78

188124

N92-10491

# Monovision Techniques for Telerobots

P.W. Goode and K. Cornils
NASA Langley Research Center
Hampton, VA 23665-5225

## 1. Abstract

The primary task of the vision sensor in a telerobotic system is to provide information about the position of the system's effector relative to objects of interest in its environment. The subtasks required to perform the primary task include image segmentation, object recognition, and object location and orientation in some coordinate system. The accomplishment of the vision task requires the appropriate processing tools and the system methodology to effectively apply the tools to the subtasks. ~~This paper describes~~ the functional structure of the telerobotic vision system used in the Langley Research Center's ~~(LaRC)~~ Intelligent Systems Research Laboratory ~~(ISRL)~~ ~~and discusses~~ two monovision techniques for accomplishing the vision subtasks.

## 2. Introduction

The telerobotic vision research objective is to adapt, develop, and evaluate noncontact sensing techniques to recognize and determine the location of objects in 3-space. To meet the objective, five goals have been established: (1) the techniques should be minimally complex in both hardware and software; (2) be generally applicable to a wide range of tasks; (3) require minimal or no alteration or premarking of the target objects; (4) be capable of mimicking a human operator (i.e., be able to provide target location information in terms of approach velocity as well as position); and (5) function in human real time (4 Hz.). An assumption that is allowed in order to minimize scene complexity is that the target objects are man made and a priori knowledge about them is available to the vision system. This is a reasonable assumption considering the nature of current and near future space operations.

## 3. System Configuration

The vision system is a distributed process within the Telerobotic System Simulation (TRSS) [1]. The system is functionally configured as two concurrent processes: the vision executive and the vision processor (fig. 1). The executive includes the functions of command interpretation, vision subtask determination, data base and modelling activities, local control activity, data conversion, and transfer of vision system status information to higher telerobotic system levels. The executive functions are performed by two modules referred to as the interpreter and the control interface. The interpreter directs the determination of target information by the vision system and the control interface processes and transmits the result to the telerobot's controller.

The interpreter's functions of command interpretation, subtask determination and sequencing, and data base organization and manipulation are hierarchical in structure and, therefore, are natural candidates for implementation as trees [2]. A tree is a collection of elements called nodes along with relationships among the nodes (e.g., parenthood, childhood, sequence, direction, precedence) that place a hierarchical structure on the nodes. A node can represent any entity (e.g., parent, child, subtask, shape, command) that does not violate the syntax or relational structure of the tree in which it exists (i.e. it must not impede the execution of the function). Trees can be subdivided into subtrees: A subtree consisting of shape nodes would represent an object, and one made up of command nodes would represent an execution imperative.

The vision interpreter is implemented as an abstract data type that allows the creation, deletion, and manipulation of trees of arbitrary size and function. The trees exist only at runtime and only when required to execute the requested function, thus, minimizing use of memory. As an example, assume that an imperative is received by the interpreter to locate a detected, but unrecognized object. The appropriate task tree is generated along with the necessary subtask, command, and object recognition subtrees embedded correctly in the task tree. The tree structure itself ensures the correct execution sequence. When the object is recognized, the recognition subtree is replaced by the object's description subtree known a priori, the location subtask subtree is generated, and the tree driven execution is performed again.

The control interface converts raw position data derived by the vision processor to a form compatible with the telerobot's control protocol. The TRSS data structure that handles dynamic system input/output is so constructed as to allow all position information to be accessed in terms of a common generic structure, generally referred to as an NSAP homogeneous matrix [3]. The matrix:

