ON CALIBRATING COMPUTER CONTROLLED CAMERAS FOR PERCEIVING 3-D SCENES

by Irwin Sobel*

Department of Computer Sciences
Technion - Israel Institute of Technology
Haifa, Israel

Abstract

A problem that arises in getting computers to perceive 3-D scenes is relating information from several different viewpoints. In particular, if the computer moves its sensor, it has to be able to predict changes in images of objects it has seen without having to completely re-recognize them. A solution to this problem has been implemented at Stanford using a calibrated camera model which expresses the relation between object space and image space as a function of the computer's control variables. The modelling problem is relatively well understood. Calibration techniques, however, are not. This article deals with these.

Descriptive Terms
robots, computer vision, artificial intelligence, photogrammetry, visually guided manipulators, automatic assembly

Introduction

Image analysis for 3-D scene perception is computationally expensive. It is thus important that a computer which moves its sensor be able to predict changes in the images of objects it has already seen without having to completely re-recognize them. We will present a solution to this problem that has been implemented at Stanford for a visually guided manipulator system.

T The Stanford hand-eye project is organized around a dual-processor PDP-10/PDP-6 computer system. Two electrically-powered mechanical arms and two standard TV cameras with pan-tilt heads are interfaced to the computer; they serve as hands and eyes respectively. In addition, the central part of the work space is a "lazy susan" turntable (see figure 1). To have the capability to visually locate objects in 3 dimensions, it is desireable to be able to see any point in the work space from two distinct viewpoints. In general this requires several fixed cameras, and/or a highly mobile camera, and/or the ability for manipulating the environment so as to turn things around to see them better. The two cameras with pan-tilt heads, and the lazy susan, facilitate this. The computer is capable of moving the 'arms, turntable, and cameras, and sensing the position of all moveable joints. Information flow is schematized for the "right half" of the system in figure 2. One camera is fitted with a zoom lens, while the other has a rotatable turret with 4 lenses. Both are fitted with color wheels. Zoom, turret-lens-selection, and color-filter-selection are under computer control. Also under computer control are focus, iris (on the zoom camera), and vidicon sensitivity. Visual information is transmitted to the computer by quantizing a TV image Into an array of 250 * 333 samples. Each sample is a 4-bit number representing 1 of 16 possible light levels. A whole image or any rectangular subfield may be read into the computer memory from a camera.

This discussion deals with only one aspect of the hand-eye system. For more information of a general and historical nature, the reader is referred to8''*13*20' z"*36. Four PhD. theses7* 12,21,34 describe other major aspects of the system. In addition another project thesis is forthcoming14 and is referenced in anticipation of Its publication. For detailed information about manipulators, see18'29'28.

*The research reported here was performed at the Stanford Artificial Intelligence Project and was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract SD183

The Problem

The problem can be separated into two main subproblems - modelling and calibration. The modelling problem is straightforward geometry; parts of it have been worked out in many places2*3'**16'26'30. To review it briefly, we want an analytic camera model relating 3-D coordinates of points in object space to the 2-D locations of their images in a digitized picture. An adequate model for these purposes can be derived by treating a picture as a central projection of object points onto a (image) plane. Such a projection can be described by (see figure 3a):

A - (A_x, A_y, A_z) the location of the center of projection in object space (camera location).

R " $[r_{ij}]$ a 3-D rotation matrix for specifying the camera's orientation. It orients an (xyz) camera frame at A w.r.t. the (XYZ) object frame (at 0 in the figure). Such a matrix is orthonormal and thus has only 3 degrees of freedom which in our case were chosen to be the elementary rotation angles (see figure 3b):
- PAN - about the unrotated y-axis (assumed parallel to the object Z-axis)
- TILT - about the once rotated x-axis
- SWING - about the twice rotated z-axis

f - the normal distance from A to the image plane

The 6 degrees of freedom contained in A and R locate and orient the camera, while f determines the basic scale of the projection. A and *It* are called external geometry parameters of the camera.

The process' of digitization further scales the image by quantization factors Mx,My which represent the density of samples per unit length in the x and y directions respectively. In addition the coordinate origin is translated from the principal point pe to the upper left comer of the image* The combined scales fMx and fMy, and p̄o are called internal geometry parameters of the camera. The resulting digitized image coordinates (hv) are called (image) horizontal and vertical respectively (see figure 3c).

Controlling the camera here means providing the ability to sense and change some or all of the external/internal geometry parameters. In general feedback sensors do not measure these parameters directly, but functions of them. We call these sensed functions control variables. For a given configuration we can

*The principle point is the piercing point of the ray from X normal to the image plane. For TV systems a small correction (< 1 pixel) is also made for the skewness of the scanning raster.

find expressions for the camera geometry parameters in terms of the control variables. These expressions always contain fixed, but unknown, parameters which must be measured. The process of satisfactorily measuring these fixed parameters is what we call calibration. Since we are interested in developping a predictive model, it is desireable to find a set of such parameters that best accounts for images of known object configurations. An analytical derivation of such a "best" set typically leads to complicated simultaneous transcendental equations which do not yield to closed form solution. Moreover, the form of these equations depends strongly upon camera geometry and the number and type of control variables.

