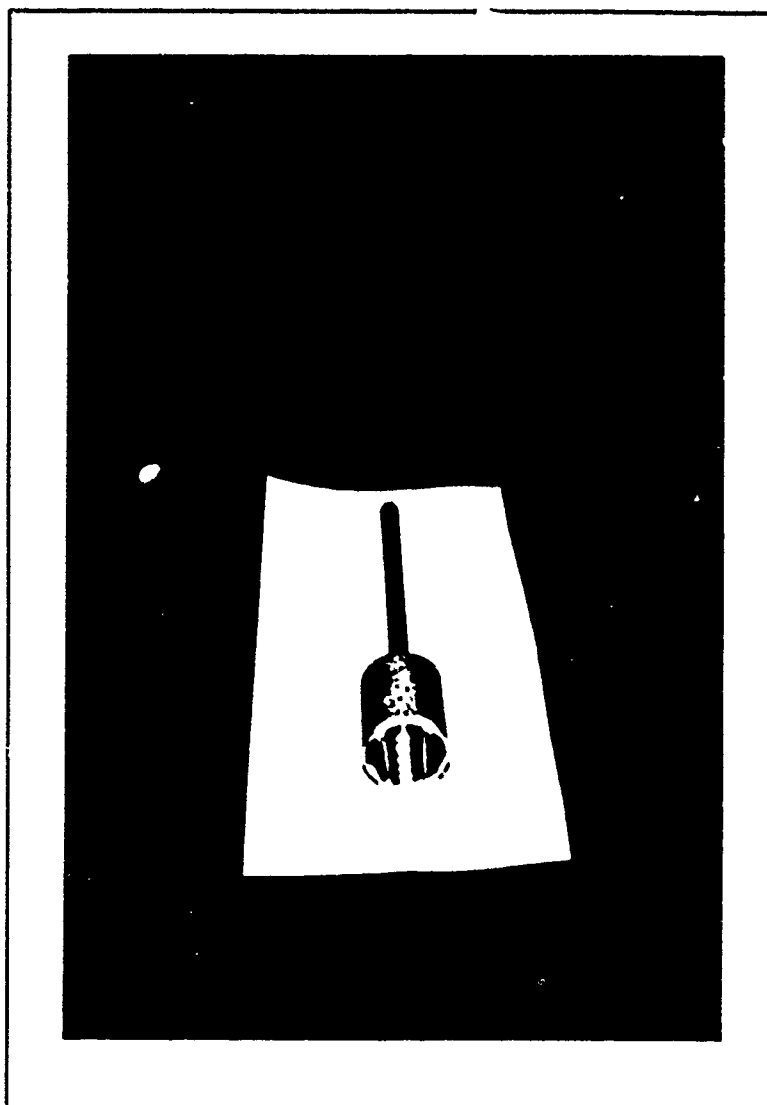


Figure A.5. Simulated Refueling Nozzle

Appendix B. *Robotic Visual Servoing System (RVSS) Software Flow*

B.1 *Introduction*

The purpose of this appendix is to provide a clear understanding of the flow and cycling of the software developed for the static look-and-move RVSS. Figure B.1 provides a logic diagram illustrating the flow of control and the decision nodes within the RVSS software. All code can be found in the AFIT Robotics Systems Laboratory in an internal report (ASRL-89-14) [7].

B.2 *RVSS Software Flow Discussion*

Upon execution of the run file "rvsmain", a welcome message is displayed on the terminal screen, as shown in Figure B.2, and questions the operator to switch power on to the camera. Upon depressing the return key, the ITEX 100 system is initialized. Upon successful ITEX initialization, a welcome message is displayed on the video monitor, see Figure B.3. There is no test to verify successful initialization of the ITEX 100 system, so if the ITEX 100 welcome message is not displayed on the screen, then the operator should input a control C to stop RVSS software execution, and determine the problem with the ITEX 100 system.

After the ITEX 100 system is initialized, the PUMA initialization routines are executed. The PUMA is initialized based on the AFIT ARCADE FORTRAN programs supporting the PUMA. This program *pumacal* calibrates the PUMA and passes back the current PUMA joint angles and calibration error status. If a calibration error has occurred, then an error message is displayed on the terminal screen, and the RVSS is exited. If no error was incurred during calibration, then a communication's channel is opened between the LSI 11/73 and "CYCLOP." Figure B.4 shows the screen message associated with execution of *pumacal*; the return key must be depressed for its continued execution. After calibration, the PUMA is placed at its initial search position completing system initialization.

