CAMERA MODELS AND MACHINE PERCEPTION

by Irwin Sobel*

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 SD-183.

* Now at the Department of Computer Science; Technion - Israel Institute of Technology
Contents

I. Introduction page 1

II. The Geometric Camera Model page 10

III. Range-Finding and Stereo Vision page 31

IV. Model Measurement and Application page 39

V. Centering page 45

References page 51
List of Illustrations

Figure 1  Stanford Hand-Eye Arrangement  page 2
2      Camera  8
3      Finding and Centering an Object  9
4      Camera Model  11
5      Coordinate Frames  12
6      Measuring Coordinate Frame  14
7      Projection of a Point  15
8      Focusing and Zooming  21
9      Motion Analogies  23
10     New Stanford Configuration  25
11     Pan-Tilt Head Models  26
12     Stanford System Pan-Tilt Head  28
13     Stadiametrics  33
14     The Stereo Model  36
15     The Calibration System  40
16     Estimates of Depth Uncertainty  43
17     Close-Up Sequence  46
### Glossary of Symbols

<table>
<thead>
<tr>
<th>Real Scalars</th>
<th>Vectors</th>
<th>Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v_0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_x$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Z$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TILT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWING</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integer Scalars</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
</tr>
<tr>
<td>$i$</td>
</tr>
</tbody>
</table>
Abstract

We have developed a parametric model for a computer-controlled moveable camera on a pan-tilt head. The model expresses the transform relating object space to image space as a function of the control variables of the camera. We constructed a calibration system for measuring the model parameters which has a demonstrated accuracy more than adequate for our present needs. We have also shown how to generalize the model to handle small systematic errors due to aspects of pan-tilt head geometry not presently accounted for.

Applications of the model are given to the tasks of (a) binocular stereo range-finding, and (b) locating objects of interest and centering them in an image.
I. Introduction

A problem that arises in getting a computer to perceive 3-D scenes is that of relating information from several different viewpoints. In particular, if the computer moves its sensor, it has to 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. The implementation utilizes a geometric model for a moveable camera. The model expresses the transform relating the object space to the image space as a function of the variables of the camera geometry.

The Stanford hand-eye project (figure 1) is organized around a dual processor PDP-10, PDP-6 computer system. An electrically-powered mechanical arm and a standard TV camera with pan-tilt head are interfaced to the computer; they serve as the hand and eye respectively. The computer is capable of moving the arm and the camera and sensing the position of all moveable joints. Visual information is transmitted to the computer by quantizing a TV image into an array of 250 x 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 the 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 to (5,6,10,17,21,33). Two recent PhD. theses (4,31) describe other major aspects of the system. In addition, three more project theses are forthcoming (9,11,18) and are referenced in anticipation of their publication. For detailed information about manipulators, see (15,20,25).
Figure 1  Stanford Hand-Eye Arrangement
At this point we need a term to describe the type of systems we are concerned with. The results presented here apply to more than just hand-eye manipulators. They additionally encompass vehicles with some sort of image sensor for guidance. At present there is a major vision research project at S.R.I. centered around such a vehicle (16). At the Stanford Artificial Intelligence Project a somewhat less elaborate vehicle (26) has also been constructed for the purpose of investigating the problems of an automatically driven automobile. We will be constantly referring to the whole system consisting of visual sensor, drive motors for moveable parts (arms, wheels, etc.), position feedback sensors; and most important, a large computer with a data structure and a set of programs for processing visual information and deciding how to accomplish a task. Mainly for convenience and also for lack of a better term we use "robot" for this purpose. Such a machine must have a way of exploring its environment. In what follows we show how this requirement, coupled with the need to complete a task within a reasonable time interval dictates that the robot be very selective in its choice of what regions to explore. Moreover, if a camera is used for exploration, then a camera-model is needed to relate information gathered from different views. Finally, we show that a camera model can significantly reduce the amount of searching needed for objects whose location is known or even partly known.

One principal task in the design of a robot is provision of the ability to acquire, store, and use information about its environment. The internal representation of the various properties (size, shape, location, color, etc.) of objects is called the robot's "world model". The particular representation now used at Stanford is described by Paul (17). The process of acquiring the
information for this model will be called "map-making". This is related to the process of map-making from photographs but is somewhat more general.

To carry out map-making in a general context will require both large memory and long calculations. Any particular machine will have some fixed data-processing rate. If it is to be useful, it must be able to accomplish a given task within some prespecified time interval. Thus, if its environment is complex enough to be interesting, the robot will have to be clever about getting the information it needs. It will have to make good use of its knowledge of the environment and the task at hand to selectively orient and tune its sensors. We call this process 'accommodation'. The word is borrowed from visual physiology where it is used specifically to denote focusing.

Gibson (7) presents several interesting examples of accommodation in humans. He points out that human perception does not merely involve the passive description and classification of sensory phenomena, but also the active direction of attention to best achieve some goal.