To be more precise, modelling can be described as follows: Write down the transformation relating the 3-D coordinates of objects to the 2-D coordinates of their images. This depends upon the position and orientation of the camera in object space - so-called external geometry - and on its internal geometry. All elements of external and internal camera geometry under computer control, should be expressed as functions of the computer's control variables $\overline{\alpha} = (\alpha_1,\ldots,\alpha_n)$. These will typically be outputs of feedback sensors. The transformation is schematized in figure 4. Typical geometric variables for a camera on a pan-tilt head are the angles of PAN and TILT, and f which is the distance from the lens rear nodal point to the image, f is affected by both zooming and focusing motions. The corresponding control variables are potentiometer or shaft-encoder readings. They are usually linearly - but not always - related to the geometric quantities. e.g-

$$\alpha_1 = \langle\text{pan pot reading}\rangle = \beta_1\langle\text{pan angle}\rangle + \beta_2 \qquad (1)$$

The camera transform can be represented by the vector equation

$$\overline{p} = \overline{C}(\overline{\alpha},\overline{P})$$

where $\overline{p} = (h,v)$ is the vector of image coordinates and $\overline{P} = (X,Y,Z)$ is the vector of object coordinates. (2)
This is equivalent to the following scalar equations:

$$h = C_h(\alpha_1,\ldots,\alpha_n,X,Y,Z)$$
$$v = C_v(\alpha_1,\ldots,\alpha_n,X,Y,Z)$$

The particular form of C_h and C_v is determined both by camera geometry, and the relations of the α's to their corresponding geometric variables. Whatever this form it will always contain parameters $\overline{\beta} = (\beta_1,\ldots,\beta_m)$ not explicitly shown in (2). Usually these are neither under computer control nor directly measureable by the computer. To determine C_h and C_v for any particular moveable camera arrangement, one must nave some way of measuring these. The process of measuring $\overline{\beta}$ camera calibration. The β's are hopefully fixed or vary only infrequently, so that calibration need be done once and updated only if the β's drift ot the geometry is modified. $\beta_1,\ldots,\beta_m$ are typically

a) numbers which relate sensor (pot or shaft-encoder) readings to angles and/or distance e.g. β_1,β_2 of (1)

or b) offset angles and/or distances between fixed elements of the camera mechanism

or c) scale factors expressing distance ratios or quantization units

The Calibration System

At Stanford we designed and constructed an operational system[30,31] for calibration of a camera with a zoom lens on a pan-tilt head (see figure 5)* There are in fact m - 18 β's:

8 in class (a) above - 2 each for relating sensor outputs to geometric quantitias associated with pan, tilt, zoom, focus.
Of the remaining 10, 7 represent the external camera geometry and fall into class (b) above, while 3 represent the internal camera geometry - 2 in (b) and 1 in (c).

There are n = 4 control variables (α's) for pan, tilt, focus, and zoom. The system calculates a "best" $\overline{\beta}$ vector given 10 or more object-image point-pairs and their associated α's. It is divided into two parts: data collection and model optimization (see figure 6).

For data collection, pictures are taken of reference objects,with known shape (usually cubic) and location that are provided to the program by a human operator. At the time an image is transferred to the computer, the program reads the camera-control feedback sensors (pan, tilt, zoom, and focus pots} to ascertain the current α. An edge-follower program ' then processes the image to extract boundary points of the object. A polygon is fit to the points. The polygon vertices are ordered to match up with the object space coordinates. Thus for the j— picture of a reference cube, the collection routines will generate the data

$$\{\ \overline{\alpha}_j\ ;\ \{\overline{P}_{ij},\overline{p}_{ij}\}_{i=1}^{6}\ \}$$

where $2 \leq j \leq N$ is the image number
and $1 \leq i \leq 6$ is the point index within an image

This is stored on a data file for use in the optimization calculation.

For model optimization, several data files are accumulated covering the working ranges of the α's. The calculation is a direct-search optimization[33] program. It starts with an initial set of β's and computes a set of predicted image points

$$\overline{p}^*_{ij} = (h^*_{ij},v^*_{ij})$$

$$\text{where} \quad h^*_{ij} = C_h(\overline{\alpha}_j,\overline{\beta},\overline{P}_{ij}) \qquad (3)$$
$$\text{and} \quad v^*_{ij} = C_v(\overline{\alpha}_j,\overline{\beta},\overline{P}_{ij})$$

which is just equation (2) rewritten with $\overline{\beta}$ made explicit. The program compares these predictions with actual image-coordinates

$$\overline{p}_{ij} = (h_{ij},v_{ij})$$

measured by the image processing input routines, and computes a mean-squared prediction-error:

$$E^2 = \frac{1}{6N}\sum_{j=1}^{N}\sum_{i=1}^{6}|\overline{p}^*_{ij} - \overline{p}_{ij}|^2 \qquad (4)$$

$\overline{\beta}$ is then systematically varied so as to minimize the the error E^2.

To get an idea of how E^2 is affected by the number of images and points/image, note that a data triplet $(\overline{P},\overline{\alpha},\overline{p})$ yields 2 scalar constraint equations for the unknown $\overline{\beta}$:

*The turret camera is a special case of this wherein the zoom control is discrete instead of continuous.

$$h = C_h(\bar{\alpha},\bar{\beta},\bar{P})$$
$$v = C_v(\bar{\alpha},\bar{\beta},\bar{P}) \quad (5)$$

These are different from (3) in that h,v are measurements and B is to be calculated. Assuming non-degenerate equations, we need at least 18/2-9 such triplets to completely constrain B. The triplets should also be Independent in the sense that each triplet yields new information (e.g. 2 P's in the same image along the same central ray yield the same p" and are not independent).

We can get an idea of the minimum number images (i. e. distinct values of a) needed, by using the fact that there are 10 basic geometric parameters for a fixed camera (a - const); A", PAN, TILT, SWING, fMy, MRAT, and pD - We can write constraint equations similar to (5) with the B's replaced by the geometric parameters, and deduce that we need 10/2-5 points P to completely solve for these. Of these basic parameters, MRAT, po * and SWING, are also B's. Another image, taken with different values for all the a's, will only have 6 Parameters unknown Of, PAN, TILT, fMy), and only 6/2-3 P's will be needed to solve for the new values. They may_In fact be part of the original 5. T h two vu e s of A are sufficient to solve for 6 more (J's $\overline{DP},\bar{P}_0$ to bring the total to 10. The two values each of PAN and TILT are also sufficient to solve for their A associated B's, leaving only the 4 B's relating the focus and zoom pots to fMy. Unfortunately each new image, created by changing focus and/or zoom gives us only one new equation i n fM_y. Thus WE need at least 2 more images of one point $\bar{P}$, not on the camera axis, to get a total of 4 values for fMy and associated pot readings to give a soluble system of 4 equations in the remaining 4 B's In summary we have a specific method of solving for the 18 bv's using 4 images using 5+3+1+1-10 data triples. The total number of distinct points can be the original 5. Thus though considerations of numbers of variables and constraint equations tell us that 9 data triples are necessary, the form of the equations seems to say that we need at least 10, arranged in 4 images as explained.