PAGE 22 INTENTIONALLY BLANK

$$\begin{vmatrix} Nx & Sx & Ax & Px \\ Ny & Sy & Ay & Py \\ Nz & Sz & Az & Pz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

is composed of an approach vector A describing the direction normal to the target plane, a sliding vector S denoting a direction normal to the A vector within the target plane and describing the rotation of the target plane about the A vector, the N vector which is the cross product of the S and A vectors, and the position vector P denoting the x, y, and z translations separating the axis systems of the camera and the target. The NSAP matrix contains all the information necessary to denote the orientation and position of the target with respect to the camera frame, and facilitates the various frame transformations that must occur in the telerobotic control process [4]-[5]. The angular parameters required for control can be extracted directly from the matrix. The decoupled angles used for finely resolved rate situations can be determined with the help of direction cosines as shown below:

$$\begin{aligned} &\text{rot. abt. } z = \arctan(Ny/Sy) \\ &\text{rot. abt. } y = \arccos(Az/(1 - Ay**2)**0.5) \qquad (1) \\ &\text{rot. abt. } x = \arccos(Az/(1 - Ax**2)**0.5) \end{aligned}$$

where checks for singularities and proper quadrants are implied. For position control situations or general system requirements, an NSAP to Euler transform has been implemented.

The vision processor performs the vision subtask as required by the executive and determines and advises the executive of the current status of vision processing. The vision processor is functionally segmented into low level, middle level, and high level processing. Low level processes include thresholding, gray level histogram generation and manipulation, and edge detection. Hardware and software implementing low level processes have generally been acquired from outside sources. Middle level processes include gray level based recognition, simple shape recognition, and target location. High level processes involve complex object recognition. Development and implementation of high level and middle level vision processes are the subjects of internal research. Two middle level processes that have been developed are discussed in this paper.

## 4. Monovision Methods

Two techniques that have application to the vision subtasks of segmentation, shape decomposition, recognition, and 3-space location are briefly discussed. The techniques are designed to extract 3-space information from a single two dimensional intensity image using prior knowledge and the principles of the perspective transformation.

The first method is based on the elastic matching [6] approach to pattern recognition and has application to shape decomposition, object recognition, and object location. It is an adaption of the linear programming technique of goal programming to the nonlinear problem of elastic matching [7]. Conceptually, elastic matching can be explained by envisioning a transparent reference image overlaying a goal image. The reference image is then warped or distorted to conform to the goal image by locally matching corresponding regions in the two images. The reference image is a flexible template that is modelled as a system of equation pairs where each equation pair represents a linear combination of patterns that a point in the reference image can describe in moving to a point in the goal image (fig. 2). The amount of displacement that each pattern contributes to the distortion is determined by identifying the values of the parameters Ai and Bi associated with each of the distortion patterns. The parameter values are derived by minimizing the absolute differences between corresponding reference and goal image points without violating the pattern constraints. This type of problem is easily modelled mathematically using the linear programming technique of goal programming [8]. The computational procedure that most efficiently resolves the optimal values of the goal programming model's parameters is the Simplex Algorithm.

The technique has been used to recognize simple three-dimensional objects of minimum curvature (i.e., near planar) and determine their location in 3-space. A single prototype shape (e.g., a rectangle) can be used to identify any of a primitive set of simple shapes by distorting it to match the image of an unknown shape. A simple shape is here defined to be a convex geometric figure formed on the surface of a sphere of large radius and the primitive set consists of rectangles, triangles, and ellipses. The values of parameters A3 through A5 and B3 through B5 yield information that allows recognition of the set members regardless of orientation. Once an object is identified, either as a simple shape or a combination of simple shapes, an exact model of its normal view is distorted to match the now known image, and information regarding its location and orientation can be derived from the parameters A0 through A3 and B0 through B3. Equations (2) through (7) show the geometric significance of the parameters.

$$\begin{aligned} A0 &= X' - X \qquad : \text{translation} \qquad (2) \\ B0 &= Y' - Y \end{aligned}$$

$$\begin{aligned} A1 &= -(1 - \text{gain}) \qquad : \text{gain} \qquad (3) \\ B1 &= -(1 - \text{gain}) \end{aligned}$$

where gain = X'/X or Y'/Y

$A2 = (X' - X)/Y$ : rotation in x-y plane (4)
$B2 = (Y' - Y)/X$

$A3 = -(1 - gain)/Y$ : perspective and (5)
$B3 = -(1 - gain)/X$ triangular shape information

