Telerobotic Rendezvous and Docking Vision System Architecture

Final Report
NASA SBIR Phase II
Contract Number NAS5-30709
Goddard Space Flight Center
Greenbelt, MD
Effective Date Aug 11, 1989

Triangle Research and Development Corporation
P.O. Box 12696
Research Triangle Park, NC 27709-2696
919-781-8148

Prepared by

Ben Gravely, Ph.D., Principal Investigator
Donald Myers, Ph.D.
David Moody
1/17/92

Table of Contents

Introduction

The organization of this report is based on the technical objectives contained in the Phase II proposal description. The overall goal of this project as described in the proposal is "*the development of a microcomputer-based vision system architecture that allows a robot system to identify an object, determine its range and orientation, and access explicit structural data on the acquired object for mating with other objects.*" The heart of the Phase II program was divided into four interactive research areas.

Vision System Definition and Design
Software Development
Optics Systems Research and Development
Robot Application and Demonstration

This report is partitioned into the research areas listed above, with the Phase II research objectives presented within each of the research areas. As described in the quarterly reports, equipment failure and delays by vendors significantly delayed the execution of the program, so that an extension from August to December, 1991, was requested.

Vision System Definition and Design

Objective 1. Project Coordination and Review

At the outset of the program, a meeting was held at NASA GSFC to meet the appropriate technical representatives and tailor the research efforts to best meet the needs and priorities of the NASA robotics program. Throughout the program, the technical progress, priorities, and milestones were reviewed with Dr. Del Jenstrom at GSFC. In addition a program review was conducted at NASA with Del Jenstrom and John Vranish, and several laboratory visits were conducted by Dr. Jenstrom.

Objective 2. Design Second Generation Vision System

In Phase I, a 16-bit 80286 microcomputer was used that required bank switching to address the image memory. In Phase II, a Macintosh computer was proposed for several reasons. The 68030 CPU of the Macintosh has no internal addressing boundaries, and second, much of the robot interface was to be performed by Lord Corporation, who had a Macintosh environment. Also, a quad processor board was available for the Macintosh that claimed parallel processing capabilities up to 40 MIPS.

However, as the program progressed, it was decided to use 80386 CPU technology. Several criteria were involved in this decision. First, 80386 technology was more compatible with the robotic programs at NASA. Second, Lord Corporation closed their robotics program, making compatibility with their system irrelevant. The Lord Puma arm was subsequently acquired by TRDC to continue the work in-house. Even though the Macintosh does not have internal memory boundaries, systems programming requires a thorough understanding of the window-based interface to the Macintosh operating system. Text-based interfaces are relatively easy to implement on the PC. We intended to use the PC not only for the vision processing, but also to control the robot. We anticipated that a bus interface card would have to be developed. There was considerable in-house experience with implementing IBM PC-compatible systems and very little experience with Macintosh-based systems. As the program progressed, the relative merits of each type of system became more obvious. In the course of discussions with NASA personnel, the speed of computation was de-emphasized in favor of robust algorithms. The speed potential was also greatly reduced when the Macintosh option was dropped.

A 25 MHz 80386 PC-AT compatible was purchased with an 80387 co-processor, 4 Mbytes of RAM, and an 80 Mbyte hard drive. Unfortunately, the time required to modify the contract to effect this purchase considerably delayed the onset of the technical development effort. Also, after the computer was acquired, it had circuit board problems that required three rounds of parts-swapping to correct.

The acquisition of video cameras was initially delayed by a mismatch between image sensors. It was desired that the CCD target and vidicon tube target be of the same size format, so that a single lens and image size could be used on each for comparison. Generally, CCD cameras use a 1/2 inch target and vidicons use a 2/3 inch target. Two 525 line, 2/3 inch format cameras were finally acquired for the program.

The 8 bit Targa M8 image acquisition card from Truevision, purchased during Phase I, was also used in the Phase II program.

Objective 3. Develop the Capability to Produce High Quality Test Targets for Optimization Studies.

The test target is basically a bar code surrounded by a special border. Each character of the bar code is represented by nine bars, alternating between black and white, three of which are wide - hence the name "code 3 of 9", or "code 39". There are three spacings which must be defined - the narrow width, the wide width, and the gap between characters. An additional parameter is the length of the each bar. The standard format for USD-3 (i.e,.code 39) allows some latitude in choosing these widths. Therefore, a "middle of the road" choice has been selected in the ratio of 1x:2x:3x for narrow:gap:wide. The length of each bar was chosen to achieve a best fill factor for the border, and is not a critical parameter.

A bar code graphics generator was developed to disassemble an alphanumeric string input, generate the binary code representing wide and narrow, black and white bars for each character, and plot the code as a screen graphic representation of the bars. Software was developed to transfer the screen images to the laser printer using HALO library routines. A rectangular border generator was added and the program interfaced directly with the HP LaserJet III printer through the HP graphical language. The program produced the bar code and rectangular border with the precise proportions defined in Phase I and printed these in landscape format on the LaserJet III printer. Using code 39 symbology, the algorithm allowed the control of such parameters as rectangle size and top/bottom spacing between the border and the bar code. The bar code and border images were then translated into 300 dot/inch resolution for printing on the LaserJet engine. The images compare favorably to the typeset bar codes received previously from CompuType and used in both Phase I and Phase II until now.

The desire to quickly modify the target label dimensions, spacings, and border shape finally led to the abandonment of the special target-generating program in favor of Claris CAD, a general purpose CAD package for the Macintosh, which was used thereafter to produce the target labels. A library of character symbols was developed for ease of incorporation into a label. Symbols were encrypted in code 39, including the numerals 0 through 9 and the '*' symbol used to indicate the start and end of the label. The CAD program was able to accurately reproduce the label geometry on the Hewlett Packard LaserJet III laser printer.

The symbol library with dimensions is depicted in Figure 1. The height of the code stripes was selected to fit within the chosen border dimensions. The bar code can be scaled to allow more characters within the available length up to the limit of resolution of the stripes by the camera.

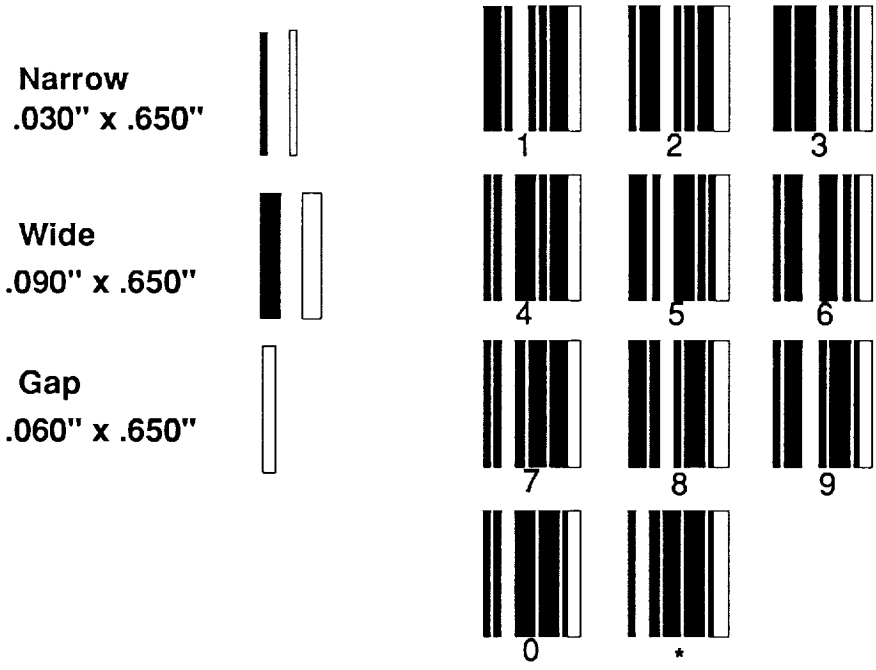

Figure 1. Target Symbols

A typical target label is shown in Figure 2. The border of the label consists of both the black rectangular border surrounding the symbols and the white space surrounding the border. The white space is required to provide sufficient visual separation of the label from the object on which it is attached. The white circles located in the corners of border contain the target corner points (TCPs). It is the centroids of the white spaces that are used as the coordinates of the four corners of the target label.

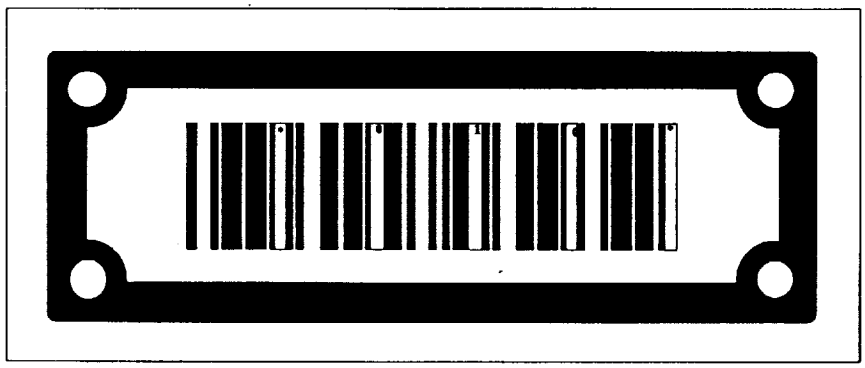

Figure 2. Target Label Geometry

The diameter of the white circles and the widths of the black and white spaces of the border are based on several criteria. Circle diameter was selected so that the circle could be discriminated at the maximum viewing distance of the camera. The maximum viewing distance was determined by the work volume of the robot to which the camera was attached. The width of the border black space was chosen so that TCP white space did not merge with the white space inside or outside of the border black space. Again, this selection was based principally on the maximum viewing distance of the camera. The width of the white spaces both inside and outside

of the border black space was chosen to be equal to the width of the black space. A white space width equal to the black border width was judged acceptable to provide discrimination for viewing angles up to 45 degrees.

Using the geometric relationships developed, the target can be sized for any application of interest to NASA.