One of the features of the optimization program is that it is Interactive:

1 - It has a rather elaborate dynamic display (figure 6b).
2 - It allows the user to restrict the search to any specified subspace of $\bar{\beta}$ space.
3 - It allows him to choose between two search algorithms"'"'81.
4 - It allows him to initialize $\bar{\beta}$ to any values he cares to.
5 - It allocs the search to be interrupted at any time and the residual error/image to be displayed as a function of any of the a's. This is to see if there systematic deviations w.r.t. any given control variable. Such a deviation would indicate a defect in the model or the equipment associated with that α.
6 - It has a test mode for exploring the convergence of the optimization algorithms in the neighborhood of any desired B The user can either manually Input 8 or take a typical value arrived at from data. Upon entering test mode, p replaces lP, forcing E to zero. The user is then allowed to add a simulated distortion function to the new P if he chooes. He is finally asked to perturb (J and observe the resulting convergence back to the ideal point.

The search is complicated by the fact that subsets of parameters are interdependent - e.g. to first order accuracy, changes in E due to changes in po, can be offset by the PAN and TILT offset β's. It has been empirically determined that there are multiple minima of E2, but they are spaced sufficiently far apart so that pur initial guesses are good enough not to go astray. Starting points are usually arrived at from rough manual measurements and calculations of the system geometry. A typical optimization sequence consists of:

1 - reading in about 10 data sets
2 - If a previous initial β has not been stored with the data, it must now be typed in.
3 - choosing an appropriate subspace of $\bar{\beta}$-space to search (maybe all of it)
4 - letting the first algorithm (usually more efficient) converge
5 - applying the second algorithm to try to improve the result if not good enough
6 - possibly returning to 3 and changing the subspace (optimizing over the whole space if not done)

The residual rros error E Is a measure of the goodness of the model, and Is typically I to 2 picture elements (pixels), which corresponds to 1.5 to 3.0 milliradians on the camera axis - with the particular optics used. Convergence typically takes about 5 min. of PDP-10 cpu time for all 18 parameters. The residual errors are due mainly to mechanical vibration of the optical system and electrical jitter of the scan electronics. These have so Ear masked lens and scan distortions.

The Future

At present there is a program at the hand-eye project to expand the scope and efficiency of the calibration system. The first step is to elimenate the operator to manually measure and specify reference-object coordinates (see figure 6a). We hope to do this by using the mechanical arm to place the objects according to a prespecified calibration sequence. The resulting scheme will look as shown in figure 7. The prediction error in effect then becomes a "coordination error" between the hand and eye; the arm controller will be reporting the object vertex positions PQR to the precision of the model relating its control variables a a r m to requested hand position and orientation. Meanwhile the camera calibration will be using the reported position information $\overline{PQR}(\bar{\alpha}_{arm},\bar{\beta}_{arm})$ to predict image coordinates. Gill12 measured this coordination error in his system for precise object manipulation. He adjusted several camera B's suspected of drifting sO as to minimize it.

In this system, predicted image coordinates $\bar{P}^*_{ij}$ are

$$\bar{P}^*_{ij} = \bar{C}(\bar{\alpha}_j,\bar{\beta},\bar{P}_i(\bar{\alpha}_{armj},\bar{\beta}_{armj}))$$

These will be in error due to errors in both arm and camera calibrations $\bar{\beta}_{arm},\bar{\beta}$ (see figure 7). The $\bar{\beta}$ errors cannot be seperated from the $\bar{\beta}_{arm}$ in the function E^2 of (4). Thus for smallest coordination error, it is desireable to jointly minimize both arm and camera models - about 40-50 parameters. This is a formidable optimization problem for 2 reasons:

1 - The behavior of the overall error function E! w.r.t. all these has not been investigated - in particular, for the presence of local minima near the global minimum.
2 - Unless the initial E2 is already quite small, our current optimization procedures may well bog down on so many parameters, so as to be useless.

This strongly suggests that we use joint optimization only as a last refinement, and even then possibly with approximate transform equations for faster convergence near the minimum. Joint calibration is ultimately desireable, since the principal use of camera calibration is to get world coordinates for the purpose of guiding the arm.

Another system being considered utilizes two cameras, and optimizes their relative orientation based upon simultaneous measurement of the same calibration object from two different viewpoints. Thus, if a first camera has been calibrated relative to the reference object-space, a second can be calibrated relative to it etc. This technique is used extensively in photogrammetry2 for compiling maps from large numbers of aerial photographs. A two-camera system, once calibrated, provides a 3-D measuring tool which can in turn be used for basic arm calibration.

What is really needed to simplify and speed-up optimization is a separable error function. For example, something of the form

$$\text{Error} = f_1^2(\beta_1) + f_2^2(\beta_2) + \ldots + f_n^2(\beta_n)$$

or even

$$\text{Error} = f_a^2(\beta_1,\ldots,\beta_k) + f_b^2(\beta_{k+1},\ldots,\beta_n)$$

which is more likely, would allow us to separately optimize parts of the system while still guarantying minimum overall error.

Yet another direction for improvement is that of calibration-updating; that is, re-calibrating the system rapidly whenever observations in the course of normal operation show that errors have grown intolerably large. The emphasis here is on the word rapid - hopefully near real-time. One way to accomplish this is to collect data from measurements made during normal operation of the system.

The fully automatic calibration system of the future will probably go through a bootstrapping phase; either an arm or a pair of eyes - whichever is more accurate - will first be calibrated. This will be done possibly with the aid of a special calibration device for providing highly accurate reference data. The calibrated half of the system will then be used to provide data for calibrating the other half, at which point the total system will be jointly optimized to minimize coordination errors.