We conclude from this that our robot should have a mapping ability and should concentrate on accommodating for, and mapping, those regions of space necessary for any given task. There is the additional problem that for a moveable camera sensor, each setting of the camera's position and adjustments gives rise to a separate image. Thus, we are faced with the task of making coherent sense of the fragmented information from several images.
Map-making from photographic images is the realm of photogrammetry. Photogrammetrists solve the problem analytically through use of a geometric camera model (1). The model is used to derive expressions for a camera transform which associates an image point with every position in the camera's field of view, and conversely, a ray in space with each image point. If we adopt such a geometric model the transform can be computed from measurements of the camera's position, orientation, and internal dimensions. There will be one such transform for each image. The collection of images will be integrated into a map.

Suppose we rotate a camera about a single viewpoint and take enough pictures to cover all viewing directions. We can then construct a 2-D map of the surrounding area. The map will give the direction in space to every point of interest visible from the viewpoint. On the other hand, if complete 3-D information is desired about the size, shape, and location of objects in the region of interest, a single 2-D map is not sufficient. In the first place, many surfaces are blocked from view by others. Further, image plane coordinates only identify a direction in space of a point relative to a viewpoint - range information is missing. To get a complete 3-D map of the environment, it is sufficient for every point to be identified in 2 images taken from different viewpoints. In part IV we discuss the problem in detail. We describe this method for getting 3-D information and also list several other approaches.

The most difficult part of map-making is accurately ascertaining camera-transforms. Not surprisingly, a good part of the work described here is devoted to...
We assume at this point that we have a robot with a moveable imaging sensor and a camera model to integrate information from different views into a 3-D world model. Our robot for efficiency in finding things, will make use of previously stored information to direct its search. This is accommodation - orienting and tuning its sensor according to some strategy. Orienting and tuning can be made more precise in terms of the accommodative-variables - those sensory parameters under direct control of the robot - e.g., camera position, orientation, magnification, focus, aperture, sensitivity, etc. The problem is that of looking in the most probable place for what one wants to find; where "looking" becomes generalized to mean selection of the proper region in the space of accommodative-variables. To do this efficiently we need a model which predicts the effects of changes in the "a-v's" on the visibility of what the robot is trying to find. Tenenbaum (31) has developed models of this sort for digitized video data from a TV camera sensor. He was concerned with predicting the effects on feature visibility of changes in the following:

- focus
- lens-aperture
- sensitivity
- spectral range (using color filters)
- quantizer digitization window

We are concerned here with the complementary problem of predicting the effects on feature location (i.e., image-coordinates) of change in:

- focus
- pan
- tilt
- zoom
To make things clear, consider a robot with a zoom camera on a pan-tilt head (figure 2). A common task that arises is to locate an object of known shape whose position in 3-D reference coordinates has previously been specified. (This is not always the case. Many times there is a need to look at an interesting feature without any prior coordinate information. This is discussed in detail in part V.) A plausible subtask statement is to pan and tilt the camera until the image of some point on the object is centered in the field of view. The camera may then be zoomed to any desired magnification. The problem of how much to change the pan and tilt angles to center the chosen object point can be solved a-priori with the use of the 3-D reference coordinates of the point and a calibrated camera model. If such a model is not available, specification of the reference coordinates of the object is of absolutely no use in finding and centering it. In this case a search must be undertaken until the object falls into the field of view and is recognized. Its image must then be centered using a time-consuming optical-tracking servo program. This is illustrated in figure 3.

In a system using a camera model for predicting pan and tilt angles, the predictions will usually be somewhat in error and a servoing program will have to be used to correct for this. The more accurate the model prediction, the less servoing is needed. Minimization of the prediction error - measured by the servo program - can be used as a criterion for refining the model parameters. We will say more about this in part V.
Figure 3  Finding and Centering an Object

(c) Centering

(b) Acquisition

(a) Search
II. The Geometric Camera Model

We begin this presentation with (a) a simple model used not only in photogrammetry, but also more recently in computer graphics and vision work. We then show (b) how effects of focusing, zooming, and changing lenses can be accounted for by this model. Finally, in (c) we express the model in parametric form for a moveable camera on a pan-tilt head. The model does not account for distortions in images due to lens aberrations or TV camera and computer electronics. These have not shown themselves to be major sources of error in our work. If they do become a problem, they can be measured and compensated for.

(a) Model for a Single View:

A camera can be thought of as forming an image which is a point projection of a field of view. This is schematically represented in figure 4 by a point and a plane. The point C is known as the 'lens-center' or center of projection. The plane which we call the 'ideal-image' plane is a reflected version of the physical image plane (shown dotted in the figure) formed by the lens. The ray through C normal to the ideal-image plane is called the principal-ray of the camera. The point where the principal-ray pierces the ideal-image is called the 'principal-point' of the image. Imagine for any point in view, the ray from the lens-center to the point in question. The ideal-image of the point is the intersection of this ray with the ideal-image plane.