Software Development

There are four steps required to locate and identify an object tagged with the special label. The first step is to *segment* the visual image into discrete objects. In this context, an *object* is defined as a union of interconnected pixels. Objective 7 addresses the segmentation problem. The second step is to *identify* the target label from the among the set of all possible objects. The third step is to determine the *pose* (position and orientation) of the label relative to the camera. Objective 5 addresses these two steps. The final step is to decode the *title* on the label and thereby identify the object to which the label is attached. Objective 6 deals with this final step.

Objective 4. Convert Existing Phase I Software into C Language

The Phase I BASIC language software determined the four corners of the original border (without holes), performed the distance and orientation calculation, and decoded the bar code for object identification. The label was placed on a large white background and scene segmentation was not required to find the label. The size of the program did not tax the DOS limitations of the PC.

As C language software development began for all the expanded tasks of the Phase II program, memory address limitations in the PC became a significant problem. A PC running under DOS in real mode is essentially limited to 640 Kbytes addressed in ten 64 Kbyte segments. Since one 64K segment of memory is required for DOS and four 64K segments for each camera image, the application program and compiler must compete for the remaining segments of conventional memory. During the program it became apparent that running in 8086 real mode with Microsoft QuickC was not acceptable. Most of the 4 Mbytes of memory in the computer were not available for program use. The necessity of manipulating 512x480x8 bit images required breaking through the 640 Kbyte DOS boundary. A study of other compilers was begun. Microsoft C 6.0 and Turbo C++ allow program overlays that permit programs segments to be stored beyond 640 K. The existing code was recompiled into Microsoft C 6.0. However, it was slow and cumbersome to use, and still did not allow direct addressing above 640K. The search for a usable compiler continued. Finally, the WATCOM C 386 compiler, used in conjunction with a DOS extended program, solved the memory problems.

WATCOM C 386 uses the PC in 80386 protected mode, resulting in a linear address space up to 1 Gbyte of system memory. In addition, the compiled code is highly optimized for speed. The switch was made to the WATCOM C compiler which resulted in approximately one month lost in the program. However, all the memory limitations and addressing difficulties were eliminated.

All Phase I algorithms, including corner detection, label decoding, and inverse perspective transform were recoded in WATCOM C. Difficulties were experienced in operating the Targa M8 video frame grabber card using the software tool kit supplied from Truevision. Acquiring a picture, storing the image, recalling the image, and other functions could not be successfully implemented with their "C" software source programs. A new set of image acquisition functions for the Targa frame grabber card had to be acquired from Truevision and rebuilt by the WATCOM compiler before the video system would operate properly.

A software shell, through which the user interacts with the program, was written in C to bind the different modules for menu and setup control, image capture, and analysis into a single package. The shell linked the MENU module, the IMAGE module, and the ANALYSIS module. The MENU module includes a set of spreadsheet-like pages that are be used to enter and modify static parameters needed in program execution, such as memory address locations, or setup commands. The IMAGE module controls the camera functions of taking a picture and the storage and retrieval of images to disk. The functions establish multiple copies of the video image in RAM for processing.

Objective 5. Develop Second Generation Border Recognition and Orientation Algorithms

The principal tasks required for border recognition and pose determination are: a) the segmentation of the label border from all other objects in the scene, b) the determination of the coordinates of the four corner points of label border in the image plane, and c) the application of an inverse perspective transform to the corner points to determine the pose of the target label relative to the camera.

Target Label Discrimination
The initial attempt at discriminating the label border from other objects consisted of applying a linear discriminate function to a set of features characterizing each object. The feature set consisted of the following statistical measures:

1. area - total number of pixels which make up the object
2. perim - total number of pixels tracing the perimeter of the object
3. numOn - the number of pixels in the object area above a threshold value
4. density - the ratio of pixels above threshold to the total pixel area of object = numOn/area
5. p^2/A - the nondimensional ratio of the square of the perimeter divided by the area
6. PAH - the perimeter angle histogram

Area and perimeter are not useful parameters by themselves because they are size dependent. Size invariant measures such as p^2/A and density were also examined. The PAH (Perimeter Angle Histogram) contains the calculated angles between all the pairs of neighboring perimeter pixels. The histogram is constructed as the perimeter of the object is traced in a counter-clockwise direction. Since each pixel can have a neighbor in only one of eight positions, the angle between neighboring pixels can be resolved to only 360/8 (=45) degrees. The relative distribution of pixel angles into eight bins provided an additional indication of the shape of the object and therefore its identity. It was anticipated that the PAH would be useful in determining an initial estimate of the angles of the sides of the label border independently of the angle determination based on the four corner points.

A trainable, deterministic pattern classifier was designed and coded. The classifier was based on a perceptron, a single layered neural network that is similar to a linear discriminate function. To facilitate classifier training, a file system was developed to store and retrieve parameter records for each object. Each record consisted of the statistical parameters associated with the object and a flag indicating whether the object was a member of the class of label borders. The file system permitted the user to calculate the parameters for each object in an image, and then created a disk file into which each object record was written or appended to an existing file. It is possible, therefore, to build a single disk file containing records of objects from many camera images. The file could then be used to train the classifier over a large number of objects.

Experimentation with camera images showed that the parameters described above provide reasonable discrimination but were not sufficiently robust to be foolproof. A range of values for

p^2/A large enough to encompass label borders of any orientation and distance is unfortunately broad enough to admit objects not of rectangular shape. The major source of error appeared to be digitization noise, which was most prominent when the label border was oriented at a 45 degree angle relative to the scan line. The PAH also suffered from similar digitization limitations.

A natural extension of the histogram concept is the Fourier transform. The coefficients of the Fourier transform of the perimeter angles contain information related to both the shape and size of the object. The first coefficient is proportional to the size of the object; the second, to its aspect ratio; the third, to its triangularity; and so on. A literature review [1,2] indicated that if the border of the object can be parameterized, the coefficients of a Fourier series expansion of the border often prove robust in discriminating between object shapes. It was therefore decided to pursue Fourier analysis as a candidate methodology for shape discrimination.

There are two principal methods for applying Fourier analysis to the border points. The first method is to find the magnitude and angle of a vector from the CG to selected points on the border of the object. The border points are typically selected by sweeping the vector in constant angular displacements through a complete 360 degree arc around the border. The magnitude of the radius then becomes the real number input to the Fourier analysis. The second method treats the x,y coordinates of the border as the real and imaginary components of a complex number. Ideally, the border points should be sampled with a constant displacement arc. The resulting border "signature" can be treated as an infinite waveform with a fundamental frequency equal to one 360 degree period around the border.

The Fast Fourier Transform (FFT) was selected as the computational tool to implement the analysis. If n samples are evenly spaced within one complete revolution of the boundary, the FFT yields the coefficients of the first $n/2$ harmonics of the Fourier series expansion of the infinite waveform. The magnitude of the coefficients, however, is shaped by the Fourier transform of one period of the boundary signature. In other words, the magnitude of the coefficients are not identical to those that would be obtained if a Fourier series expansion was performed on a waveform of infinite duration; however, the resulting coefficients are unique for each object shape. An additional constraint imposed by the FFT is that the number of sample points must be a power of two. The bandwidth of the FFT, and consequently the number of harmonics contained in the Fourier analysis, is also dependent upon the number of sample points.

Since the scene segmentation (described in Objective 7) produces the object boundary, the FFT method was selected to parameterize the object boundary. The number of sample points was determined through experimentation. It was found that 16 points (yielding the coefficients of 8 harmonics) provided sufficient discrimination between rectangular label borders and rectangles of different aspect ratios.

The magnitude of the DC component, $|f(0)|$, of the Fourier series is a function of the position of the object in the image plane. In fact, if the coordinates of the sampled border points are referenced to the CG of the object, $|f(0)|$ approaches 0. The $|f(1)|$ coefficient is a function of object size, and the $|f(n)|$, $n>1$, coefficients are functions of object shape. The phase of the coefficients is a function of object orientation. Literature review has revealed a normalization technique to produce coefficients that are independent of position, orientation, and size of the object. However, the technique is computationally intensive and deemed not necessary at this stage of the work. The approach selected was to ignore the phase information of all coefficients and normalize $|f(n)|$, $1<n<8$, by $|f(1)|$.
It appears that five of the Fourier coefficients in combination with density (defined above) provide a sufficient feature set for object discrimination. The pattern classifier therefore simply checks to determine if each of these parameters is within acceptable ranges. The ranges are

produced by a training module that accepts sample label borders and determines mean, variance, maximum and minimum values for each of the parameters. Only the maximum and minimum parameter values are used by the classifier for discrimination.

Location of Corner Points
Five different methods for locating corner points were coded and tested experimentally.

1) An initial estimate of the corners of the label border can quickly be determined by differentiating the chain code of the border points found above. For this application, it was reduced to running a 1x3 differentiator window through the 16 sampled points of the border. The points with the 4 largest derivatives were selected as the corner points of the border.

2) A finer estimate can be determined by running a 1-D differentiator window through the entire set of border points. The size of the 1-D differentiator window must be chosen as a compromise between sensitivity to noise and precision of corner point location.

3) Another method performs a first order curve fit to each of the 4 groups of sample points found by method 1) to represent the edges of the border. The intersections of the 4 lines determine the corner estimates. The distribution of the 16 samples around the perimeter yields at most 4 and at least 2 points for each curve fit.

4) The corner search method used in Phase I was also implemented in C. The method is based on stepping through each border points until an exact corner point is found, and contains five algorithms. To save time, the method was applied to the reduced data set obtained after method 1) defined a small window in the neighborhood of the corners.

5) The fifth method is similar to the second, except that all pixels in the border were fit to one of the 4 first order curves. The curve fit is achieved by finding the principal eigenvector of each of the point groups. Each of the border pixels was sorted into one of the four line segments using the Hough transform [6].

Method 1) offers the advantages of speed, efficiency, and independence of object orientation but only produces rough estimates of the corner points for coarse, real-time servoing of the robot towards the target. With method 2), the proper size of the filter and magnitude of the coefficients could not be found to produce acceptable corner points under various lighting conditions and viewing angles. Method 3) was found to offer no advantages over method 1) alone. Method 4) was particularly sensitive to border noise when the label was aligned with the horizontal viewing axis. Method 5) was robust, but slow. A Hough transform on a larger number of points required several minutes of processing time.