$A4 = (X' - X)/a**2$ : semicircular shape information (6)

$B4 = (Y' - Y)/b**2$

where $a**2 = X**2 - Y**2$ and $b**2 = Y**2 - X**2$

$A5 = -(1 - gain)/Y**2$ : elliptical shape information (7)
$B5 = -(1 - gain)/X**2$

Equations (8) through (10), which are based on properties of the perspective transformation [9], show the parameters' relationship to the range, pitch, and yaw respectively of the target object relative to the camera's axis system.

$$range = (f*Wo*(2 - A1))/((1 - A1)*Ws) \quad (8)$$

where f is the focal plane distance of the camera/lens system, Wo is the object width, and Ws is the camera's image sensor width,

$$\tan \phi = 2*f*A3/(1 - A1) \quad (9)$$

where $\phi$ is the pitch angle, and

$$\tan \theta = 2*f*B3/(1 - B1) \quad (10)$$

where $\theta$ is the yaw angle.

Using a slightly different template (fig. 3), the technique has also been used to recognize arc segments and to decompose a geometrically complex object into its constituent shapes. The template is modelled as a system of n general equations of the second degree each of which represents a point on the arc segment of interest. The relative values of the derived parameters A, B, C, D, E, and F indicate the conic type of which the arc segment is a part (fig. 3) and their numerical values can be used to obtain the axis orientation, the foci, the vertices, the axis intercepts, and the eccentricty of the conic.

One way of determining a demarcation between simple shapes in an object's image is to locate boundary reversals (fig. 4). This is indicated when there is a rotation of axis between two adjacent arc segments such that the axes lie in diagonally opposite quadrants. The vertices of arc segments at the boundary reversals are used as end points of lines that subdivide the object's image into convex shapes that can be approximated by the primitive set.

By linearizing the problem, the computational efficiency of performing elastic matching is increased so that it becomes feasible as a real time procedure. Previous methods (e.g., exhaustive enumeration and dynamic programming) have required running times that are exponentially related to the number (n) of point pairs involved in the match:

$$T(n) \propto r**n \quad (11)$$

where r is the number of possible global match configurations. For an n variable problem, the worst case running time of the Simplex Algorithm is linearly related to n:

$$T(n) \propto n \quad (12)$$

When the flexible template is transposed to its dual [7]-[8], each pair of points to be matched requires a variable. Thus, the addition of point pairs has little impact on the running time of the elastic matcher [7]-[8]. When using the technique for object location, the position update frequency is 4 Hz., which is in the realm of human real time (1.333 to 4 Hz.). It must be noted that most of the time in the position determination/manipulator activation cycle of the current testbed is consumed by the image processing activity and not by the parameter identification and location calculations. A faster image processor would allow frequencies approaching video frame rates (30 Hz.).

The second method determines the location and orientation of a planar object from any four points on the object that describe a reasonably convex quadrangle. Given the inter-vertex distances of the quadrangle and the optical parameters of the camera, the rotational and translational displacements between the object and camera can be uniquely determined.

The distance and orientation of the quadrangle relative to the lens axis frame can be solved in a closed form. The object points are defined as perspective projections of the image points along rays originating at the lens center, that is

$$Ti = Ki*Ii \quad (13)$$

where the quadrangle <I0, I1, I2, I3> denotes the projection of the target <T0, T1, T2, T3> on the image plane (fig. 5). The axis system is chosen such that the x and y components of the projected image (Ix, Iy) lie on the image plane and Iz equals the focal length of the camera. In their paper on passive ranging, Hung and Yeh [10] prove that there exists a unique vector K which relates the target quadrangle and its image quadrangle and that it can be described in terms of the projected image points and the inter-vertex distances. The distances between the pairs of vertices can be described by a unique pair of nonzero real numbers, alpha and beta, independent of the coordinate system chosen, such that

$$I3 = I0 + \mathrm{alpha}*(I1 - I0) + \mathrm{beta}*(I2 - I0) \qquad (14)$$

where noncollinearity implies that

$$\mathrm{alpha} + \mathrm{beta} \neq 1 \qquad (15)$$

Equations (13) and (14) can be rewritten as

$$k3*T3 = k0*T0 + \mathrm{alpha}*(k1*T1 - k0*T0) + \mathrm{beta}*(k2*T2 - k0*T0) \qquad (16)$$