We can define the location and orientation of the camera in space by introducing two coordinate frames into our model (see figure 5). First, we define a reference \((X,Y,Z)\) frame with origin at S which corresponds to the physical space. C will now be taken to be the coordinate vector of the lens-center in this reference
Figure 5 Co-ordinate Frames

Reference Frame

Camera Frame

Principal-Ray

Principal-Point

Ideal Image
frame. We next define a camera \((x,y,z)\) frame at \(C\) with

1) \(z\)-axis pointing along the principal-ray toward the ideal-image plane.

2) \(x\)-axis parallel to and having the same sense as the image-horizontal. (Image-horizontal is defined below.)

3) \(y\)-axis defined such that \(x,y,z\) axes constitute a right-handed frame.

The orientation of the camera is the \(3 \times 3\) rotation matrix \([R]\) whose columns are direction cosines of the camera basis vectors relative to the reference basis. Recall that all rotation matrices are orthonormal and thus \([R]\) satisfies the condition \([R][R]^t = [I]\) or alternatively \([R]^t = [R]^{-1}\).

The image on which measurements are made is a scaled version of the ideal-image. The measuring coordinate system is chosen to originate at the upper left corner of the view window. In our TV system the view window is defined by the scanning rectangle. The coordinate axes are chosen parallel to the sides of this window and designated 'horizontal' and 'vertical'. A general image point \(p\) has coordinates \([h,v]\). The principal point is labelled \(p_0 = [h_0,v_0]\) in this system (see figure 6).

We will specify the transform relating the measured coordinates \(p\) of the image of a point with its reference coordinates \(P\) in two steps: (figure 7). First, we will

* In evaluating image-coordinates, correction to vertical coordinates (not shown here) is made for the fact that the horizontal TV scan is slightly inclined from the true horizontal axis. The change is small (at most 1 resolution unit) and can be ignored for many purposes. \(V_{\text{act}} = V_{\text{meas}} + h/333\) gives the correction.
Figure 6: Measuring Coordinate Frame

\[ y = (y - y_0) / M_y \]
\[ x = (x - h_0) / M_x \]

Coordinates:
Ideal - Image

Coordinates:
Measured - Image
Figure 7: Projection of a point
express \( p \) in terms of the camera frame coordinates \( P' \) of the point, then we will express \( P' \) in terms of the reference frame coordinates \( P \) for the point.

If \( P' = [x, y, z] \) represents camera coordinates for the point then by a similar-triangle calculation its projection on the ideal-image plane is given in the same frame by

\[
[x(f/z), y(f/z), f] = (f/z)P'
\]

where \( f \) is the distance from \( C \) to the ideal-image plane. \( f \) is called the "principal-distance" of the image. The first two components of this vector represent the ideal-image coordinates relative to the principal point. In practice, these are not measured directly. Measured image coordinates differ from the ideal in one or both of two aspects (ignoring distortion):

(i) **Scaling**: In photographic systems uniform enlargements or reductions in the developing and printing process can be characterized by a magnification factor \( M \). In sampled data systems the coordinates of a point do not represent distances but sample numbers. These are related to distances in the ideal-image by a sampling interval which can also be represented by a scale factor. In our system we allow for different sampling intervals in the \( x \) and \( y \) directions and describe them by \( M_x \) and \( M_y \) respectively.

(ii) **Translation**: The measuring origin is displaced from the principal-point which then becomes \( \mathbf{p}_0 = [h_0, v_0] \).

By applying the scaling by \( M_x, M_y \) and then the translation by \( [h_0, v_0] \) to the ideal-image coordinates \( (f/z)[x, y] \) we arrive at the desired expression

\[
p = [h, v] = (f/z)[(M_x)x, (M_y)y] + [h_0, v_0]
\]

for the coordinates of \( p \) in terms of those of \( P' \). It is convenient to redefine \( p \) in homogeneous coordinates (see 23, 24) to be \( p = [h, v, 1] \). This representation is invariant under scalar multiplication. The \([h, v]\) form can always be recovered by
dividing the vector by the multiplier which is carried as the third coordinate. With
p in this form we can write the above equation in matrix form as

\[
\begin{bmatrix}
  h \\
  v \\
  1
\end{bmatrix} = \frac{1}{z} \begin{bmatrix}
  fM_x & 0 & h_0 \\
  0 & fM_y & v_0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} \equiv \frac{1}{z} \begin{bmatrix}
  p'_x \\
  p'_y \\
  p'_z
\end{bmatrix}
\]

where we call the 3 × 3 matrix [INT] the "interior-orientation" of the camera.

Note that f,M_x,M_y,h_0,v_0 all relate the image measurement frame (h,v) to the 3-D
camera frame (x,y,z) at C.