A new technique was developed as a compromise between the desire for fast processing speed and the desire for subpixel resolution achieved by using multiple pixels to locate a corner. In this approach, small white circles were embedded within the black border at each of the four corners. The diameter of the circles was chosen so they could be detected by the vision system at the maximum distance in the work envelope of the robot. The initial estimate of the corner locations comes from the 1D filter applied to the 16 sample border points (method 1), and is used to define a small rectangular window containing a circle. The scene segmentation image processing algorithms are then applied to this window to discriminate all objects within the window. The circle is easily discriminated from other objects based on size and Fourier coefficients. The CG of the circle gives the corner location, or TCP, used in the orientation calculation.

The final method is robust, and offers the advantages that the same scene segmentation algorithms defined in Objective 7 are used to locate the corners. Sub-pixel resolution of a corner point is easily achieved by finding the CG of all points in the circle. The method increases the physical complexity of the target slightly, but eliminates all the rotational sensitivity of straight lines experienced in the original algorithms. The potential to encode the scene segmentation algorithms in hardware for fast operation and multiple use in each part of the analysis offers an economy of code and speed of operation for small computer platforms.

Determination of Label Pose

Determination of the position and orientation of the label relative to the camera involves several steps. The first step is to determine the coordinates of the four target corner points (TCPs) relative to the camera. Figure 3 illustrates the geometry of the problem. Given the coordinates $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$ of the TCPs in the image plane of the camera, the task is to find the coordinates of the actual TCPs $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ in the target plane. The algorithm developed by Yung [3] was used in Phase I to find the $\mathbf{p}$ vectors. The second step is to find the transformation matrix Λ that relates camera and target frames. The problem reduces to the solution of the matrix equation

$$1) \qquad [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4] = \Lambda[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$$

for the elements of Λ which contain the three coordinates and three angles of position. The algorithm described by Myers [4], also used during Phase I, was employed to find elements of Λ.

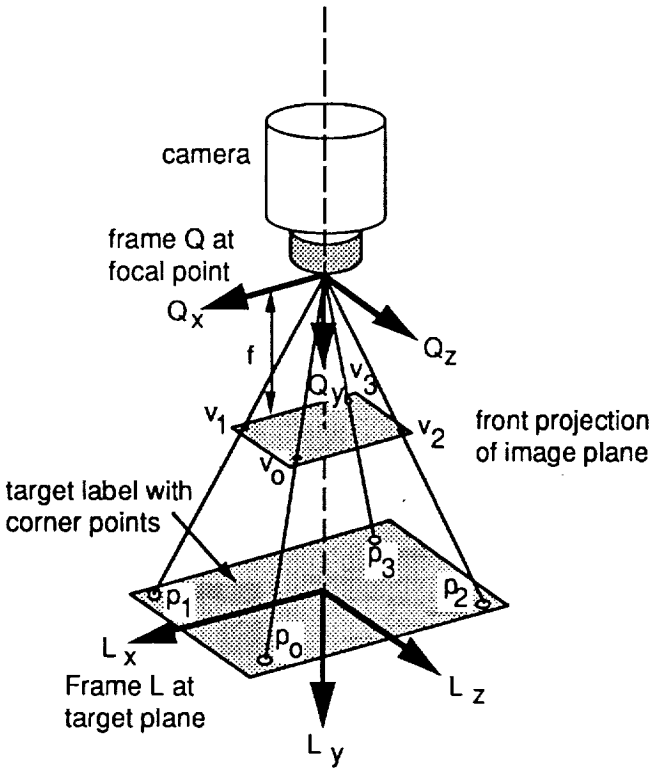

Figure 3. Schematic Illustration of Inverse Perspective Problem

During Phase II, the *inverse* perspective transform was recoded into WATCOM C. In addition to implementing the *inverse* transform, a *forward* perspective transform was also written to calculate the coordinates of the TCPs in the image plane of the camera for comparison with the

original values. The *forward* and *inverse* transforms taken together provide a means to test the accuracy of the corner point algorithms.

The program permits the user to specify the angular and translational displacements describing the relationship between coordinate frame **Q**, fixed at the focal point of the camera, and frame **L**, fixed in the plane of the TCPs. The Forward program implements the transformation from frame **L** to frame **Q**, then performs a perspective transformation to project the TCPs on the image plane of the camera. The coordinates of the TCPs in the image plane then become the input to the inverse perspective transform which finds the transformation matrix **A** relating frames **Q** and **L**. Then **A** can be solved to determine if its rotational and translational components match the angular and translational offsets input by the user.

In addition, a generalized package for performing coordinate transformations was developed. The package provides various operators for 4x4 homogeneous transformations. The function names along with a brief description of their functionality are listed below.

trsl() - sets translation components of a homogeneous matrix
vao() - sets the rotation part of a homogeneous matrix given the a and o vectors
rot() - sets the rotation part of a homogeneous matrix given a rotation angle about a vector
eul() - sets the rotation part of a homogeneous matrix given a set of euler angles
rpy() - sets the rotation part of a homogeneous matrix given roll, pitch, and yaw angles
noaTOeul() - sets euler angles from the rotation part of a homogeneous matrix
noaTOrpy() - sets roll, pitch, yaw angles from the rotation part of a homogeneous matrix
trident() - sets a homogeneous matrix to the identity transform
assigntr() - copies one homogeneous matrix into another
trmult() - computes the transform product R = T1*T2
vecmult() - computes the vector product r = T1*t2
inver() - computes the inverse of a homogeneous matrix
assignvect() - copies one vector into another
dot() - returns the real dot product of two vectors
smul() - multiplies a scalar with a vector
sdiv() - divides a scalar into a vector
cross() - computes the cross product of two vectors
unit() - reduces the magnitude of a vector to unity
norm() - computes norm of a vector

This package also proved particularly useful in determining the moves required of the robot for the demonstration.

Objective 6. Develop Shading-Tolerant Bar Code Algorithms

After determination of the four corner points, the line through the center of the bar code pattern can be calculated and used to step through the pixels, separating them by a fixed threshold into light (1) and dark (0) values in a binary chain. The binary chain can be thought of as a "calculated" video line through the center of the bar code pattern and is called the *pseudo-video* line. The most difficult part of the bar code discrimination involved distinguishing the narrow black and white bars in sequence. The sequence has low contrast, which means the gray level amplitude difference between black and white data peaks is small. Shading variations across the bar code could easily cause a fixed threshold to fall above or below some data peaks, thus causing them to drop out of the data stream, resulting in failure of the algorithm. The research

effort centered on finding improved dynamic thresholding methods for the binary separation of the *pseudo-video* line running through the center of the bar code. Several approaches to dynamic thresholding and data normalization were studied.

Gain and Offset
A gain and offset algorithm was implemented to increase the gray level swing from black to white (gain feature) and center the data about an appropriate value (offset feature). The result was a slight improvement (i.e. a decrease) in sensitivity to the threshold value, but no improvement was obtained for shaded images.

Noise Reduction Routine
A smoothing routine designed to minimize noise in the data without changing resolution was implemented in addition to the Gain and Offset routine above. The *noise routine* compared the gray level steps between data points with a chosen "noise" number. Data points with transitions less than the "noise" band were labeled the same as the previous point (0 or 1), and points with transitions greater than the "noise" band were labeled the opposite to the previous point. The results were mixed. Improvements could be achieved in bar code discrimination, but the algorithm was too sensitive to a particular "noise" number, which changed from image to image.

Second Order Least Squares Thresholding
In this method, a least-squares curve fit of second order was used to generate a variable, local threshold curve for binary division of the data points. Improvements were immediate, but various problems appeared. The white spaces at each end of the bar code had too great an effect on the ends of the curve, so the white data points at each end of the data set were omitted from the curve fit calculation. The calculated threshold curve was then interpolated back out to the ends of the data set. An effort was made to use the extreme data peaks as a reduced set for calculation of the threshold curve. Unfortunately, this caused a decrease in accuracy for some magnifications and was abandoned.

The least squares threshold curve approach with truncated ends has proven to be the most robust technique. It could accurately identify bar codes with significant shading across the image.

Objective 7. Develop Scene Segmentation Algorithms

Edge Detection Theory
The target identification program of Phase I used label images on a completely clear, noise-free background. The task in Phase II was to find the label image in a very noisy, high-contrast background. Considerable time was spent initially in Phase II reviewing the image processing literature for work related to edge enhancement and identification. The bar code and rectangular border, of course, consist of straight line segments that simplify the analysis.

Edge detection is typically preceded by filtering and thresholding. Although a classical linear low-pass filter can be sufficient, it usually blurs the edges. Median filtering is a nonlinear signal processing technique useful for image noise suppression. It has been shown [5] to preserve edges better than simple low-pass filtering. Another advantage is that it can be used iteratively to remove noise without degrading edge sharpness. In median filtering, the value at a given point is replaced by the median of the values of points within the neighborhood of the point.

Some other edge-preserving filters which are variations of the simple median filter include 'linear combination of medians," "weighted median filters," and "iterative median filters.' If the statistical properties of the noise can be determined, these techniques prove superior to simple

median filtering. White noise, impulse noise, and salt-and-pepper noise have been studied extensively by Justusson [5].

Thresholding is typically performed after filtering to discard background pixels. The pixels that are below (whiter than) the average pixel value are removed from the scene. This not only reduces the number of features which must be considered, but also makes the feature shape more closely resemble the actual object. In the best scenario, the gray level histogram of the image will display two peaks (bimodal). The image can then be segmented using the pixel value that represents the minimum between the two peaks. In cases where the histogram is not bimodal, the image is divided into smaller images and a threshold is assigned based on the interpolation of the local thresholds found for the nearby smaller images (Chow-Kaneko technique [6]).

There are numerous methods available for edge detection. For general didactic value, some are described below. Levialdi [7] classifies the various methods of edge detection as follows:

Local Methods
One local method uses a gradient operator $\Delta f(x,y) = (\partial f/\partial x + \partial f/\partial y)$ whose magnitude is given by

$$1) \quad | f(x,y) | = (\partial f/\partial x)^2 + (\partial f/\partial y)^2 \ ,$$

and the orientation is given by

$$2) \quad \emptyset = \tan^{-1}\{(\partial f/\partial x) / (\partial f/\partial y)\} \cdot$$

The gradient orientation is defined as the direction of maximum gray level change measured over a small area of pixels. It is the local direction of steepest descent or ascent on the intensity surface. Most preferred are the Sobel, Roberts, Kirsch, Compass, and Prewitt gradient operators using the largest acceptable window area.