The next phase of software execution is a recursive search algorithm, suitably called *search*. During this phase, an image is acquired at the current search position, and that

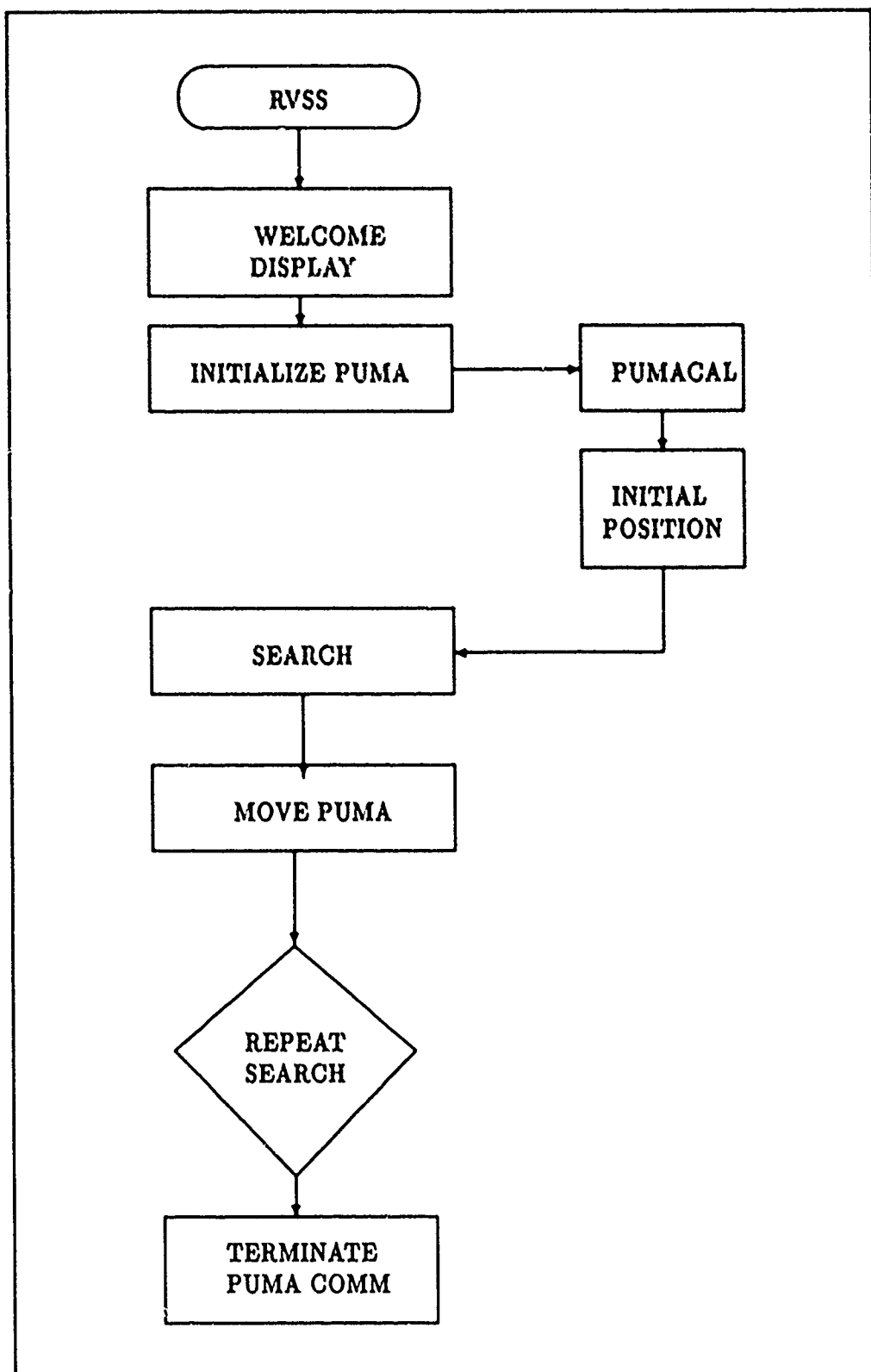


Figure B.1. Robotic Visual Servoing System (RVSS) Software Logic Diagram

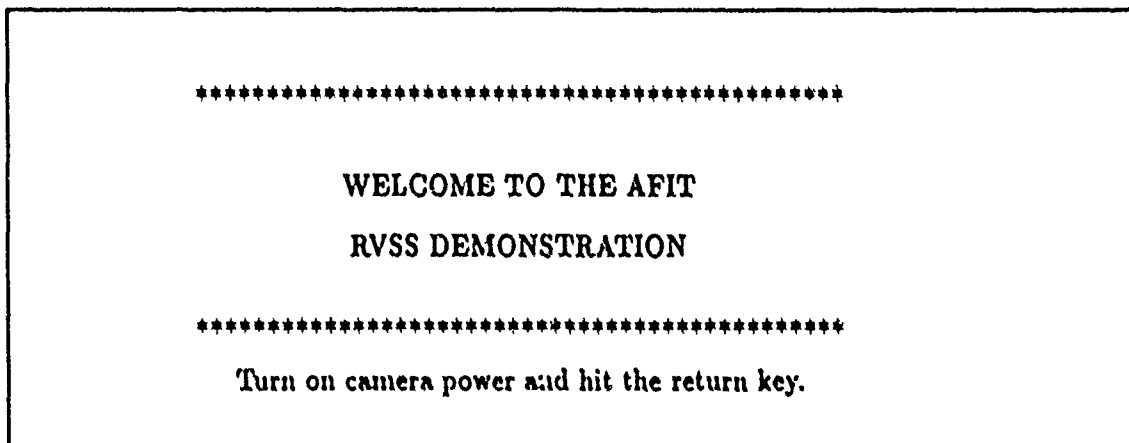


Figure B.2. RVSS Welcome Message

image is passed to the recognition algorithm, where a histogram is produced from the image and a white level threshold value is produced. Based on this threshold value, this image is binarized into black and white pixels (black is background and white is considered possible images). An analysis algorithm determines if there are too many or too few white pixels in the image. If there are, then the PUMA is moved to its next search position and the search cycle continues. If there are not, then it determines if the white pixels represent the object.

If the receptacle is identified, then the range and offset parameters of the receptacle are determined and passed to the forward kinematics algorithm, where the cartesian position of the receptacle is calculated. After the position is found, the position of the receptacle is tested to determine if it is in range of the PUMA. If the receptacle is not in range, then the receptacle is centered in the image, and the search is continued, but with the search position frozen to this search area. Again, image acquisition and object recognition is performed to track the found receptacle. If the receptacle is lost after it has been recognized and tracking is being performed, then the PUMA is placed back into its initial position, at the current joint one angle), and the search is begun again in a normal manner.

If or when the receptacle is within range, the inverse kinematics module is executed and the first three PUMA joint angles required to place the end-effector near the receptacle