By substituting for the Ti and dividing by k3, equation (16) can be transformed to

$$I3 = (k0/k3)*(1-\mathrm{alpha}-\mathrm{beta})*I0 + (k1/k3)*\mathrm{alpha}*I1 + (k2/k3)*\mathrm{beta}*I2 \qquad (17)$$

where the I vector represents the (x, y, z) coordinates of the image points. Noting that k3 is common to all the right hand terms, it can be considered a scaling factor that reduces the target quadrangle from its original dimensions to its projected dimensions at the image plane where k3 equals 1. Thus, from similarity, Hung and Yeh describe k3 in terms of the relationship of the magnitudes of the real and projected diagonals:

$$k3 = \|T0 - T3\|/\|(k0/k3)*(1 - \mathrm{alpha} - \mathrm{beta})*I0 - I3\| \qquad (18)$$

This information is sufficient to solve for the three dimensional positions of the quadrangle vertices (Ti) in the camera axis frame. The quadrangle orientation, described by the equation of the normal to the plane occupied by the quadrangle in 3-space, is determined by substituting the coordinates of any three vertices into the general equation of the plane. Solving the system of simultaneous equations gives the following explicit expressions for the orientation vector in terms of the quadrangle vertices derived above:

$$\begin{aligned}
Ax' &= (T1y*T2z-T1z*T2y+T0z*T2y-T0y*T2z+T0y*T1z-T0z*T1y)/(D(T)) \\
Ay' &= (T1Z*T2x+T1x*T2z+T0x*T2z-T0z*T2x+T0z*T1x-T0x*T1z)/(D(T)) \qquad (19) \\
Az' &= (T1x*T2y-T1y*T2x+T0y*T2x-T0x*T2y+T0x*T1y-T0y*T1x)/(D(T))
\end{aligned}$$

where

$$D(T) = T0x*(T1y*T2z-T1z*T2y)+T0y*(T1z*T2x-T1x*T2z)+T0z*(T1x*T2y-T1y*T2x) \qquad (20)$$

and Ax, Ay, and Az are determined from Ax', Ay', and Az' by normalizing by the magnitude of the vector (Ax', Ay', Az').

Once the positions of the quadrangle vertices and the direction of its normal are known, the vectors that comprise the NSAP matrix can be found. The approach vector A is the orientation vector derived above. The sliding vector S is related to the slope of the base of the quadrangle with respect to the camera frame. It is the x, y, and z components of the vector T1 - T0 normalized by its length. The position vector P is simply the components of the selected point of approach on the quadrangle <T0, T1, T2, T3>. The intersection of the diagonals is commonly chosen.

For each probable target, it is necessary to determine and specify the alpha and beta parameters, based upon the inter-vertex distances of the quadrangle for each target introduced. One approach to entering new models in the data base is to automate this task in a one shot initialization procedure by processing one frame of the target image from a camera position normal to and at a known distance from the target. These parameters are calculated and stored in the data base. The calculations are based on equation (13) (and its transformations) with the K vector known. The results are presented here without derivation.

$$\begin{aligned}
\mathrm{alpha} &= V2/V1 \qquad (21) \\
\mathrm{beta} &= V3/V1
\end{aligned}$$

where

$$\begin{aligned}
V1 &= I0x*(I2y - I1y) + I1x*(I0y - I2y) + I2x*(I1y - I0y) \\
V2 &= -(I0x*(I3y - I2y) + I2x*(I0y - I3y) + I3x*(I2y - I0y) \qquad (22) \\
V3 &= I0x*(I3y - I1y) + I1x*(I0y - I3y) + I3x*(I1y - I0y)
\end{aligned}$$

The raw state information consisting of the three translational and the three angular displacements of the target from the camera generated by both the elastic matcher and quadrangle projection methods is converted to the NSAP matrix. This matrix is input to the interface control section of the vision executive for further processing.