Since edge determination is based on the gray level difference between neighboring regions, image elements will be extracted which do not lie on an edge. Several studies [8,9] have compared the performance of different types of local operators for visual images; however, performance appears to be image specific.

Regional Methods
Regional methods use a circular neighborhood such as Hueckel's operator [6] which involve solving a functional. Though these methods exhibit good noise immunity and are orientation invariant, the computational cost is heavy. Further approximations of this method have been developed to reduce the computational cost.

Global Methods
A linear shift-invariant spatial filtering operation can be performed on the image to minimize the mean square estimation error [6]. Such methods have proven very efficient over a wide range of images.

Sequential Methods
Two different sequential methods based on raster tracking and omnidirectional tracking are discussed by Rosenfeld and Kak [10]. In the raster tracking method, the image is analyzed by scanning rows in the manner of a TV raster. This method suffers from the disadvantage that the results depend upon the orientation of the raster and the direction in which it is scanned. Raster

tracking is more difficult for oblique curves. This method could be made more efficient by scanning in both directions, but would have additional computational cost.

Dynamic Programming Methods
Bellman's dynamic programming techniques can be applied to edge detection in images to find what is termed the "best boundary" [8]. A criteria often used is the weighted sum of high cumulative edge strength and low cumulative curvature [9]. Another method is to use a sequence of thresholds in the vicinity of a pixel having an optimum value from a gray level histogram to separate stable regions that demonstrate only slight variations on application of the thresholds. Heuristic methods can be more efficient than dynamic programming methods; however, dynamic programming builds paths efficiently from multiple starting points, which may be useful in some applications.

Relaxation Methods
The sequential methods discussed already cannot be speeded up by parallel processing techniques since their results depend upon the order in which the points are examined. Relaxation methods consist of making probabilistic decisions regarding classification at each point in parallel while updating the decision iteratively based on decisions made at the previous iteration at neighboring points. Unlike sequential methods, relaxation method is order-independent and hence can be made much faster by parallel processing [6].

Scene Segmentation
In order to separate a specific image from the background clutter, segmentation must be performed. This is basically a method of dividing the image field into subsets by assigning each element to a class depending upon the pixel value. There are several different techniques by which this can be accomplished. In one method, the Sobel direction operator is applied to the image after median filtering and thresholding in order to obtain the Sobel angles at the pixel points. Each pixel is then allocated to a different range of angles based on an "overlapping partitioning" method by Burns [6]. This partitioning scheme avoids overmerging problems as the partition size becomes smaller. For example, the first partition can be defined with a zero degree center with each partition segmenting in a 20 degree range. The partition is then rotated by 10 degrees, allowing overlapping . The segmentation is carried out by labelling the absolute angles with numbers that represent the partitions rather than a single value. This enables the pixels to be grouped into one region built by a "region growing" algorithm.

Experimental Efforts - Thresholding and Edge Detection
Image-Pro II image processing software was installed on the PC in order to examine various image processing algorithms. Image Pro II supports several standard processing techniques such as contrast enhancement (including sliding and stretching), spatial filtering (including both convolution and nonconvolution filters), histogram equalization, contouring, thresholding, and various mathematical image combination operators.

Using Image-Pro II, experiments were conducted with several high pass filter and edge detection algorithms previously proposed to identify and segment the target label from a "busy" background image. In order to simulate such background images, labels were copied onto transparencies and superimposed on photographs. The photographs were of various qualities, ranging from glossy to nonglossy and from high to low contrast. The busiest backgrounds, however, came from wrapping the ORU models with aluminized film. Every crinkle in the film creates a high contrast line or contour.

Each of the following algorithms, in isolation and in various combinations, was tested on the images: high pass filtering, edge detection (including Roberts, Sobel, and Laplace), median filtering, and several contrast equalization techniques. It was found that high pass filtering is of little use in isolating the label from the background. Although the edges of the bars and border

are enhanced, high pass filtering also enhances the "salt and pepper" noise in the background, producing an image which appears grainy. Median filtering, which is often used prior to edge enhancement, does little to improve bar code discrimination.

Of the various edge detectors, the Sobel operator appeared to be the most robust in enhancing the edges of the bars and borders, with little sensitivity to edge orientation. However, application of the Sobel on the raw image resulted in the enhancement of all straight edges in the image. A means to discriminate between background and foreground was sought.

Thresholding was examined as a means to minimize the number of pixels processed by the edge detector. It was found that pixel intensities of the bar code were located in the 0-50 grey scale band, on a scale of 0 (black) to 255 (white). Therefore, by simply thresholding the raw image at 50 prior to edge detection, a large portion of the unwanted pixels in the image were eliminated. Alternatively, an intensity histogram can be performed on the entire raw image in order to more carefully choose the threshold value.

In summary, pre-filtering the raw image by thresholding followed by the Sobel convolution filter provided an acceptable prelude for discrimination between a target label and a "busy" background. The threshold value may also be chosen dynamically from an intensity histogram analysis.

Connected Components

The image remaining after the pre-processing reveals interconnected series of straight lines, some of which are associated with the label, and some of which are associated with other objects in the image. A connected component analysis is required to indicate which series of lines are interconnected and should therefore be considered as components of the same object in the visual scene. Objects can then be discriminated based on the statistical properties of their components.

A connected component routine was implemented in C. The routine is similar in design to those developed at NIST in the mid 1980s. The routine consists of several distinct and functionally independent modules.

The first module performs run-length coding of the binary, thresholded image. Each row is scanned to locate the pixel address of transitions from high-to-low and from low-to-high. The result is a series of pixel strings marked by the pixel addresses where the string begins and ends. The resulting image is generally much more compact than the image preceding run-length coding.

It is assumed that the operator of the manipulator will not signal for autonomous operation unless a label is clearly visible and completely contained within the field of view. However, a border check was included to discard all pixel strings that extend to the edges of the image.

A second pass is performed to group each of the row pixel strings into a "connected component." Connectedness is determined by checking pixel strings immediately above or below another string. Pixel strings which touch only across diagonal pixels are not considered as connected. Since only straight lines are sought, this definition of connectedness provides a reasonable means to further eliminate pixel strings from consideration. The end result of the connected components function is a set of boundaries enclosing separate objects, one or more of which represent label boundaries.

Objective 8. Develop A Locally Resident CAD Data Base

The information encoded in the target label essentially serves as a pointer into a computer data base that characterizes each object. The purpose of the data base is to provide special information about the object after it is identified from the bar code. The fields for each data base entry consist of:

(1) **object identification number** - a number classification system that uniquely identifies all ORUs.
(2) **object name** - an ASCII character string associated with the object ID number that provides a common name for each ORU.
(3) **grasp location** - a homogeneous transformation matrix that relates the location of the robot connector to the target label.
(4) **approach location** - a homogeneous transformation matrix that relates a pose relative to the connector through which the robot end-effector must pass in preparation for grasp.
(5) **object height** - a homogeneous transformation matrix that relates the target label relative to the base reference plane of the object.

Software modules were created to edit and examine the database, to convert to/from position/angle and homogeneous representations, and to efficiently multiply homogeneous matrices.

A more complete database could include an extensive graphic model of each object suitable for computer rendering. Following identification of the object, and determination of its position and orientation relative to the viewing camera, the computer model could be graphically overlayed on the camera image. Such an overlay would serve to verify that the object has been properly identified, and could be used to highlight regions of critical interest on the object for further analysis.

After an extensive survey in Phase II, no CAD software was found that provided sufficient hooks to create an object model and integrate it efficiently into the system software.

Optical Systems Test and Calibration

Objective 9. Develop Specifications and Procedures to Define camera Characteristics. Design and Build an Optical Illuminator to Measure Lens, camera, and Image Sensor Parameters

Video Camera Considerations for Robot Vision Systems
Advances in the digital processing of visual images have extended the accuracy and precision of measurements of object sizes, distances, and orientations. As the demands for higher precision measurements continue, the characteristics of the imaging device become increasingly critical to the overall system accuracy. This area of the project concentrates on video camera imaging systems, and particularly focuses on cameras employing image tubes and CCD sensors.

Image tube cameras have been available since the 1930's and represent a mature technology based on electron beam scanning. There are a variety of photosensitive targets for visual and infrared viewing ranges. CCD cameras use solid state arrays and although they have only recently achieved the quality and resolution needed for vision systems, they are the most rapidly growing vision imaging system. CCD cameras are much more compact and mechanically rugged than image tube cameras. They have accurate, fixed target geometries, and require less

power. Image tube cameras, however, are less expensive, have higher resolution, and are generally more radiation resistant than CCD cameras. One of the goals of this research project is the testing of both types of cameras to determine their relative strengths and weaknesses for imaging and vision tasks in a space environment.

Because the two technologies are quite different in the way they produce images, they will be dealt with separately. The term "camera" usually means the complete imaging system, including the lens. Since the camera lens is a key contributor to geometric distortion and shading in either camera type, its contributions will be separated out.

Image Tube Cameras

The quality of a camera employing electron beam scanning tubes is determined by three areas of technology - the image tube itself, the camera deflection circuits, and the video amplifier circuits. The image tube in a camera will be generically referred to as a vidicon tube, although there are many different types with different target structures and characteristics. Figure 4 illustrates a typical tube.