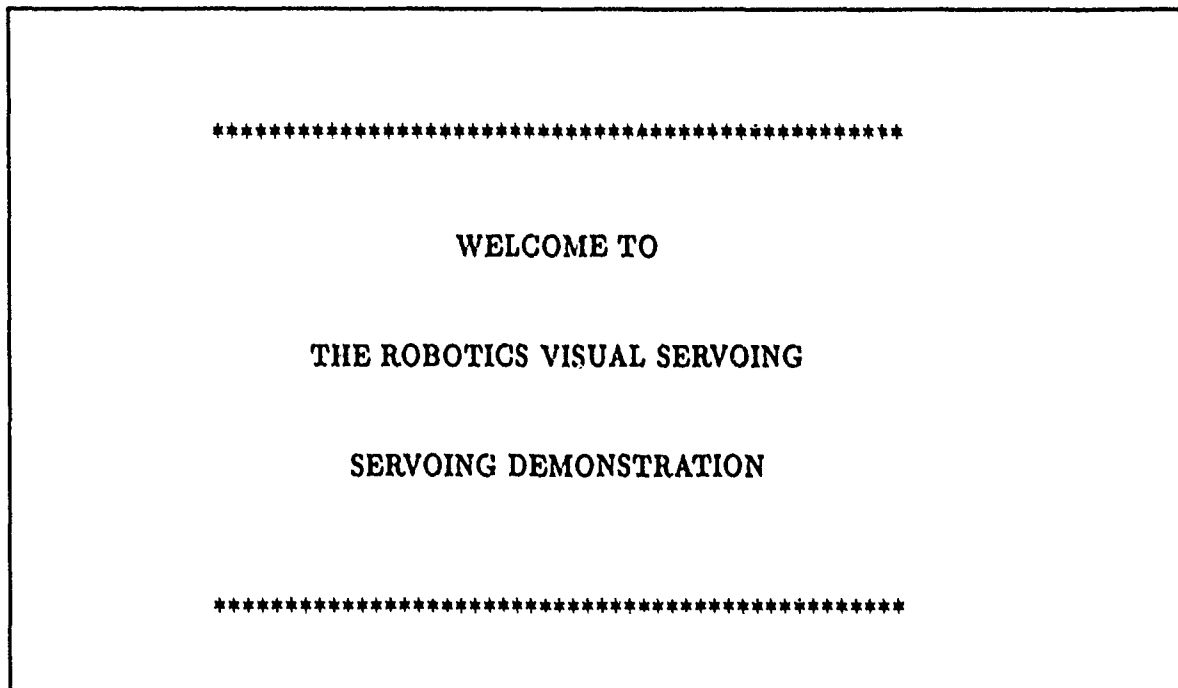


Figure B.3. ITEX Welcome Message

are calculated. Prior to any movement these angles are tested, and if any of them violate PUMA motion constraints, then the receptacle is again centered in the image and the search algorithm is begun again, with the search position frozen. If the angles are within the range of the PUMA's motion area, then the PUMA is moved forward to the receptacle to the calculated first three joint angles.

The operator is then asked to input a carriage return to place the PUMA back to its initial position stance, and asked if another search is requested. If not, then the communications with the LSI 11/73 is terminated and execution of code is stopped. If the operator wishes to proceed again, then the search is begun at the current joint one position), without equipment initialization and welcome messages being displayed.

```
*****  
WELCOME TO  
PUMACAL  
*****  
Turn on robot arm power, and hit the return key when ready.
```

Figure B.4. PUMACAL Welcome Message

Bibliography

1. Miller, Capt. Mikel M. *Implementation of a Visual Servoing System for Evaluation of Robotic Refueling Applications*. MS Thesis, AFIT/GE/ENG/87D-45. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1987.
2. "No. 1 B-2 Completes First Phase of Flight Envelope Expansion Tests," *Aviation Week and Space Technology*, pp. 30, October 2, 1989.
3. Leahy, Capt. Michael B. Personal Interviews. Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1989.
4. Duvall, David J. *Motion Control for Aircraft Refueling Applications*. MS Thesis, AFIT/GA/ENG/88D-11. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1988.
5. Milholen, Vernon W. *Experimental Evaluation of Impedance Control for Robotic Aircraft Refueling*. MS Thesis, AFIT/GE/ENG/89D-32. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1989.
6. Lambert, Capt. Lawrence C. *Evaluation and Enhancement of the AFIT Autonomous Face Recognition Machine*. MS Thesis, AFIT/GE/ENG/87D-35. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1987.
7. Shipman, Capt. Richard P. *Robotic Visual Servoing System (RVSS) User's Manual and Software Description* Internal Report ASRL-89-14. AFIT Robotics and Automation Laboratory, Dept. of Electrical Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1989.