We can now complete the specification of a camera transform by relating the (x,y,z)
frame at C to the reference (X,Y,Z) frame at S. Such a relation can be expressed
by an appropriate rotation in 3-space and a translation by the vector C. The
rotation is simply the orientation matrix [R] defined above. The relation can be
written as

\[
P' = [R](P-C) = [R]P - [R]C
\]

Recognizing that -[R]C is the reference frame origin S as seen from the lens-
center we can rewrite this as

\[
P' = [R]P + S
\]

If we again introduce homogeneous coordinates by redefining P = [X,Y,Z,1] we can
concisely represent all this by a single 3 × 4 matrix as
We call $[\text{EXT}]$ the "exterior-orientation" of the camera. Finally, we can use homogeneous coordinates to express the overall camera transform from a point $P = [X,Y,Z,1]$ to its image $p = [h,v,1]$ as a linear transformation.

$$p = \frac{1}{z} [\text{INT}] [\text{EXT}] P = \frac{1}{z} [\text{CAM}] P$$

where $3 \times 4$ product matrix $[\text{CAM}]$ is called the "camera transform". If we restrict $P$ in the last equation to just those points in the $z = 0$ or "ground-plane" of our reference frame, we can drop the third column of $[\text{CAM}]$ leaving a $3 \times 3$ matrix $[\text{COL}]$ satisfying

$$\begin{bmatrix} h \\ v \\ 1 \end{bmatrix} = \frac{1}{z} [\text{COL}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$[\text{COL}]$ is a 1:1 transformation associating a unique point in the ground-plane with every point in the image plane. It is called a collineation and is uniquely determined by 4 point pairs $\{(X_i,Y_i), (h_i,v_i)\}_{i=1}^4$. The calculation is a standard one in projective geometry, and is given in (29).

The inverse of $[\text{COL}]$ defined $[\text{COL}]^{-1}$ is useful when we want to find the ray in reference coordinates associated with the image point $p = [h,v,1]$. If we compute
the vector \([\text{COL}]^{-1}p\) and "normalize" it by dividing by its third component to get the result in the form \([X,Y,1]\) then \([X,Y]\) represents the intersection of the desired ray with the ground-plane. The point \([X,Y,0]\) and the lens-center \(C\) define the ray corresponding to the image point \([h,v]\).

\([\text{COL}]\) and its inverse have been heavily used in computer visual systems. The main reason for this is the ease with which they can be calculated. The geometrical parameters \(C,[R],f,M_x\) etc. described above need not be explicitly measured. Instead, 4 \([X_i,Y_i]\) ground-plane points in the field of view are marked by some easily detected visual feature - at Stanford the right angle of a dark right triangle was used on a light background - and their ground-plane coordinates are manually typed to a calibration routine. Next, the TV image is searched for the features and their image-plane \([h_i,v_i]\). The required \([\text{COL}]\) matrix can be solved for (see 29) by applying some straightforward linear algebra techniques to the homogeneous coordinates of the point pairs.

(b) Focusing, Zooming, and Changing Lenses:
The camera model described above does not take lenses into account. It assumes that cameras are simple pinhole devices with the pinhole being located at the lens-center. Ideally, the principal distance \(f\) accounts for effects on the image coordinates of focusing, zooming, and lens changing. Focusing amounts to moving either the lens or image plane along the optical axis of the camera until the desired region is in sharp focus. Tenenbaum (31) at Stanford and Horn (12) at MIT have both independently developed algorithms for doing this automatically with a computer 'eye'. The direct effect on our model of these focusing motions is to
change $f$ and thus the scale of the image coordinate system. Zooming amounts
to continuously changing the focal length of a lens while keeping its focus
contant. The effect is to move the lens-center along the optical-axis relative
to the image-plane. This again shows up in the model as a change in $f$. With
focusing, the change in $f$ is usually considered a secondary effect, while with
zooming it is the desired primary effect.

The effects of focusing and zooming are illustrated in figure 8. Changing
lenses and refocusing are equivalent to zooming to a new focal length. Because
$f$ can be changed by both focusing and changing lens-focal-length, it is convenient
to think of it as having two additive components - one due only to focal length
changes and the other due only to the effects of focusing.

In practice, lens changing and zooming also have the second-order effects of
moving the principal-point about in the image-plane. This is not a serious
problem for our calibration system which has to deal with a small number of
fixed focal length lenses. We just store a list of principal-point coordinates
indexed on lens number. For a zoom lens principal-point trajectory in the
image plane as parametrized by focal length can be measured and stored to the
required accuracy.

At this point it must be mentioned that zooming (or lens changing ability) is
not of much use unless there is some facility for pointing the camera so as
to center the feature of interest in the field of view (see figure 3). This
focusing: changes \( f \) while holding \( d \), \( f_l \) constant till the point of interest at \( d \) is in sharp focus. The lens equation relates these distances for the sharp focused situation:

\[
\frac{1}{f} + \frac{1}{d} = \frac{1}{f_l}
\]

zooming: changes \( f \), \( f_l \) simultaneously while keeping the image in sharp focus. The lens equation holds throughout the process.