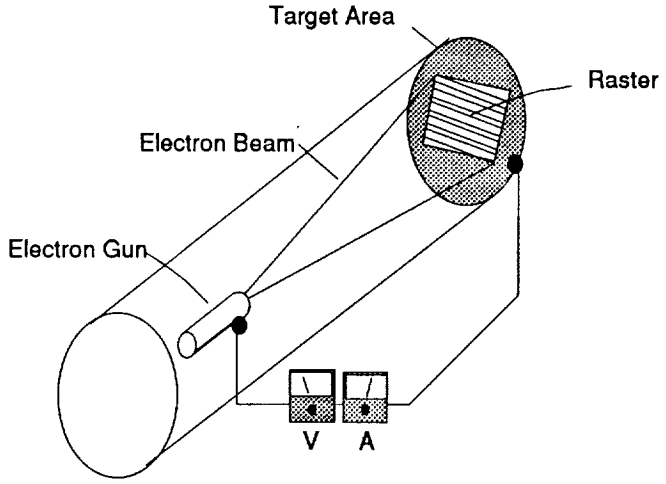

Figure 4. Structure of Camera Image Tube

An electron gun generates a beam of electrons that are focussed and accelerated toward the target of the tube. The beam is deflected by magnetic fields to trace over a rectangular area, or *raster.* The target is a thin film that conducts electricity in lighted areas, but does not conduct in dark areas. In the nonconducting areas, the beam initially charges the surface until the local charge density is enough to prevent the beam from landing. When the beam strikes an illuminated area with a lower charge density, a current flows through the circuit. The relationship between the output current, I, and the incident light flux, Φ, is

$$3) \qquad I = S\,\Phi^{\gamma},$$

where S is the sensitivity of the target and γ is a linearity factor. Both S and γ are considered constants for each tube, but they may be functions of the wavelength of the incident light, the ambient temperature, and other operating conditions that can change with time. Typical target currents are in the 100-500 nA range. It is desired that an image device be perfectly linear in response - that is, $\gamma = 1$. Unfortunately, vidicon tubes have gammas that vary. A study of 31 new Plumbicon ® tubes showed gamma values from 0.925 to 1.075. Equation 3 also applies to video display tubes, which are not linear in brightness with electron beam current. Gamma correction circuits are common in vidicon cameras and display monitors.

The signal current can also be represented as

$$4) \qquad I = \frac{DQ}{Dt},$$

where Q is the charge transferred in each interval of time. This charge is the product of the area of the beam, A, and the charge density per unit area, J, on the target,

$$5) \qquad Q = J\,A\,.$$

Combining eqns. 4) and 5) yields the relation

$$6) \qquad I = A\frac{dJ}{dt} + J\frac{dA}{dt},$$

where dA/dt represents the area swept by the beam per unit time interval. If a circular beam of area A_o and diameter D moves a distance L, then the swept area is A_o + LD, and the time derivative of this quantity yields

$$7) \qquad \frac{dA}{dt} = 0 + D\,\frac{dL}{dt} = DV,$$

which is the product of the beam diameter and its velocity across the surface. The beam velocity could be broken into orthogonal components obeying the relation

$$8) \qquad V = \sqrt{V_x^2 + V_y^2},$$

but this will not be necessary for rectangular patterns. Equation 6) can now be written as

$$9) \qquad I = J\,D\,V + A\frac{dJ}{dt}\,.$$

This relation says that the instantaneous signal current is made up of two terms. The first term applies to the normal operating mode where the charge transfer is the result of a beam of diameter D, swept with velocity V over a region of charge density J. The second term gives the additional contribution to the signal current that occurs when the charge density changes within the area swept by the beam. A variation in illumination across the target can cause this condition.

Since any experimental attempt to measure camera and tube parameters requires a stable and uniform light source, the second term in eqn. 9) can be set to zero, leaving

$$10) \qquad I = J\,D\,V\,.$$

Shading is defined as a change in the signal as a function of position on the target, so it is instructive to examine what happens if eqn. 10) is differentiated with respect to the position variable, z,

$$11) \qquad \frac{dI}{dz} = J\,D\frac{dV}{dz} + J\frac{dD}{dz}V + \frac{dJ}{dz}D\,V\,.$$

For uniform illumination with a stable light source, a perfect camera/tube system would have perfect shading and eqn. 11) would be identically zero. The terms on the right side of 11) therefore define three sources of error in the target current as a function of position.

The dV/dz term is called the *geometric distortion* because it represents a variation in scan velocity with position.

The dD/dz term is the *dynamic focus* error caused by a change in the beam diameter as a function of position. Most CRT displays incorporate dynamic focus correction circuits, but the low deflection angle of image tubes makes this unnecessary. One can see that the first two terms are camera errors and not tube errors.

The dJ/dz term is the *tube shading* error caused by a change in current density of the target as a function of position. Since one of the assumptions is uniform illumination, the density change can only be caused by a variation in the target sensitivity over the image area It is possible to construct illuminators that achieve optical uniformity greater than 98% over the target area, so that under proper test conditions, the third term can represent only target sensitivity variations.

The parameters that must be set for a vidicon camera to operate properly include

beam current*
beam focus*
beam centering*
blanking widths*
video gain and offset*
horizontal sweep linearity*
raster size and centering.

Items marked with an * indicate factory settings that are usually not changed in the field. The field test parameters that determine the quality of a camera include

shading,
geometric distortion,
spurious signals,
bandwidth,
signal to noise ratio,
horizontal/vertical resolution, and
temperature/voltage stability.

Measurement of temperature and voltage stability requires an extensive laboratory facility with environmental chambers, which is beyond the scope of this program.

Optical Illuminator

The tests for shading, spurious signals, and geometric distortion require an optical illuminator which presents a "perfect" image to the target of the image tube. Several images are required, including a blank white field with uniform shading bounded by a black edge of the proper raster size, a geometric target with lines or circles, and a resolution target with lines of decreasing width. An illuminator was constructed specifically for this project because commercial units cannot achieve the uniform illumination needed. Commercial units have illumination uniformity in the 4% range. The illuminator constructed for this program is about than half that value. Typical camera specifications may allow 8% shading.

An optical bench and mounts were purchased for holding the lamp, the shading corrector, the image target, and the projection lens. However the components could not be aligned on a straight axis due to poor construction of the mounts and it was decided to design and fabricate all the components for a new optical test bed. The holders, rods, translators, and base plate were all fabricated in the lab shop.

Figure 5 illustrates the camera mount and illuminator structure. The light from the lamp goes through a field limiting aperture and into the shading corrector. At the far end of the corrector is the target holder, which consists of a glass photographic plate with image patterns on it, mounted on a horizontal slide mechanism. The target images are 100:1 photographic reductions of standard video resolution and linearity charts. The projection lens creates an aerial image from the pattern on the glass plate. A metal enclosure open on the camera end minimizes the amount of stray light from the room and lamp that hit the camera target. An x-y-z translation table provides accurate positioning of the camera image target in the aerial image plane. The aerial image was calibrated for shading and size with an new Instaspec 512 linear CCD detector made by Oriel Corp. Unfortunately, the detector electronics were defective and had to be returned for repairs, further delaying the program. The detector head includes a linear array of 512 pixels with 50µm spacing and is calibrated for sensitivity traceable to the NIST. The detector was mounted in the camera position with the array surface in the aerial image plane.

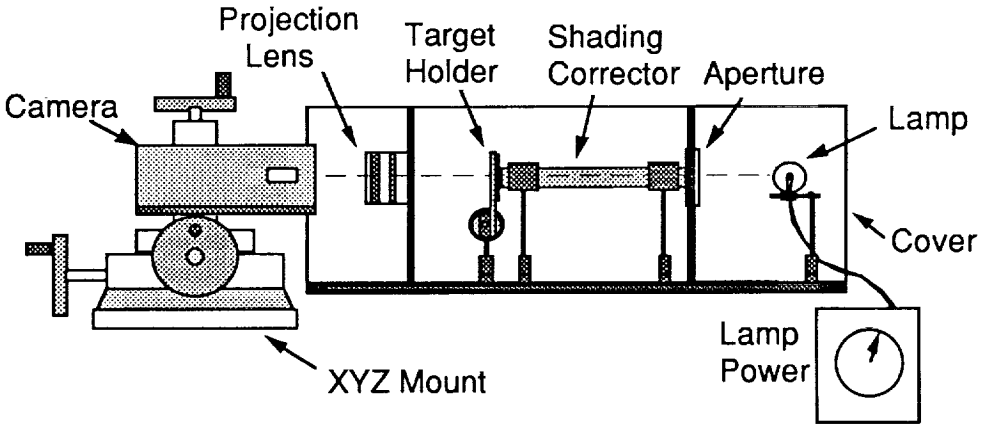

Figure 5. Camera Test Illuminator and Mount

The image size for a 2/3 inch camera format is 8.8 x 6.6 mm. The results of the illuminator testing show shading uniformity of 2± 0.5% and an image border size of 8.8 x 6.6 ± 0.05mm.

Definitions and Equations for Shading and Spurious Signals
Figure 6. shows the black and white video levels on a midfield horizontal scan line resulting from an optical image with a dark border around the outside and a broad spot past midfield. The video signal amplitude is the black to white voltage swing of the signal and results from the gain setting of the amplifiers for a specific target current. The target current results from the intensity of illumination on the target.

The video offset is the black level to ground voltage, which is adjusted by a control called "black level", "pedestal", "setup", or some other name. The control sets the reference level of black above ground.

Since it is usually more accurate to measure all signals with respect to ground, the following quantities are defined

B = black level voltage (black to ground voltage),
W = white level voltage (white to ground voltage), and
V = W - B = video signal voltage (white to black voltage)

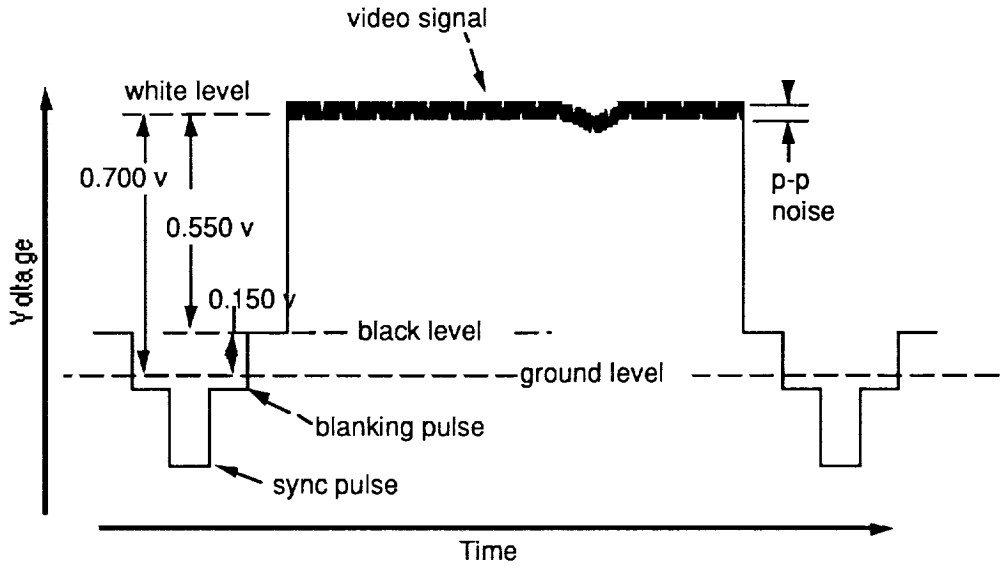

Figure 6. A Video Line Showing Signal Levels

Signal stabilities are determined by measuring B and W for the same data point under a series of test conditions, usually variations in temperature and line voltage, and computing the video gain stability

12) $DV = V_{max} - V_{min}$,

and the offset stability

13) $DB = B_{max} - B_{min}$.