References and Bibliography

1. Agin, G., "Representation and Description of Curved Objects", PhD. thesis Elec. Eng. Dept. Stanford Univ. July 72. Available as AIM-173 from Stanford A.I. Project

2. American Society of Photograrametry, Manual of Photogrammetry, 2 vols., Falls Church, Va., 3rd Ed., 1966 (see esp. Vol. 1 Ch. X on Analytical Photogrammetry).

3. Baumgart, B., forthcoming PhD. thesis on Computer Graphics and Visual Perception, Stanford Univ. Computer Science Dept.

4. Brent, R.P., "Algorithms for Finding Zeros and Extrema of Functions Without Calculating Derivatives", PhD. thesis, C.S. Dept., Stanford U., Mar. 1971, available as CS-198

5. Duda, R., Hart, P., book in press: Pattern Classicflcation and Scene Analysis, Dec, 1970, avail. in preprint form from Stanford Research Institute.

6. Earnest, L.E., "On Coosing an Eye for a Computer", Stanford AIH-51, April 1967

7. Kalk, G., "Machine Analysis of Multi-Body Scenes", PhD. thesis, Stanford U..C.S. Dept. 1970, avail, as CS-180 and AIM-132

8. Feldman, J.A., et. al., "The Stanford Hand-Eye Project", Proc. First Int. Conf. on A.I., Wash., D.C., 1969

9. Feldman, J.A., "Getting a Computer to See Simple Scenes", IEEE Student Jour., Sept. 1970

10. Gibson, J.J., The Senses Considered as Perceptual Systems, Houghton-Miflin, Boston 1966

11. Gibson, J.J., Percept!on of the Visual World, Houghton-Miflin, Boston, 1950

12. Gill, A., "Visual Feedback and Related Problems in Computer Controlled Hand-Eye Coordination", PhD thesis, Elec. Eng. Dept., Stanford U., avail. AIM-178 from Stanford A.I. Project

13. Grape, G., "Computer Vision Through Sequential Abstractions", Internal Stanford A.I. Project Memo (SAIL0N) June, 1969

14. Grape, G., "Computer Vision Through Sequential Abstractions", forthcoming PhD. thesis, C.S. Dept., Stanford U.

15. Horn, B.K.P, "Focusing", MIT Project MAC A.I. Memo No. 160, May 1968

16. Horn, B.K.P., "Shape from Shading", MIT PhD. thesis, avail, from MIT A.I. Project as MAC TR-79, Nov. 1970
for geometry see "V1SMEM: A bag of "robotics" formulae, MIT A.I.Lab. Memo (Dec. 1972)

17. Hueckel, M.H., "An Operator Which Locates Edges in Digitized Pictures", JACM, V. 18, NO. 1, Jan. 1971, pp. 113-125. Also Stanford AIM-105

18. Kahn, M., "Optimal Control of a Hydraulic Arm", PhD. thesis, Stanford M.E. Dept. 1969, avail, as Stanford AIM-106

19. Nilsson, N.J., "A Mobile Automaton: An Application of Artificial Intelligence Techniques", Proc. IJCAI, pp.509-515, May 1969

20. Paul, R., Falk, G., Feldman, J., "The Computer Representation of Simply Described Scenes", Proc. Illinois Graphics Conf., U. of 111., Urbana, 111. April 1969

21. Paul, R., "Studies With a Computer Controlled Arm", PhD. thesis, C.S.Dept., Stanford U. Aug. 1972 Avail, as Stanford A1M-

22. Perkins, D., MIT PhD. thesis Math. Dept. on Stereo Perception of Line Drawing Pairs, Oct. 1970

23. Pieper, D.L., "The Kinematics of Manipulators Under Computer Control", M.E. PhD. thesis, Stanford U., Oct. 1968, avail, as Stanford AIM-72

24. Pingle, K., Singer, J., Wichman, W., "Computer Control of a Mechanical Arm Through Visual Input", Froc. IFIP Congress 1968 Vol.2

25. Pingle, K., "Visual Perception by a Computer", in Automatic Interpretation and Classification of Images, Ed. Grasselli, Academic Press, N.Y., London 1969, pp. 277-284. see also "System Manual for Hand-Eye Hackers", Stanford A.I. Project Internal Memo (SAILQN) June 1971

26. Roberts,L.G., "Machine Perception of Three-Dimensional Solids", MIT Lincoln Lab., Lexington, Mass. Tech. Rpt. No. 31S, May 1963

27. Roberts, L.G. "Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs", MIT Lincoln Lab., Lexington, Mass., Memo MS14Q5 May, 1965

28. Scheinman, V.D., "Design of a Computer-Controlled Manipulator", Engr. thesis M.E. Dept. Stanford *V.,* avail, as Stanford AIM-92, June 1969

29- Schmidt, R., "A Study of the Real-Time Control of a Computer Driven Vehicle", PhD- thesis Stanford U. Elec. Eng. Dept,, Aug. 1971, avail as Stanford AIM-149/CS-231

30. Sobel, I.E., "Camera Models and Machine Perception" PhD. thesis, Stanford U., Elec- Eng. Dept. May, 1970, avail, as AIM-121

31. Sobel, I.E., "Operating Manual for Camera Calibration", Stanford A.I. Project Internal Memo Oct. 1972

32. Springer, C.E., Geometry and Analysis of Projective Spaces, W.H. Freeman and Co., San Francisco, 1964 (see sections 5.2 and 7.1)

33. Temes, G.C., Calahan, D.A., "Computer Aided Network Optimization - The State of the Art", Proc, IEEE Vol. 55, No.11, Nov. 1967, pp.1832-63

34. Tenenbaum, J.M., "Accomodation in Machine Perception", PhD. thesis, Stanford U. Elec Eng. Dept. Aug. 1970, avail, *as* Stanford AIM-134/CS-182