\textbf{Figure B} Focusing and Zooming
brings us to the next part of the modelling problem.

(c) Model for a Camera on a Pan-Tilt Head:
A pan-tilt head provides a camera with the same type of pointing ability that human neck muscles provide for a person's head (figure 9(a), (b)). The anthropomorphism can even be extended. Our computer visual system has the ability to select any rectangular subfield of the TV camera's field of view for processing. This selection can be made in one TV frame time (1/30 second). If we consider moving a small subfield from the right image boundary to the left image boundary in one frame-time with a typical 25 mm. focal length lens ($\frac{1}{2}$ radian field of view), the effective angular velocity is 15 rad/sec. Thus, if we associate panning and tilting movements with gross head movements, subfield selection corresponds in speed and range to the movement of the human eye within its orbit (figure 9(c)). This analogy is meant only to be geometrical and mechanical rather than functional. There are, of course, many other things involved in both systems and a complete discussion is beyond the scope of this paper.

Note that just as head movement produces both a rotation of the eye and a translation, pan-tilt head movement changes not only the direction of view but the position of the viewpoint (lens-center). Both situations are accounted for by the fact that the center of rotation and the lens-center do not coincide. This has the disadvantage of complicating the modelling problem, but offers the advantage of a limited stereoscopic range-finding ability (cf. part IV).
Figure 9: Motion Analogies

(c) Subfield motions

(b) Panning—Top view

(a) Tilt—Side view
So a pan-tilt head provides a restricted type of camera motion. To have a complete mapping capability it is desirable to be able to see any point in the robot's 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. At Stanford, we are presently implementing a system with two cameras on pan-tilt heads, placed around a "lazy-susan" rotating work platform as shown in figure 10. All motions of the cameras and "lazy-susan" are under computer control. They can be thought of as forming a set of accommodative variables. The general modelling problem is to be able to predict the effects of changes in these variables on the perceived scene. We proceed with the model for a camera on a pan-tilt head. This approach also applies to cameras on vehicles - e.g., the Stanford and S.R.I. carts mentioned earlier.

One can look at the pan-tilt head as a simple form of manipulator having two degrees of freedom - the angles of PAN and TILT. A "manipulator" for our purposes can be thought of as a chain of articulated links with variable joint angles (see figure 11(a)). Pieper (20) has considered the problem of finding the position and orientation of the free end of a manipulator given its joint angles, and the more difficult inverse problem.
Figure 10 New Stanford Configuration

Rotating work platform
Figure 11 Pan-Tilt Head Models
In terms of our camera model the pan-tilt head has the effect of changing the lens-center C and orientation [R]. To describe this motion it suffices to express [R] and C as functions of PAN and TILT. We will restrict consideration here to an ideal pan-tilt head, where PAN and TILT axes intersect at a point P₀ and are orthogonal (figure 11(b)). In addition, the pan-axis is assumed parallel to the Z-axis of our reference frame. In the case of [R] we start by choosing a reference orientation of the camera in order that the working range of PAN and TILT will be in the interval [0,π]. This orientation for our system is shown in figure 12 and can be represented as a reflection and relabelling of coordinate axes [R₀]; it takes X → -z, Y → x, and Z → -y. This relates the reference frame [X,Y,Z] orientation directly to a camera frame [x,y,z] with PAN = TILT = 0. PAN is defined to be the angle of rotation of a camera frame translated to P₀ about the y-axis in the direction x → z. TILT is defined as a subsequent rotation of the camera frame about the once-rotated x-axis in the direction z → y. For completeness it is convenient to define a third rotation, SWING, about the twice rotated z-axis in the direction y → x. The rotations PAN, TILT, and SWING are represented by matrices [RP], [RT], and [RS] respectively. We can express [R] in terms of these elementary transformations as [R(TILT,PAN)] = [RS][RT][RP][R₀]

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(TILT) & -\sin(TILT) \\
0 & \sin(TILT) & \cos(TILT)
\end{bmatrix}
\begin{bmatrix}
\cos(PAN) & 0 & \sin(PAN) \\
0 & 1 & 0 \\
-\sin(PAN) & 0 & \cos(PAN)
\end{bmatrix}
\]

[R₀] and [R₀] are fixed. [RT] and [RP] have been expanded here to show their dependence on TILT and PAN. [RS] is used to account for any slight SWING misalignment in the camera.
Figure 12: Stanford System Pan-Tilt-Head
Reference orientation $[R]$ for PAN=0, TILT=0
We can use \([R(\text{TILT}, \text{PAN})]\) to describe the motion of any point on the camera body. A point \(P_c\) on the camera body has reference coordinates \(P\) given by:

\[
P = P_0 + [R^t(\text{TILT}, \text{PAN})]P_c
\]

where \([R^t(\text{TILT}, \text{PAN})]\) is the transpose of \([R(\text{TILT}, \text{PAN})]\). The coordinates of \(P_c\) are measured with respect to a camera frame translated to \(P_0\). In particular, if we set \(P\) equal to the lens-center \(C\) we get an expression for \(C(\text{TILT}, \text{PAN})\) in terms of \(P_0\) and an offset vector \(DP\) (figure 11(b)).

\[
C(\text{TILT}, \text{PAN}) = P_0 + [R^t(\text{TILT}, \text{PAN})]DP
\]

We need only measure \(P_0\) and \(DP\) once. Thereafter, if the camera is moved the camera transform can be quickly updated from the angles \(\text{PAN}\) and \(\text{TILT}\) using the above expressions for \(C\) and \([R]\) to recalculate the camera's exterior orientation.