8. King, F. G. and others. "Vision Guided Robots for Automated Assembly," *1988 IEEE International Conference on Robotics and Automation*. pp. 1611-6, 1988.
9. Kiovo, A. J. and N. Houshangi. "Real-Time Vision Servoing of a Robotic Manipulator with Self-Tuning Controller," School of Engineering, Purdue University, pp. 1-35, August 1989.
10. Gershon, David and Isaac Porat. "Vision Servo Control of a Robotic Sewing Machine," *1988 IEEE International Conference on Robotics and Automation*, pp. 1830-34, 1988.
11. Harrell, R. C. and others. "Vision Guidance of a Robotic Tree Fruit Harvester," *Intelligent Robots and Computer Vision*, Vol 579, pp. 537-45, 1985.
12. Feddema, J. T. and O. R. Mitchell. "Vision-Guided Servoing with Feature-Based Trajectory Generation." *IEEE Transactions on Robotics and Automation*, Vol 5, pp. 691-700, October 1989.
13. Fu, King-Sun, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill Book Company, 1987.
14. Fairhurst, Michael C. *Computer Vision for Robotic Systems*. Hertfordshire, Great Britain: Prentice Hall International Ltd, 1988.
15. Goshtasby, Ardesbir. "Template Matching in Rotated Image," *IEEE Transaction on Pattern Recognition and Machine Intelligence*, PAM-7, pp. 338-340, May 1985.

16. Kim, Hwang-Soo and others. "Object Recognition Using Multiple Views," *IEEE International Conference on Robotics and Automation*, pp. 28, 1985.
17. Wang, Y. F. "Matching Three-Dimensional Objects Using Silhouettes," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-6, pp. 513, July 1984.
18. Hung, Yubin. "Passive Ranging to Known Planar Point Set," *IEEE International Conference on Robotics and Automation*, pp. 80, 1985.
19. Imaging Technology, Inc. "FG-100-Q User's Manual." Woburn, Massachusetts: July 1987.
20. Leahy, M. B., Jr. *The RAL Hierarchical Control System User's Guide, Version 1.0*. Internal Report No. RAL-67. Robotics and Automation Laboratory, Dept. of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy NY, April, 1986.
21. Avalon Computer Systems, Inc. "Vaccelerator Operation's Manual." Glendale, California: June 1989.