## 5. Future Work

The vision system development in the ISRL centered on the processing of single, two dimensional, intensity based (i.e., video) images. The next research phase will involve the extension of the system to process single three dimensional range based images as well as further refinement of the two dimensional techniques. The successful development of a laser vision sensor based on the FM-CW radar technique will support the next phase [11].

## 6. References

[1] F.W. Harrison, Jr. and J.E. Pennington, "System Architecture for Telerobotic Servicing and Assembly Tasks," presented at the SPIE 1986 Cambridge Symposium on Optical and Optoelectronic Engineering, Cambridge, Massachusetts, October 26-31, 1986.

[2] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, Data Structures and Algorithms, Addison-Wesley, Reading, Massachusetts, 1983.

[3] C.S.G. Lee, "Robot Arm Kinematics, Dynamics, and Control," IEEE Computer, Vol 15, No. 12, December 1982, pp. 62-80.

[4] R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control, MIT Press, Cambridge, Massachusetts, 1981.

[5] L.K. Barker, "Kinematic Equations for Resolved-Rate Control of an Industrial Robot Arm," NASA TM-85685, November 1983.

[6] B. Widrow, "The Rubber Mask Technique," Pattern Recognition, Vol 5, 1973, pp. 175-211.

[7] P.W. Goode, "A Multifunction Recognition Operator for Telerobotic Vision," Proceedings of the AIAA Guidance, Navigation, and Control Conference, August 1986.

[8] F.S. Hillier and G.J. Lieberman, Introduction to Operations Research, 3rd edition, Holden-Day, San Francisco, 1980.

[9] R.M. Haralick, "Using Perspective Transformations in Scene Analysis," Computer Graphics and Image Processing, Vol 13, pp. 191-221, 1980.

[10] Y. Hung, P. Yeh, D. Harwood, "Passive Ranging to Known Planar Point Sets," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 80-85, 1985.

[11] F.E. Goodwin, "Coherent Laser Radar 3-D Vision Sensor," Proceedings of Sensors '85, November 1985.

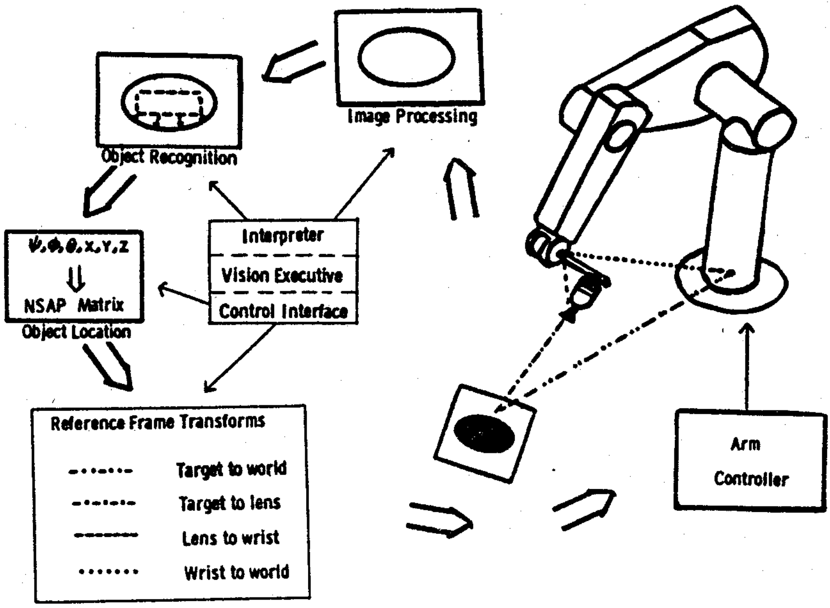


Figure 1. - System configuration.