This simple model has so far proved adequate for our experiments. If greater accuracy is required it may be necessary to account for second order effects such as:

- non-intersection of pan and tilt axes
- the pan-axis not being perfectly parallel to the reference z-axis
- the tilt-axis not being orthogonal to the pan-axis.

All these deviations can be handled by the more general 3-link arm model illustrated in figure 11(a). For more details the reader is referred to Pieper (20). An extended discussion of the use of the simple model for centering is given in part V of this paper.
At this point it is interesting to note the improvements brought about in the introduction of this model to the Stanford hand-eye system. In the original system (21) any time the camera was moved, 4 reference marks had to be manually placed on the table-top (ground-plane) in the field of view and their reference coordinates typed to a calibration routine. This routine then analysed their images and computed a collineation for the new view as described in II(a) above and (29). The inverse collineation gave two reference coordinates for points lying in the ground-plane. Reference coordinates for points above the ground-plane could only be deduced for certain features which the system had extra-knowledge about, such as corners of objects known to be cubes sitting on the ground-plane. The need for manual intervention every time the camera was moved, effectively nullified the advantage of having the pan-tilt head under program control.

At present, with a calibrated pan-tilt head the system is able to update its collineation immediately after moving the camera without any need for manual intervention. In addition, availability of the lens-center coordinates C provides the basis for a general stereo range-finding ability. Before we describe how this is done we digress to give some background on the general problem of range-finding.
III. Range-Finding and Stereo Vision

As mentioned in the introduction, a single viewpoint only serves to identify a direction in space to a point of interest. If we have an inverse collineation and a lens-center C for a given view we can calculate the reference-frame expression for the ray which a point P must lie on as was described in II(a). The distance from P to C is defined to be its "range" from the camera. We distinguish range from "depth" here which we define to be the camera-frame z coordinate of the point as shown in figure 7. In many cases, range is not the quantity of interest, but depth or more often the reference-frame coordinates P of the point. These quantities are related in a straightforward way.

If we know the range or depth of a point and its image coordinates we can use the inverse interior-orientation of the camera \([\text{INT}]^{-1}\) to locate it in the camera-frame as \(P'\). We can then transform to reference frame coordinates using

\[ P = R^t P' + C. \]

We will now mention some methods of getting range or partial range information. The last of these will be the binocular-stereo method which lends itself particularly well to our formulation of the problem. Several of the techniques listed are usually discussed as depth cues in texts on vision, psychology, and psycho-physiology. For a more complete discussion of these the reader is referred to Gibson (8).

(1) **Ground-Plane Assumption**: If objects of interest lie on a common plane a one-one collineation between this plane and the image plane can be calculated. One can then determine the exact location of all points in the ground-plane from their
images. If actual range to the camera is wanted, the position of the camera lens-center must be known. This method has been widely used in the past, in computer visual systems.

(2) **Lidar**: A 2-axis deflectable Laser range finder is being considered (3). Such a device could also be constructed to give reflectivity information. Thus, providing us directly with all 3 camera-frame coordinates for a point.

(3) **Stadiametric Measurements**: If some dimension of an object is known and it lies parallel to the image plane, its depth can be computed from the image-coordinates and interior orientation parameters (figure 13). Note that this procedure requires recognition of the object in question and prior knowledge of the object.

(4) **Focus**: The depth of a point can be calculated from the lens-equation after sharp focusing on the point and measuring the resulting principal-distance \( f \). The smaller the lens focal ratio ("f/number") used, the better the depth resolution attainable for a given lens. Note that a camera can do significantly better than humans at this because camera lenses can typically achieve much smaller f/numbers than human eyes. Horn (12) and Tenenbaum (31) have independently developed focus-ranging systems.

(5) **Perspective**: This is a term often used in vision psychology to cover a whole class of facts about images which can be deduced from the assumption that the image is a point projection of a 3-D space. The simplest and most frequent example is
Simple case: Known dimension D is parallel to y axis. The generalization is straightforward. Ideal image: \( y_2 = y_1 \). Line length in ideal image: \( y_2 - y_1 = \frac{\sqrt{\left(x_2 - x_0\right)^2 + f^2}}{f} \).