35. Von Senden, M., Space and Sight, The Free Press, Glencoe, 111., 1960 edition

36. Wichman, W., "Use of Optical Feedback in the Computer Control of an Arm", Engr. thesis, Stanford U. Elec. Eng. Dept. Aug. 1967, avail as Stanford AIM-56

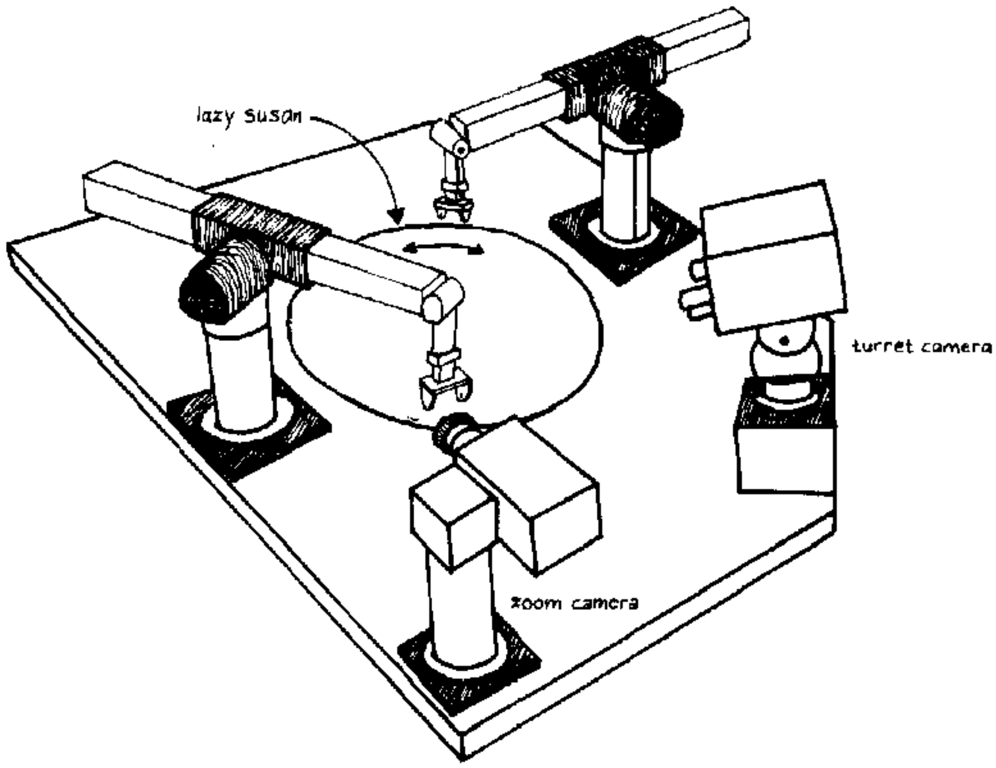


Figure 1 STANFORD HAND-EYE CONFIGURATION

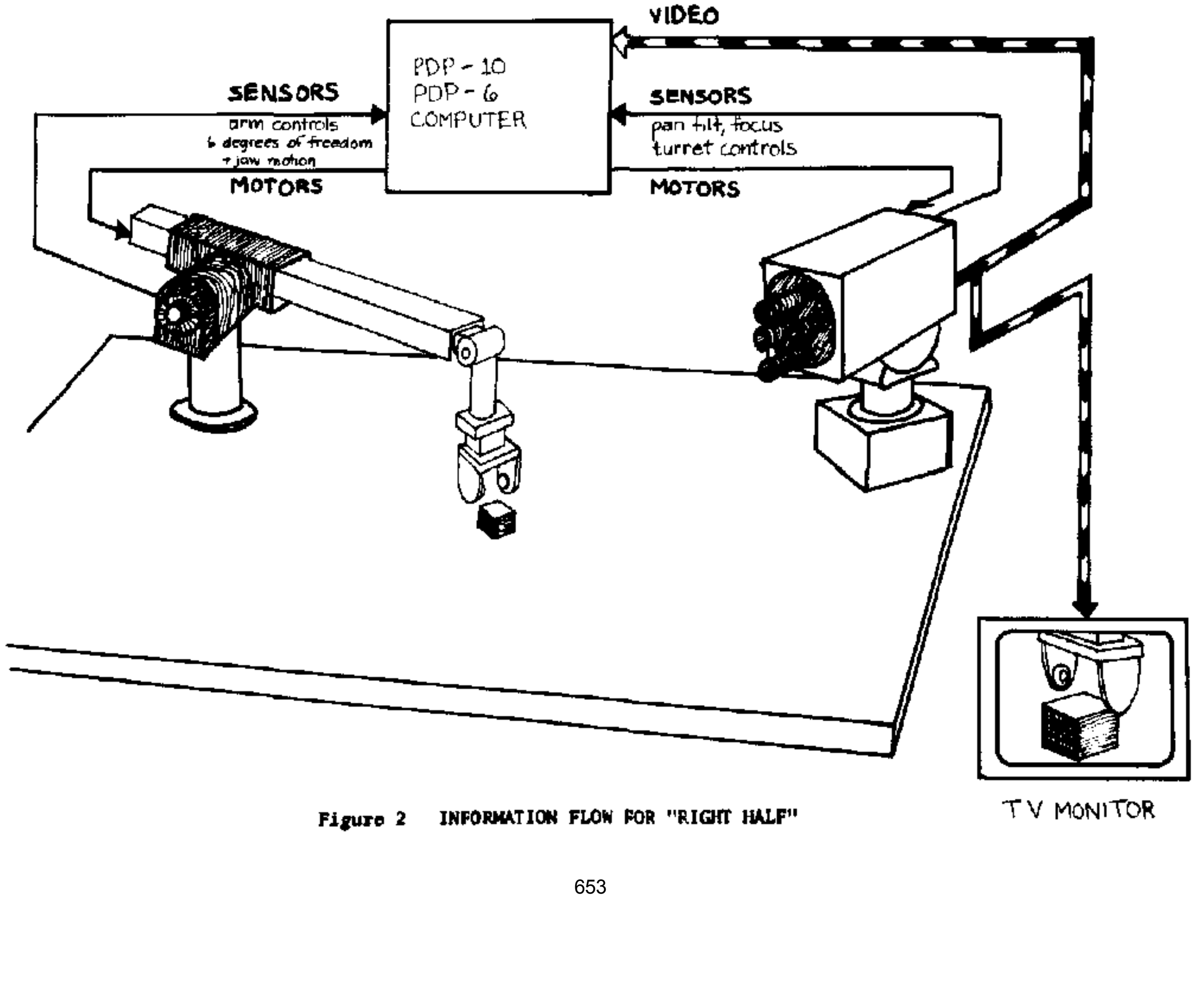
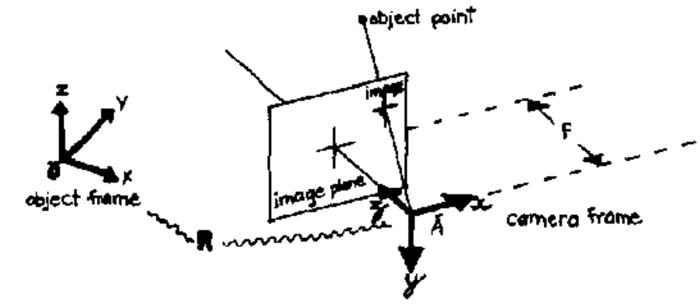
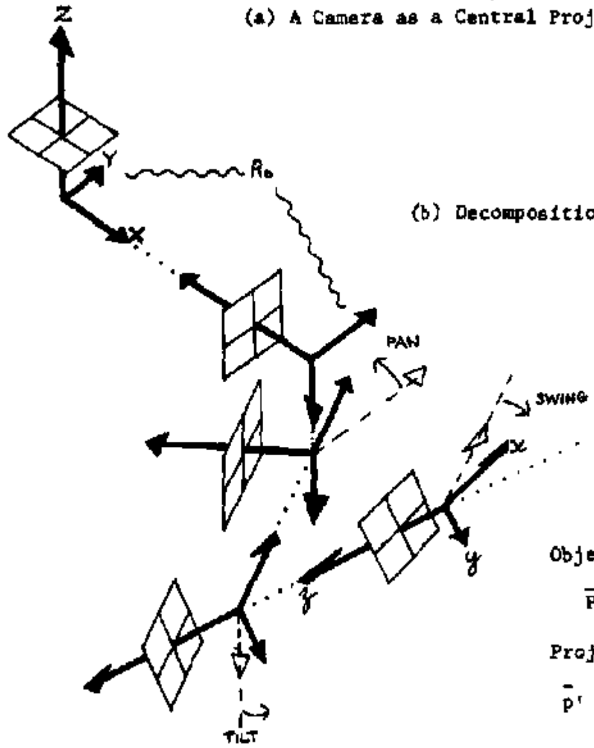