$A_0, B_0$ : Translation

$A_1, B_1$ : Gain

$A_2, B_2$ : Rotation in X-Y plane

$A_3, B_3$ : Perspective of triangular shape information

$A_4, B_4$ : Semicircular shape information

$A_5, B_5$ : Elliptical shape information

$f(x, y)$ : Model function

$f(x', y')$ : Image function

$$X + A_0 + A_1X + A_2Y + A_3XY + A_4(X^2 - Y^2) + A_5XY^2 = X'$$

$$Y + B_0 + B_1Y + B_2X + B_3XY - B_4(Y^2 - X^2) + B_5YX^2 = Y'$$

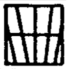
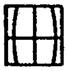
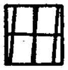
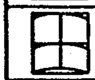
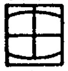
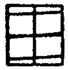
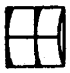
| Distortion pattern | Distortion term |
|---|---|
|  | XY |
|  | $XY^2$ |
|  | Y |
|  | $Y^2$-$X^2$ |
|  | $YX^2$ |
|  | X |
|  | $X^2 - Y^2$ |

Figure 2. - Elastic template.

MINIMIZE SUM OF ABSOLUTE DEVIATIONS OF MODEL POINTS FROM IMAGE POINTS

SUBJECT TO:

$$Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F = 0 \qquad i = 1,n$$

NONTRIVIAL SOLUTION CONSTRAINT

| | |
|---|---|
| $B^2 - 4AC = 0$ | PARABOLIC ARC |
| DEGENERATE CASE | LINE |
| $B^2 - 4AC < 0$ | ELLIPTIC ARC |
| DEGENERATE CASE | POINT |
| $B^2 - 4AC > 0$ | HYPERBOLIC ARC |
| DEGENERATE CASE | CORNER |
| $\tan 2\theta = \frac{B}{A - C}$ | THETA IS ANGLE OF CONIC'S AXIS WITH X AXIS |

Figure 3. - Arc segment identification.

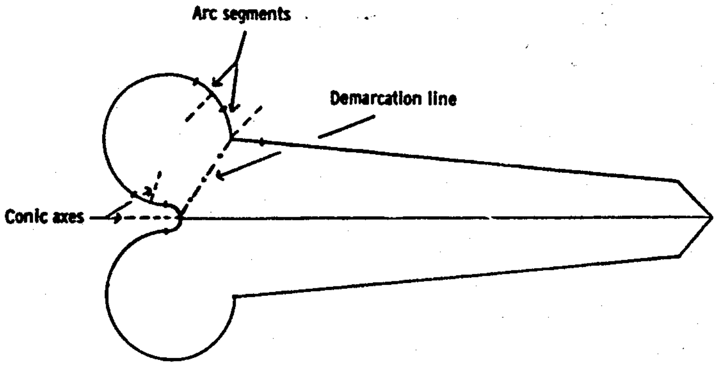


Figure 4. - Shape decomposition.

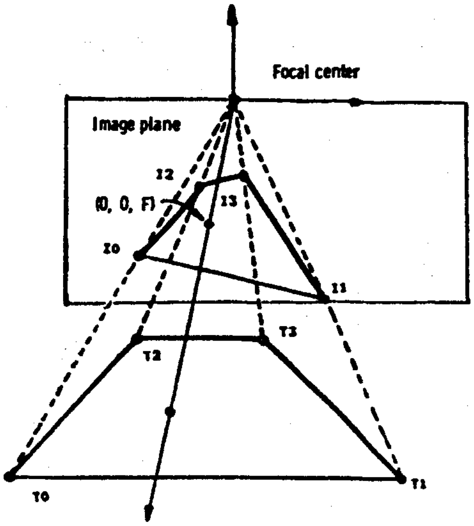


Figure 5. - Quadrangle projection.