Vita

Richard Shipman was born on October 18, 1962, in La Mirada, California. He spent most of his childhood moving around the western United States from California to Montana. He graduated from high school in Globe, Arizona (a small copper mining community) in 1980. He then attended California State University, Fullerton, California and graduated in 1985 with a Bachelor of Science in Electrical Engineering. After graduation he attended the Air Force Officer's Training School in San Antonio, Texas and on April 5, 1985 he was commissioned a Second Lieutenant.

His first duty assignment was at Los Angeles Air Force Station, Los Angeles, California and he was assigned to Systems Command Space Division. His duties there included DoD Space Systems Mission Readiness Manager, Space Transportation System (STS) Flight Planning Manager, and STS Payload Safety Manager. While as Space Division, he met and married Marie Brassie in 1987, and as an added bonus acquired an eight-year old son, Derek. In 1988, he was assigned to the Air Force Institute of Technology (AFIT) School of Engineering, Wright-Patterson AFB, Ohio to earn a Master's Degree in Electrical Engineering. In April of 1989, he was promoted to Captain.

Permanent address: 3598 Golden Meadows Ct.,
Dayton, Ohio 45404

Vita

Richard Shipman was [REDACTED] He spent most of his childhood moving around the western United States from California to Montana. [REDACTED]

[REDACTED] He then attended California State University, Fullerton, California and graduated in 1985 with a Bachelor of Science in Electrical Engineering. After graduation he attended the Air Force Officer's Training School in San Antonio, Texas and on April 5, 1985 he was commissioned a Second Lieutenant.

His first duty assignment was at Los Angeles Air Force Station, Los Angeles, California and he was assigned to Systems Command Space Division. His duties there included DoD Space Systems Mission Readiness Manager, Space Transportation System (STS) Flight Planning Manager, and STS Payload Safety Manager. [REDACTED]

[REDACTED] In 1988, he was assigned to the Air Force Institute of Technology (AFIT) School of Engineering, Wright-Patterson AFB, Ohio to earn a Master's Degree in Electrical Engineering. In April of 1989, he was promoted to Captain.

[REDACTED]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/89D-48			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION School of Engineering		6b. OFFICE SYMBOL (If applicable) AFIT/ENA		7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Flight Dynamics Laboratory		8b. OFFICE SYMBOL (If applicable) WRDC/FEIMB		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Visual Servoing for Autonomous Aircraft Refueling UNCLASSIFIED					
12. PERSONAL AUTHOR(S) Richard P. Shipman, Capt, USAF					
13a. TYPE OF REPORT MS Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1989 December	
15. PAGE COUNT 92					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Visual Servoing, Robotics, Aircraft Refueling		
FIELD	GROUP	SUB-GROUP			
12	9				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Thesis Chairman: Michael B. Leahy, Capt, USAF					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Capt Michael B. Leahy			22b. TELEPHONE (Include Area Code) 513-255-4283		22c. OFFICE SYMBOL AFIT/ENG

BLCK 19. (Cont)

The design and development of autonomous robotic systems for military applications is of major interest to the Air Force. The robotics research group at the Air Force Institute of Technology (AFIT) is actively involved in autonomous aircraft refueling research. The approach being undertaken at AFIT is the integration of vision and robot technologies allowing visual information, regarding identification and location of an aerial refueling receptacle on an aircraft, to be fed back to a robot controller which develops trajectory information needed for compliant robot motion.

The purpose of this thesis effort was to modify, develop, and evaluate existing visual processing algorithms to allow the AFIT PUMA-560 to visually acquire and track a half-scale mock-up UARRSI aerial refueling receptacle, and place a simulated refueling nozzle in the close proximity of the receptacle's slipway. This was accomplished by developing visual object recognition and robot servoing algorithms which: analyzed images, recognized and determined the position of the receptacle, and calculated proper PUMA joint angles for end-effector placement.

Based on this approach, a static look-and-move robotic visual servoing system (RVSS) was demonstrated which: identified, located, and servoed the PUMA's end-effector to the close proximity of the refueling receptacle. The RVSS provides the visual recognition and servoing necessary so that when combined with a compliant motion controller completes the insertion task, thereby completely demonstrating the AFIT concept for robot refueling. In addition, the initial architecture for a dynamic look-and-move RVSS was developed based on parallel processing. This system provides AFIT with the capability to meet Air Force initiatives for research on further design, development, and application of visual servoing for autonomous aircraft refueling tasks.