Figure 2 INFORMATION FLOW FOR "RIGHT HALF"



(a) A Camera as a Central Projection



(b) Decomposition of R to elementary rotations

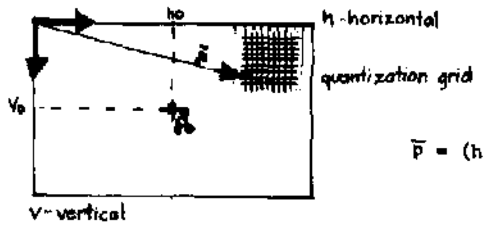
$$R = R_0*[PAN]*[TILT]*[SWING]$$

Object point $\overline{P}$ in camera frame is

$$\overline{P}' = (\overline{P} - \overline{A})*R \equiv (x',y',z')$$

Projection is (after dropping z-coord.)

$$\overline{p}' = (f/z')\overline{P} = (fx'/z', fy'/z') \equiv (x,y)$$



(c) The Digitized Image

$$\overline{P} = (h,v) = (xM_x+h_0, yM_y+v_0)$$

Figure 3 MODELLING

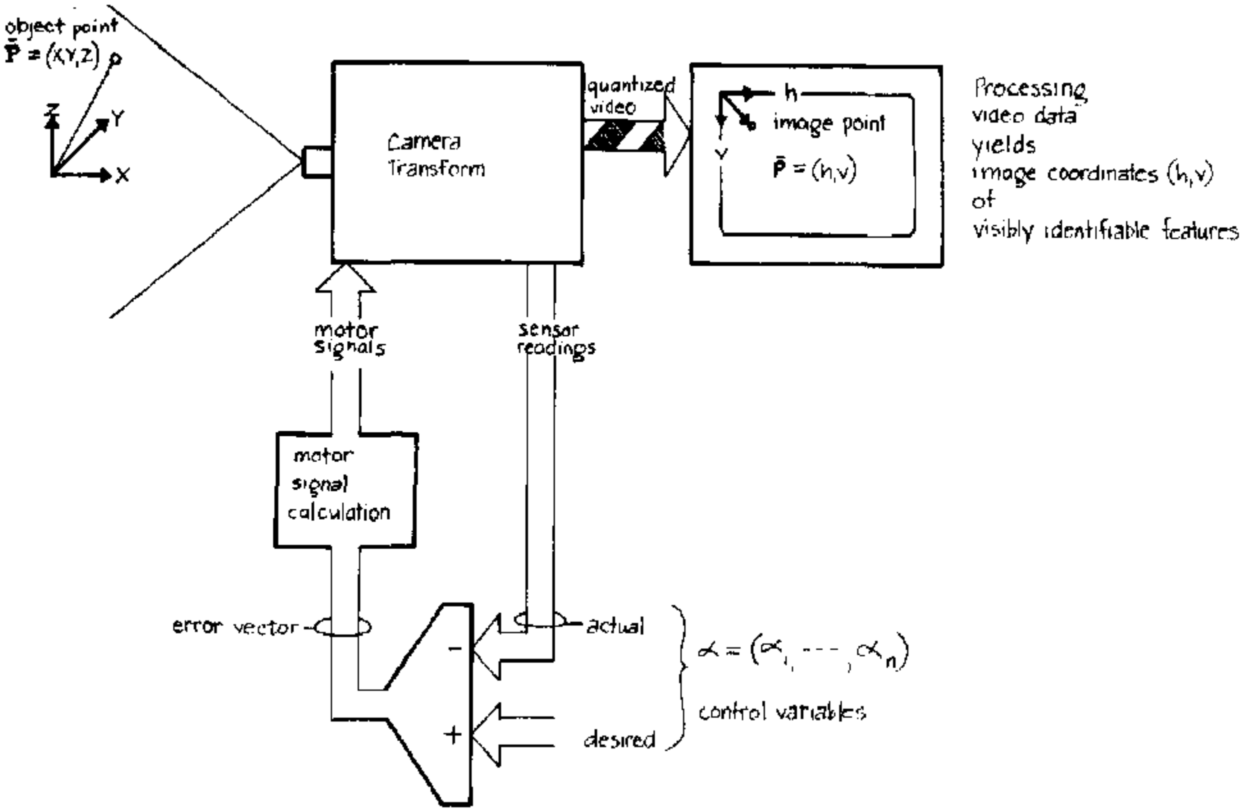
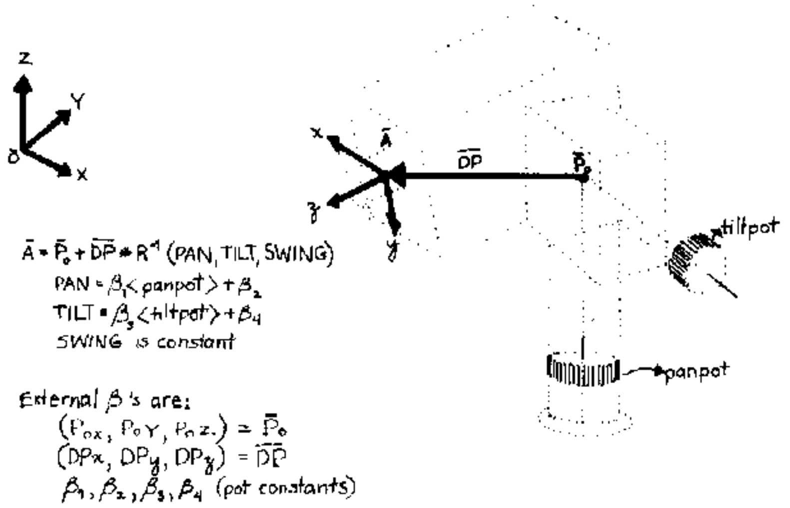
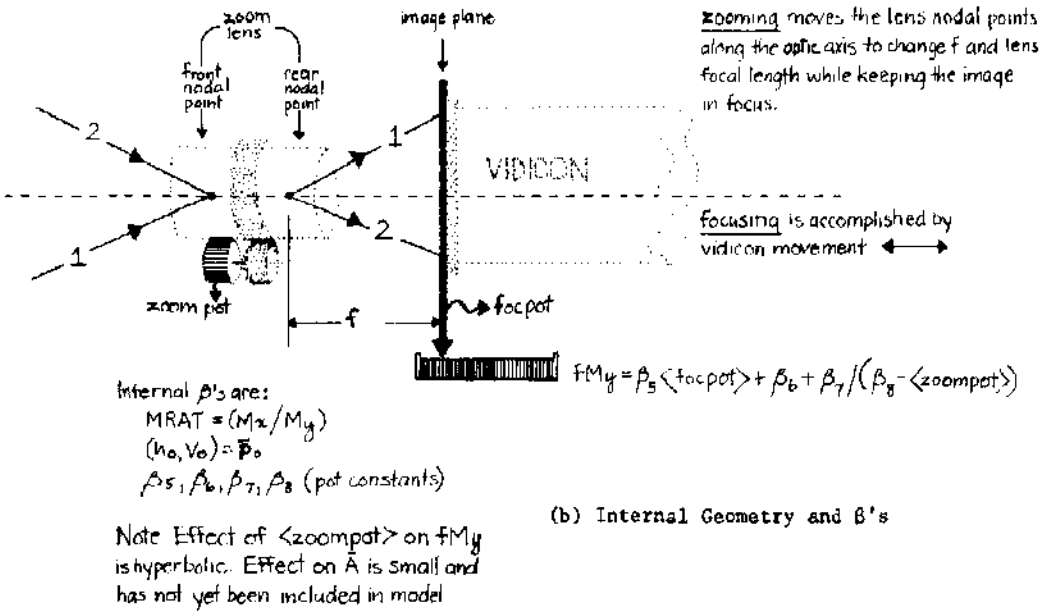