Signal stabilities are determined by *voltage* variations in the signal. Testing the signal and geometric stabilities of the two cameras was not within the scope of this program, however, the definitions and procedures are included here for completeness.

Shading Definition and Measurement
Shading is the variation in target sensitivity as a function of position. It is probably the most difficult parameter to measure accurately of all video tests. Ideally, the test requires perfect illumination and linear video amplifiers with no spurious signals or deflection distortions. In other words, the camera circuits and optical pattern should contribute nothing to the signal variation. All image tube calibrations should be performed in a calibrated camera, and all camera tests should be performed with a calibrated tube.

In order to measure shading and spurious signals, the video level must first be set to a standard value, V_{avg} for a blank, white image. If V_{max} represents the highest video signal (black to peak

white) in the field, and V_{min} represents the lowest video signal in the field, then the shading over the raster is

$$14) \quad \%S = \frac{V_{max} - V_{min}}{V_{avg}} 100,$$

where V_{avg} is the standard reference value, set up to be the average signal height in the center of the raster. The shading measured by this method is called *full field shading. Horizontal* shading can be defined for values taken from a horizontal line. *Vertical* shading can be defined the same way with data from a vertical line down the raster. With a computerized frame grabber, the histogram peak defines Vavg, and either side of the width can be used to determine Vmax and Vmin. In this case, the noise width must be subtracted from the histogram width to accurately represent the true values of Vmax and Vmin . Field shading requirements can range from 8%, which is typical, to 3-5% for high tolerance applications.

The measurement of spurious signals is the same as shading, with appropriate definitions for the the width and %S tolerances allowed. Spurious signals are frequently caused by amplifier leakage, feedback, and oscillations not related to the image surface. Spurious signal levels over 1% can cause highly visible artifacts.

Blemishes on the target caused by imperfections in the light sensitive coating can be specified according to position in the image, x and y width (time) and signal height (%S).

Geometric Distortion Definition and Measurement
Geometric distortion and raster size and centering stabilities are determined by *time* measurements of specific signal markers. Size and centering variations are measured from the time between two optical markers in the image, originating from vertical or horizontal lines in the image. Figure 7. illustrates a horizontal video line scanning over two vertical lines in the image.

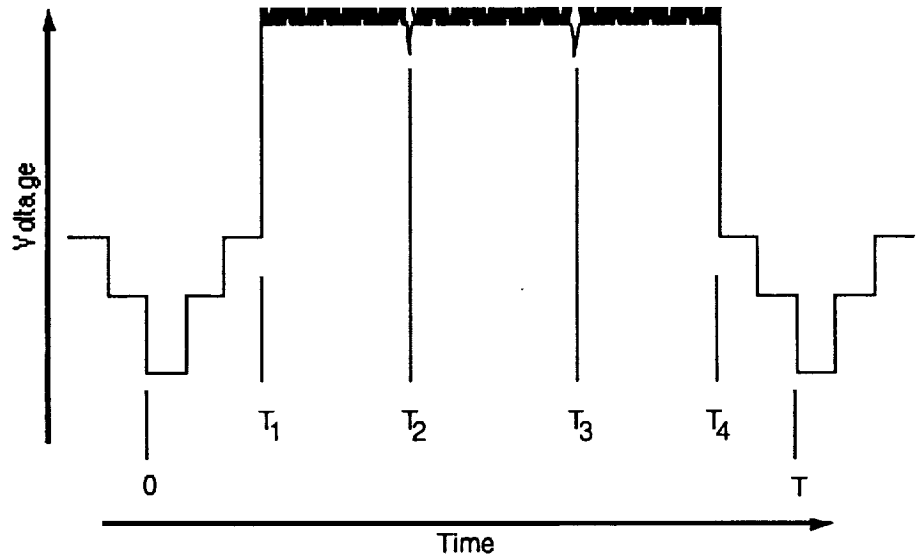

Figure 7. A Video Line with Time Markers for Geometric Measurements

If $T = T_3 - T_2$ is the time between two optical markers on a horizontal scan line, the change in the width of the raster over a set of measurements representing varying conditions is

15) $DT = T_{max} - T_{min}$,

and the percentage change is

16) $$\%T = \frac{DT}{T_{ref}} 100,$$

where $T_{ref} = \{T_3 - T_2\}_{ref}$ is a reference value taken at standard temperature and voltage. Raster size and centering stabilities are measured in a similar manner.

Geometric distortion is measured with a series of equally spaced optical markers across the image, both horizontally and vertically. The uniformity of spacing is measured and the extremum compared to the standard value. The use of the frame grabber card reduces the measurements to easy practice, but there are two noise contribution to the signal that must be corrected, the signal to noise ratio of the analog video signal, and the digitization tolerance of the frame grabber. The video analog noise value is steady and may be removed as a systematic error. The digitization error is random and can only be treated as a deviation tolerance.

Camera- Frame Grabber Interactions

Modern camera calibration measurements and image processing are performed with a frame grabber installed in a computer. The live video image seen directly from the camera on an analog monitor is not identical to the image seen on the computer monitor. The frame grabber clips the sides of the image, then transfers it to the computer VGA display card. The interactions between the camera, frame grabber, and VGA display card were analyzed to understand the asymmetry that appears between the two images.

The camera produces a 525 line, 30 Hz field, consisting of two 60 Hz scans of 262/263 lines each, interlaced 1:1. A single horizontal line has a period of approximately 62 µs, comprised of 52 µs of active data, and 10 µs of flyback time. The active window is defined by horizontal blanking pulses added to the video stream to mask nonuseful data. Likewise, the vertical scan has a 16.66 ms period, comprised of 14.6 ms of active scan time and 2 ms of flyback time. The active vertical window is defined by blanking pulses added to the video stream to mask nonuseful data.

The TARGA a/d converter runs continuously, digitizing data on the fly. This digitized stream includes all sync pulses, flyback intervals, etc. The output signal to the video display is this unprocessed digital stream, treated as though it were a simple analog signal. The digitized data stream is also sent to the memory buffer for storage. The maximum 512 x 512 pixel memory area is filled by selecting 512 pixels from each horizontal scan line, and selecting the number of vertical scan lines to be active. Since the data stream contains many more than 512 pixels per horizontal line, a delay timing window is set to choose the starting point on the line for centering the data. The digitized width comprises about 80% of the full horizontal scan width. The board can digitize up to 512 vertical lines out of 525, which will include portions of the blanking and flyback regions. Since there are only 480 active lines in a typical NTSC video field, the maximum usable video image consists of 512h x 480v pixels.

The VGA computer display board cannot display standard NTSC video signals, because the number of scan lines and timing pulses do not match NTSC standards. VGA formats include 800h x 600v and 640h x 480v. This means that the VGA image will be distorted from the original NTSC image. The distortion appears in the form of a compressed horizontal axis, which shortens the horizontal length of objects. While this distortion is not severe, it does affect the geometric perception inherent in designing object recognition software. A decision was made to display all pictorial images on the Targa monitor, and all the program commands and menus on the VGA display.

Results of Measurements
The illuminator was tested for shading and geometric distortion with the Oriel CCD line array detector. Table 1 shows the results

Table 1. Shading and Geometric Distortion Values

1. Illuminator *(from CCD linear array)*

Horizontal Shading		*Vertical Shading*	
top	1.70-2.00 %	right	1.19-0.49 %
center	1.17-1.83 %	center	1.18-0.98 %
bottom	1.10-1.80 %	left	1.57-0.88 %
Horizontal Distortion		*Vertical Distortion*	
top	0.570 %	right	0.000 %
center	0.000 %	center	0.000 %
bottom	0.570 %	left	0.760 %

2. Pulnix TM 545 Camera *(from Targa board)*

Field shading from histogram width = 5.88%

Horizontal Distortion		*Vertical Distortion*	
top	-0.160 %	right	0.000 %
center	0.000 %	center	0.000 %
bottom	0.000 %	left	0.210 %

Resolution - 225-250 lines horizontal mid-field

3. Lens and Pulnix TM 545 Camera *(from Targa board)*

Horizontal Distortion		*Vertical Distortion*	
top	0.114 %	right	1.667 %
center	0.000 %	center	0.000 %
bottom	-0.227%	left	-1.061%

Resolution - 225-250 lines horizontal midfield

4. Panasonic WV 1550 Camera *(from Targa board)*

Field shading from histogram width = 9.07%

Horizontal Distortion		*Vertical Distortion*	
top	0.170 %	right	0.230 %
center	0.000 %	center	0.000 %
bottom	0.350 %	left	0.930 %

Resolution - 250-275 lines horizontal midfield

The results indicate that the geometric distortion values of the cameras are negligible, with the systematic errors being the same order of magnitude as the readings. The camera lens brings geometric distortion up to about 1.7%, which is still very small. The only real distinctions are the slightly flatter shading of the CCD camera (5.88% vs 9.09%) and the higher resolution of the vidicon tube (250/275 lines vs 225/250 lines). The CCD camera was also more sensitive to light than the vidicon tube. However, the vidicon beam current, and video amplifier gain settings

were not checked. Visually, the vidicon picture appeared sharper than the CCD image, and sensitivity differences were adjusted with the lens iris.

Robot Application and Demonstration

Objective 10. Demonstrate Autonomous and Shared Autonomous Construction Tasks Representative of NASA Goals.