By similar triangles, ranges to \( P_2 \) is:

\[
z_2 = \frac{\sqrt{y_2^2 + f^2}}{y - y_1} = \frac{\sqrt{\left(x_2 - x_0\right)^2 + f^2}}{f} = 2
\]

Figure 13: Stadiametrics
the image of a set of lines known to be parallel. The convergence of the image lines allows one to order points on them according to depth.

(6) **Interdiction**: If an object is partially or wholly blocked from view relative depth information can be deduced. It could be recorded in the form of relations such as "point A is in front of (behind) point B".

(7) **Texture Gradients**: This can be thought of as another case of use of perspective information. Here, increasing spatial frequency of the image of a surface having a regular texture denotes increasing depth.

(8) **Relative Rates of Motion**: This covers a class of relative depth measurements which can be made by a moving observer in a stationary environment. In particular, objects which are close to him will have higher velocity components normal to the line of sight. Since the environment is assumed stationary this imposes a depth ordering on the space.

(9a) **Monocular Parallax**: This can be viewed as a special case of stereo vision. The term is used whenever views are generated sequentially by the same camera (or viewer) in different positions. In humans, this occurs whenever the observer is moving relative to the object of interest; such motion being caused either by voluntary body movement or independent object motion or both.

(9b) **Binocular Stereo Vision**: The method we will describe yields reference coordinates of points which are visible in two different views. The stereo
problem can be subdivided as follows (see figure 14).

(i) **Point Choice:** Identify an image point of interest in one of two views: Say \( p \) in view a of figure 14. It is necessary to make this identification on the basis of visual characteristics of the image. Picking an arbitrary set of image coordinates is not sufficient for correlation.

(ii) **Correlation:** Finding the corresponding image point in the second view. The correspondence is established on the basis of visual properties - e.g., texture, color, edges, corners - which the two images have in common. In the example of figure 14, \( p_b \) in view b is the correspondent of \( p_a \). They are the images of the corner of a block and can be identified (not necessarily uniquely) by three intersecting edges.

(iii) **Reconstruction:** From the image coordinates of the correlate points \( p_a \) and \( p_b \) calculate reference coordinate \([X,Y,Z]\) of the point \( p \) giving rise to the images.

We assume here that the first step of point choice has already been accomplished by some higher level decision process and restrict this discussion to the other two steps. We will treat reconstruction first. It is a straightforward application of the camera model. The idea is to express the rays \( \text{ray-a} \) and \( \text{ray-b} \) containing \( p_a \) and \( p_b \), in the reference frame using the camera calibration information. \( P \) is theoretically the intersection of these two rays. Due to errors in measurement these rays will usually not intersect. In practice, a "best" intersection, as defined below, is chosen.

The rays \( \text{ray-a} \) and \( \text{ray-b} \) can be expressed in terms of the lens-centers \( C_a, C_b \) and the inverse collineations \([\text{COL}_a]^{-1}, [\text{COL}_b]^{-1}\). First we use the inverse collineations to find the ground-plane intercepts \( P_a = [X_a, Y_a, 0] \) and \( P_b = [X_b, Y_b, 0] \) as described in II(a). The desired rays are specified by the two point pairs \( C_a, P_a \) and \( C_b, P_b \). The equations in parametric form are:
Figure 34: The Stereo Model
\[
\text{ray-a}(t_a) = t_a(P_a - C_a) + C_a \\
\text{ray-b}(t_b) = t_b(P_b - C_b) + C_b
\]

where \(t_a\) and \(t_b\) select specific points on the rays, they can be thought of as unnormalized range measures, having values 0 at the lens-centers \(n\) and 1 at the ground-plane. The solution for "best" intersection mentioned above is given in the appendix of (27). If the rays intersect, it gives the intersection. If they are skew, it gives the point midway between them on their common normal.

The camera transforms can be used to greatly reduce the search for the correlate \(p_b\) of an image point \(p_a\). The search can be constrained to a straight line in image \(b\) through use of the following observations. Notice (see figure 14) that knowledge of \(p_a\) constrains the unknown \(P\) to lie on ray-a given above. If \(P\) lies on ray-a, its image \(p_b\) must lie on the image of ray-a in camera \(b\). This is the required straight line. The line equation can be calculated by applying the \(b\) camera transform \([\text{CAM}_b]\) to the ray-a equation above. The result is the line in the \(b\) image parametrized by the range parameter \(t_a\). In particular, the ray-a equation will be of the form \([X(t_a), Y(t_a), Z(t_a)]\). To be consistent with our definition of \([\text{CAM}_b]\) we add a fourth coordinate of 1 to this, putting it into homogeneous representation \(\text{ray-ah}(t_a) = [X(t_a), Y(t_a), Z(t_a), 1]\). The line equation will then be

\[
P_b(t_a) = \begin{bmatrix}
    h_b(t_a) \\
    v_b(t_a) \\
    1
\end{bmatrix} = \frac{1}{z_b(t_a)}[\text{CAM}_b]
\]