Figure 4 MOVEABLE-CAMERA TRANSFORM



(a) External Geometry and β's



(b) Internal Geometry and β's

Figure 5 THE STANFORD ZOOM CAMERA

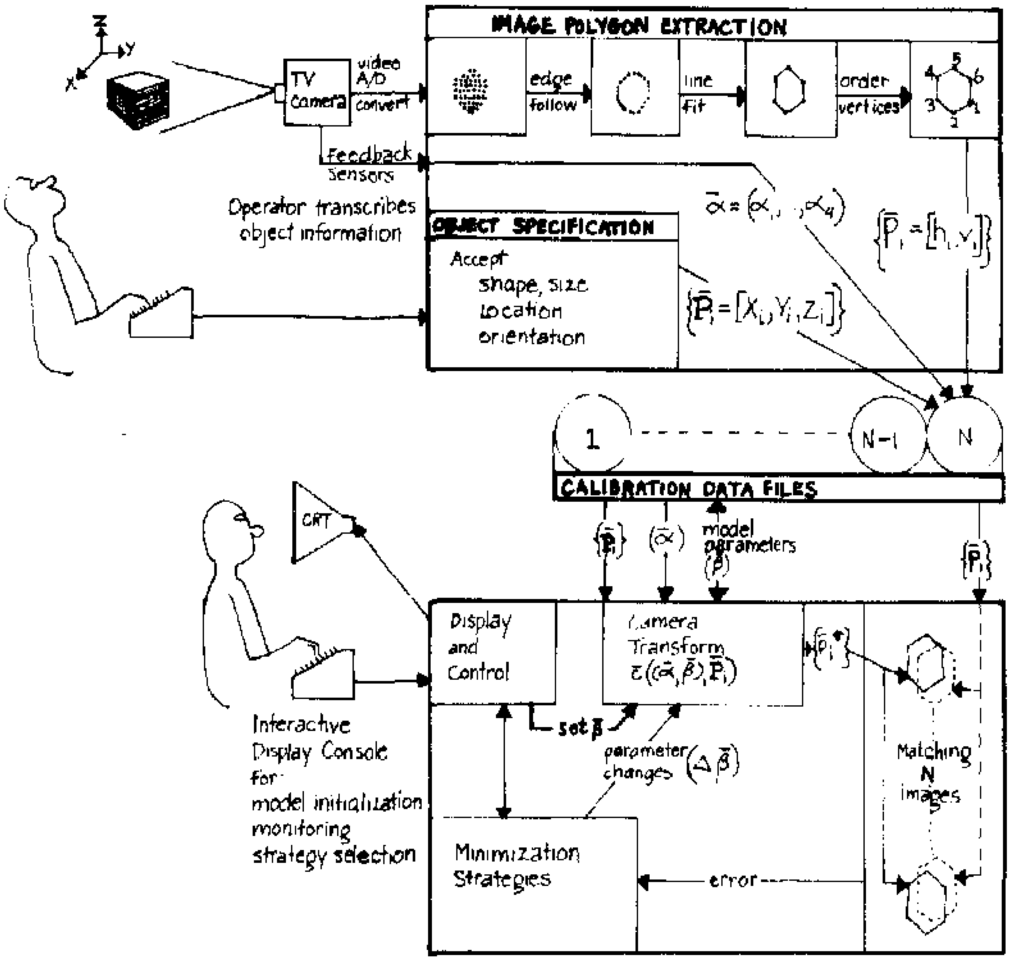


Figure 6
THE CALIBRATION SYSTEM

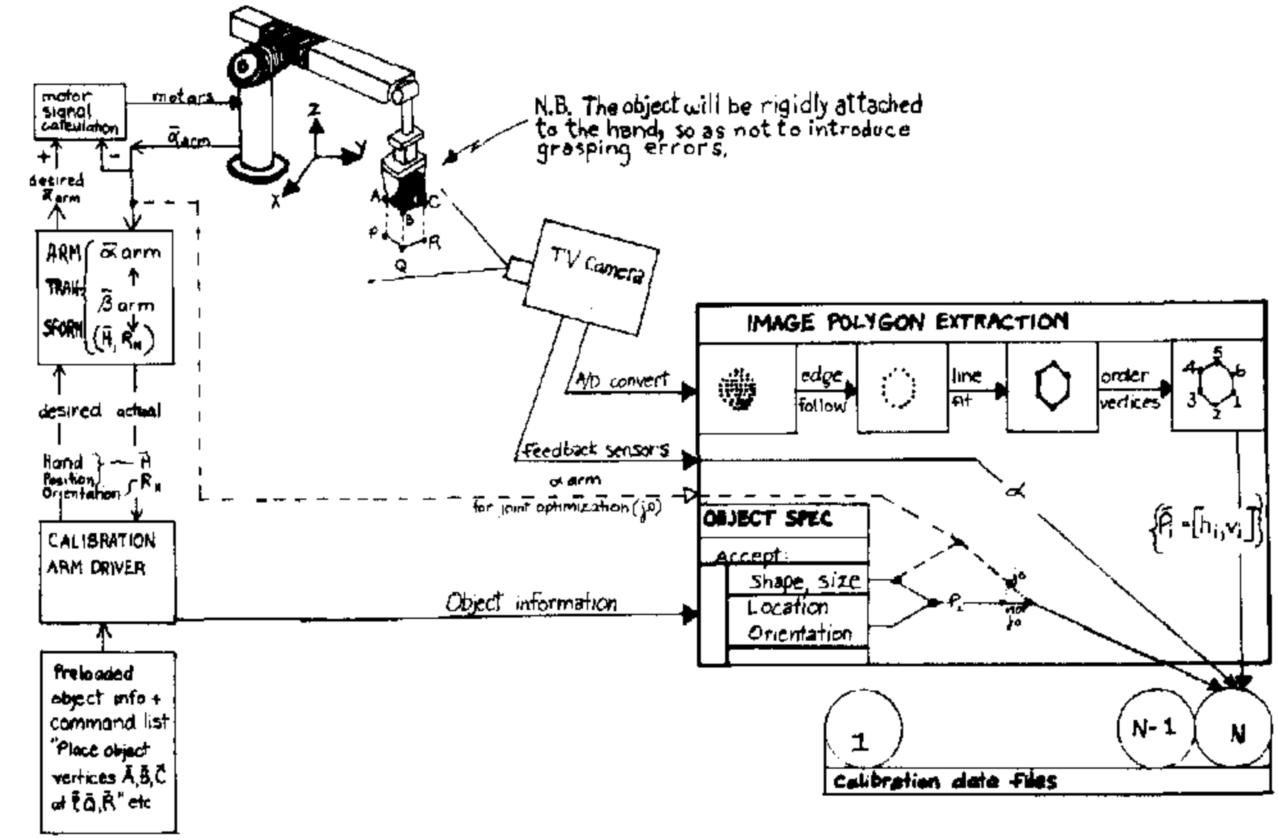


Figure 7 PROPOSED AUTOMATED DATA COLLECTION