Robot End-Effector
During the course of this project, the priorities for NASA's robotics program changed. Through consultation with Dr. Del Jenstrom, and John Vranish of Goddard, the demonstration part of this project has been directed toward ORU replacement tasks. Toward this effort, an H-plate and a parallel jaw gripper were obtained from NASA GODDARD. Unfortunately, the Puma robot has a weight limit of five pounds and the gripper was too large and heavy to use. In order for the gripper to clear the H-plate and close within the notches, each finger must travel approximately 1.5 in. Most grippers within the payload constraints of the robot have throw ranges of at most 1.0 in. per finger.

As a compromise, the length of the H-plate was reduced to accommodate an available, pneumatically actuated, parallel jaw gripper with a 2.0 in. throw range. The commercial fingers were replaced with fingers designed from the specifications supplied by Goddard and fabricated in a local machine shop. The gripper system appears to be fairly robust in performance with respect to rotational misalignments (roll) of the H-plate in the plane of the H-plate; however, for rotational misalignments (pitch and yaw) out of the plane of the H-plate, tolerance in the parts allows some motion.

A bracket was designed and fabricated for attachment to the robot wrist flange, to which the gripper assembly and CCD camera were mounted. The bracket includes slots to accommodate adjustment of the gripper and camera relative to the center line of the wrist flange. The gripper is mounted along the flange center line and the camera 60 mm above the center line. The bracket also allows rotational adjustment of the camera along its yaw axis with respect to the gripper.

Figure 8 shows the Puma end effector with fingers, camera bracket, and camera.

Robot Interface
Four interfaces were considered between the PC host computer and the PUMA robot: 1) *Internal ALTER*, 2) *External ALTER*, 3) *SLAVE*, and 4) a servo level interface. The *ALTER* interfaces permit offsets from the current robot pose to be accepted from the host. The offsets are specified in a Cartesian tool reference frame. With *External ALTER*, a background task running under VAL monitors communication with the host and passes the desired position changes to the foreground control program. With *Internal ALTER*, the communication is handled directly by the foreground program. *Internal ALTER* requires a fixed communication rate; *External ALTER*, does not. The *SLAVE* interface accepts joint angle set points at a fixed 28 ms sampling rate. With both *Internal ALTER* and *SLAVE* it is incumbent upon the host to supply data at a constant, fixed rate. *External ALTER* is more cumbersome to implement but makes less demand on the host computer system.

A fourth interface possibility, replacing the robot servos, was rejected, and a design for an interface through *External ALTER* was developed.

Building on previous work done by Dr. Myers at Lord Corporation, a sixteen-bit bidirectional parallel interface with appropriate handshaking was designed and constructed. Driver software from an 8 bit interface developed for a previous task was rewritten for the 16 bit interface. The demonstrations were conducted open loop with the PC providing *relative* move commands to the robot.

The higher level software for implementing a set of commands to control the robot from the PC was designed, coded, and debugged. Both the low and high level drivers were coded in C on the PC side of the interface and in VAL on the robot side of the interface. Since the robot is controlled through *relative* move commands, the PC requires no knowledge of the absolute position of the robot, and, therefore, the command set is exclusively unidirectional from the PC to the robot. The command set consists of five instructions:

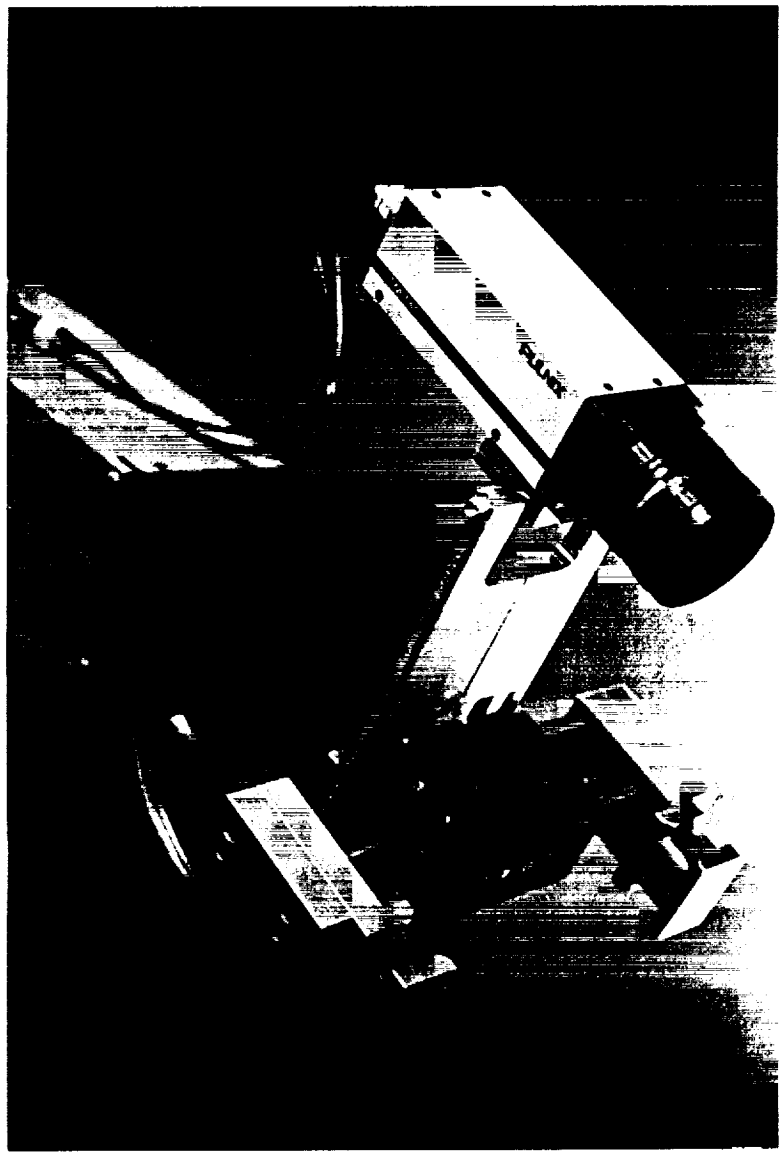

Figure 8. Robot End-Effector

1) TELEOP -- a mode in which the robot can be moved under operator control using the teach pendant until the operator desires autonomous operation.

2) MOVE_TO -- a mode in which the robot is commanded to *move to* a position relative to its present location. The relative offsets will typically be derived from the vision system.

3) MOVE_THRU -- a mode in which the robot is commanded to *move through* a relative position but not to stop at that location. This command is typically used to force the robot to approach the ORU handle from a specified direction in preparation for grasp.

4) OPEN -- open the gripper fingers.

5) CLOSE -- close the gripper fingers.

PC-Dataglove Interface
The VPL Dataglove was intended to serve as the master control device for the robot. It was anticipated that the operator would move the robot in a master-slave mode until the ORU target was within the field of view of the camera, at which point the operator would "signal" the robot vision system to perform the ORU acquisition autonomously. The Polhemus on the glove provides an ideal sensor for tracking the operator's hand position, and the finger position sensors permit the operator to "signal" the vision system through finger gestures.

Dataglove software was written to complement sample driver software written in Turbo C supplied by the glove manufacturer. The software was enhanced to provide the functionality desired for this project. The software was then ported to Microsoft QuickC and finally to WATCOM C for use in the current project software environment.

However, the limited bandwidth for communications between the PC and the robot does not allow the Dataglove to provide smooth control in the movement in the robot arm. It became necessary to position the robot with the teach pendant that, although limited in its ability to move the arm in any direction not aligned with the tool or world coordinate frames, does provide for smooth and controllable movement of the arm.

Demonstration hardware
The demonstration chosen as most appropriate for this NASA program was ORU replacement simulation. Two 12 inch square by 2 inch deep metal pans were constructed as receptacles, and a matching pan was fabricated to fit into the other two (Figure 9). The modified H-plate was mounted on a cylinder extending several inches above the surface of the male pan. Target labels were mounted on the male pan and the receptacles The bar code titles identify them as *010*, *020*, and *030*. All the pans were wrapped in aluminized mylar film to maximize the busy background image and simulate actual conditions.

The demonstration incorporated all the features of target recognition, target location, bar code reading, data base descriptions of the objects, and ORU acquisition and docking using the robot. In the course of implementing the demonstration, several features of a robust operator-robot interface became apparent, based upon the concept of co-autonomy, which allows the human operator as much or as little control as desired.

The sequence of events in the demonstration were:

1. Operator positions the robot camera to see the target label on the ORU box.
2. The vision system reads the label, determines position and orientation, identifies the object, then accesses the proper data base for geometric information about the location of the H-plate and the perimeter of the ORU box.
3. The operator positions the robot camera to see the target label on the desired ORU receptacle box.
4. The vision system reads the label, determines position and orientation, identifies the object, then accesses the proper data base for geometric information about the perimeter of the ORU receptacle.
5. The operator indicates acceptance of the identification of the two components and commands the robot to mate the two objects.
6. The robot moves to an approach position specified in the data base for the ORU box and repeats step 2 for a final approach to grasp the H-plate.
7. The robot grasps the ORU H-plate, moves the ORU box to an approach position specified in the data base of the ORU receptacle, then mates the two pieces.

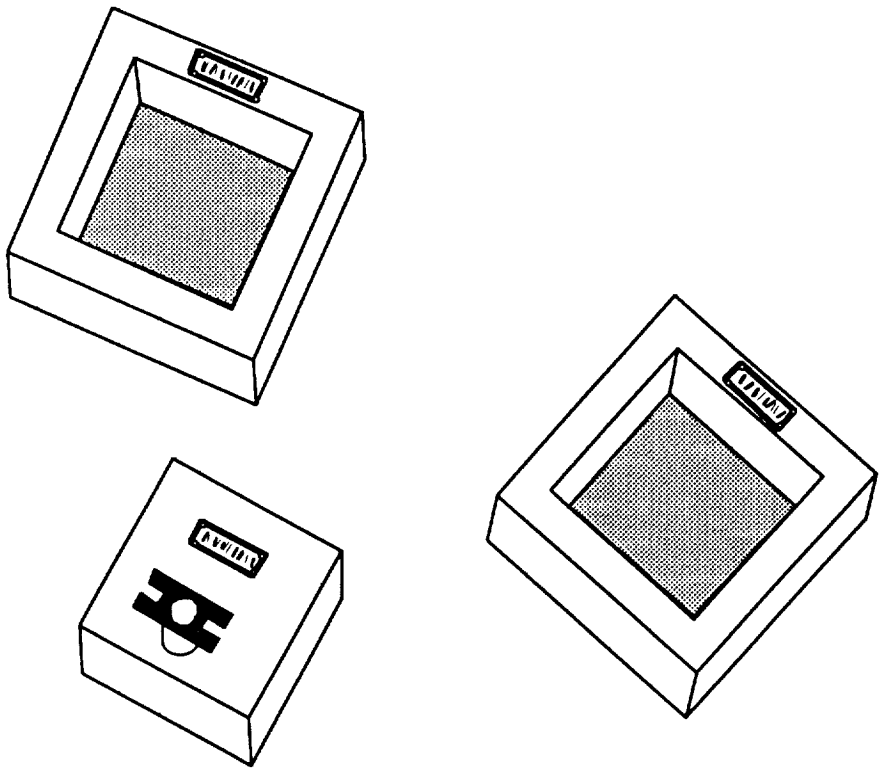

Figure 9. Simulated ORU Components for Robot Demonstration

Co-autonomy
The vision system software has operator entry points at every major action. If the program fails to properly execute a portion, the operator can intervene and type in the correct information, or direct the recalculation of an item.

If the program sees no anomalies, it runs automatically to completion, and no operator intervention is needed. The operator can choose the level of interaction with the program:

1) every action (report and wait for acceptance)
2) major actions (report and wait for acceptance)
3) none (report failures only).

This procedure guarantees a very high level of success and minimizes the operator's time for "normal" situations, but allows decisive interaction for problem cases. In summary, the co-autonomy program is

- Extremely robust - the task will be completed,
- very flexible - allows operator control at several different levels, and
- Allows programs to be put into service even while improvements to algorithms are continuing.

A video tape of the demonstration is included with this report.

Camera performance
The CCD camera was used throughout the software development, and the vidicon camera substituted at the end for a demonstration run. Generally, the CCD camera had slightly lower shading and geometric distortion. It was more sensitive to light, but had lower resolution. The same lens was used on both cameras.

The vidicon camera trials were not successful. The program would fail in the connected components module and be unable to identify the border of the label. This failure appears to be the result of the higher resolution of the vidicon camera. Throughout the software, the image data is scanned in units of three or more pixel jumps to save time. For the higher resolution camera, three pixel jumps were enough to cause lines to break, loosing continuity. The CCD camera would see the pixels as connected. Such programming choices distinguish the characteristics of one camera over the other. None of the other characteristics of the two cameras appears to distinguish them for robotic applications. The higher resolution of the vidicon camera means it should be successful at farther distances from the label than the CCD camera. (See Future Work.)

Summary
This research program has successfully demonstrated a new target label architecture that allows a microcomputer to determine the position, orientation, and identity of an object. It contains a CAD-like data base with specific geometric information about the object for approach, grasping, and docking maneuvers.

Successful demonstrations have been performed selecting and docking an ORU box with either of two ORU receptacles.

Small, but significant differences have been seen in the two camera types used in the program, and vision sensitive program elements have been identified.

The software has been formatted into a new co-autonomy system which provides various levels of operator interaction, and promises to allow effective application of telerobotic robotic systems while code improvements are continuing.

Future Work

The developments of this research program have opened the door to many new concepts and practical improvements in robot vision systems. Many software code improvements are possible now that the concept of a co-autonomous operating system has been developed.

New transforms
M. K. Abidi at the Univ. of Tennessee has recently published another solution to the inverse perspective transform problem. His solution offers two potential advantages over the Univ. of Md. algorithm [3] that was used in this program. First, the solution is overdetermined, offering the potential of decreased sensitivity to noise. Second, the solution is independent of camera focal length. The major disadvantage is that the solution is more intensive computationally. This disadvantage may not be severe relative to the other algorithms required to detect the target. Telephone conversations have been held with Abidi, and he has offered to share his software with TRDC.

Focus Variations
The current methods for the inverse transform assumes a very simple camera optical system based on the classical pin -hole camera which has infinite depth of focus In the real world, changing the focal length of the camera lens will be a necessity for future applications. The

effect of a variable focus on vision algorithms has not been studied in detail. The relationship of Abidi's solution to practical variable focus camera systems has not been determined.

This research program focussed on acquiring and analyzing one target label at a time. For docking, two targets are acquired sequentially, the coordinates stored, and the move commands executed. In this scenario, the camera is located on the wrist of the robot and moves with the gripper. When there is nothing in the gripper, the arm can carry the camera around to view different scenes. When a target object is in the gripper, the target fills the camera view, and it cannot see any other objects. Docking and manipulations are then executed blindly from stored information.

Parallel Target Acquisition
To increase the utility of the robot vision system, a parallel target acquisition mode is proposed. In this mode the camera is located away from the gripper where it can see a larger scene containing several target labels, In this mode, docking is dynamic, with continuous data capture, calculation, and position updating of both targets as they come together. The camera location and mounting method, whether fixed or movable must be determined.

Zoom Lenses
In the parallel acquisition mode, the camera covers a larger "sight volume" (analogous to "work volume"), with a larger field of view covering several target labels, The use of a zoom lens will allow closer views for identification of individual targets, followed by more distant views for maneuvering and docking. Calibration of the varying focal lengths of the lens and testing the algorithms over the full "site volume" become the major tasks in this area. This work requires a motorized lens that is controllable by the computer, and calibrated for lens position as a function of motor position. Most autofocus systems are self-contained, which operate autonomously to peak the data in the image without any feedback as to the mechanical position of the lens. They cannot be driven digitally by a computer signal. We have found a company who has devised a simple digitally controlled motor with a belt drive coupling to a lens focus ring. The concept appears adaptable to any lens.

Window Masks
An important side benefit of the program is the combination of the scene segmentation/label detection capability and the data base description of objects. These two features form a powerful new tool for the reading of control panels gauges, switches, valve handles, and other visual displays.

An application could work like this: A target label is placed on an electrical control panel with various switches, gauges, and other readouts. The target label is identified by the vision system and the object name determined in the standard manner. Stored in memory is a mask pattern that is superimposed over the scene with "holes" that fit over the gauges, switches, or other items for analysis. Since the angle, orientation, and identification of the control panel is known from the target label, the orientation and size of the mask is also determined, allowing great flexibility in the use of the feature. The analysis of gauge readings is directed to the data in each of the windows in the mask, giving instant scene segmentation. For example, the angle of a meter in the image (its pose) is already known from the label data, so that parallax errors can be corrected immediately. Decoding a needle position in a meter, or switch handle position, or digital display characters is then greatly simplified.

Exactly the same technology could be applied to be the remote inspection of mechanical control panels, or individual components such as valve handles, which are scattered throughout hazardous areas, such as nuclear reactor facilities or space environments.

References

[1] E. Person and K-S Fu, "Shape Discrimination Using Fourier Descriptors," **IEEE Trans on Systems, Man, & Cybernetics**, vol. smc-12, no. 3, Mar 1977.

[2] T.R. Crimmins, "A Complete Set of Fourier Descriptors for Two-Dimensional Shapes," **IEEE Trans on Systems, Man, & Cybernetics**, vol. smc-7, no. 6, Dec 1982.

[3] Y.H. Yung, P-S, Shu, D. Harwood, "Passive ranging to Known Planar Point Sets," **IEEE Conf on Robotics and Automation**, St. Louis, MO, Mar 25-28, 1985.

[4] D. R. Myers, M. Juberts, and S. A. Leake, "Enhanced Telemanipulator Operation Using a Passive Vision System", Proc. of IEEE Cong. on Systems, Man, and Cybernetics, Tuscon, AZ, Nov. 1985

[5] B.I.Justusson, "Median Filtering: Statistical Properties", **Two Dimensional Digital Signal Processing II**, ed. by T.S.Huang, Springer Verlag, 1981.

[6] Ballard and Brown, **Computer Vision**, Prentice Hall, 1982.

[7] S.Levialdi, "Finding the Edge", ed.by J.C.Simon and R.M.Haralick, **Digital Image Processing**, 105-148, 1981.

[8] Peli and Malah, "A Study of Edge Detection Algorithms", **Computer Graphics and Image Processing,** vol. 20, pp. 1-21, 1982.

[9] Abdou and Pratt, "Quantitative Design and Evaluation of Enhancement/thresholding Edge Detectors", **Proc.IEEE,** vol. 67, pp 753-763, 1979.

[10] Rosenfeld and Kak, **Digital Picture Processing**, vol.2, Academic Press, 1981.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
Final Report		

4. Title and Subtitle

Telerobotic Rendezvous and Docking
Vision System Architecture

5. Report Date

January 17, 1992

6. Performing Organization Code

287

7. Author(s)

Ben Gravely, Ph.D., Principal Investigator
Donald Myers, Ph.D.
David Moody

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

Triangle Research and Development Corporation
P.O. Box 12696
Research Triangle Park, NC 27709-2696

11. Contract or Grant No.

Contract Number NAS5-30709

13. Type of Report and Period Covered

Final - 28 months

12. Sponsoring Agency Name and Address

Goddard Space Flight Center
Greenbelt, MD

14. Sponsoring Agency Code

NASA/GSFC

15. Supplementary Notes

16. Abstract

This research program has successfully demonstrated a new target label architecture that allows a microcomputer to determine the position, orientation, and identity of an object. It contains a CAD-like data base with specific geometric information about the object for approach, grasping, and docking maneuvers. Successful demonstrations have been performed selecting and docking an ORU box with either of two ORU receptacles. Small, but significant differences have been seen in the two camera types used in the program, and camera sensitive program elements have been identified. The software has been formatted into a new co-autonomy system which provides various levels of operator interaction, and promises to allow effective application of telerobotic robotic systems while code improvements are continuing.

17. Key Words (Suggested by Author(s))

Robotics, Vision, Docking, Targets

18. Distribution Statement

1 copy Patrick Logan
5 John Vranish
2 NASA Scientific & Tech. Information Facility

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
		32	

NASA FORM 1626 OCT 86