\[
\begin{bmatrix}
    X(t_a) \\
    Y(t_a) \\
    Z(t_a) \\
    1
\end{bmatrix}
\]
Any a-priori knowledge of range can be used to bound $t_a$ and, thereby, shorten the search for $p_b$. At very worst, the point will always be between the ground-plane and the lens-center. Thus, $0 \leq t_a \leq 1$. Usually, the condition that the point lies within some predetermined work space can be used to bound $t_a$ tighter. At best, a range estimate with accuracy bounds from focus information (see 31) will significantly shorten the allowable interval for $t_a$. The correlation search can be further reduced if, as is usually the case, $p_a$ is an edge point; that is, if $p_a$ lies on the visible boundary between two regions. In this case, an edge detector can first extract edge points along the line $p_b(t_a)$ within the allowable interval. In most situations, there will only be a few of these, say $n$, that lie on the line. $p_b$ may then be chosen from these $n$ points by maximizing a local correlation operator. In fact, if there is enough "point information" (i.e., bits of intensity and/or color) this may be sufficient to make the final decision.

Note that this scheme requires that the properties used to find $p_b$ are invariant from view a to view b. These properties must also have sufficient spatial resolvability to yield accurate image coordinates. When neither of these assumptions hold, the stereo process will fail. In general, this happens with smoothly curved surfaces having little or no texture or identifying marks. Horn (13) has recently presented a scheme for using shading information from directional lighting to deduce shape in just these situations. The scheme is complementary to stereo vision and should succeed in many of the situations where stereo fails.
IV. Model Measurement and Application

The biggest problem in implementing a system of the type just described is measurement of the camera parameters. They can be roughly approximated from manual measurements of the physical dimensions of the camera. However, since accurate predictive ability can significantly reduce the time spent in searching, it paid us to design a calibration system which is capable of choosing an optimal set of camera parameters. The criterion chosen for optimality was minimization of prediction error. A more complete description of the system will appear in a forthcoming paper (28) and is also available in (27). We will briefly outline it here.

The calibration system was subdivided into two main parts: data collection and model optimization (see figure 15). The data used were images of a calibration object along with detailed specification of its shape and location in the reference coordinate frame. The images were processed by an edge-follower program (14,22), which extracted points lying on the outer boundary of the object. A polygon was then fitted to the edgepoint set. The vertices of this polygon were ordered and associated with the reference-frame coordinates of the corresponding object vertices. This yielded as basic data for model optimization

\[ P_i = [X_i, Y_i, Z_i] \] are object vertex coordinates

\[ \{P_i, P_i\}_{i=1}^{6n} \] where

\[ p_i = [h_i, v_i] \] are their corresponding image coordinates

and \( n \) is the total number of images supplied (all the objects were chosen to give 6-sided image polygons).

The images were chosen to give a good sampling of both the working range of the pan-tilt head and possible locations within the image rectangle. The data were
Figure 15
The Calibration System

(a) DATA COLLECTION

(b) MODEL OPTIMIZATION
written on special calibration data files - one per image for later use in model optimization. Each file also provided storage for a camera model which could be read and updated by the model optimizer. Data collection is illustrated in figure 15(a).

The model optimization routine (figure 15(b)) used information from a number of images to find a "best" camera model. It was initialized with estimates of the camera parameters. Based on these, it made a prediction of the image of each reference point . The predictions were then compared with the coordinates found by the image processing routines and an overall mean square prediction error

\[ E = \frac{1}{6n} \sum_{i=1}^{6n} (p_i - p_i^*)^2 \]

was calculated. A display-based interactive system (27,28) was built which allowed the user to select one of several strategies for changing the camera parameters to find a minimum value of the mean square error function. Note here that the camera parameters enter into the error measure through the predictions . The original algorithm used was a direct-search procedure. For a review of such procedures, the reader is referred to (30). Later, a more efficient algorithm designed by Brent (2) was added to the system. Usually, the combination of the two worked better than either one alone. The algorithms were chosen for their generality rather than for efficiency. The idea was to be able to easily change optimization criteria without having to supply partial derivatives which most efficient algorithms (e.g., Newton-Raphson) require.
After the optimization algorithm converged the usual residual errors were about 1-1.5 resolution units rms.

An operational test of this system for stereo vision was initially made with a fixed second camera in addition to the one on the pan-tilt head. An L-shaped test object was used. Models for both cameras were generated from a reference sequence of about 10 images per camera. One stereo pair of reference images - one from each camera - was used for the test. The problem of correlating the vertices of the images was solved by using the calibration system's orientation program to order vertices in both views from its knowledge of the calibration object. The reference coordinates obtained from stereo reconstruction were compared with measured coordinates. The errors ranged from .04" to .09" for the six vertices of the object. The object was roughly 30" from the camera.

An estimate of overall stereo-prediction accuracy can be made from knowledge of the residual rms. errors of the two camera calibrations. Each of these, as a first approximation, can be used to define an uncertainty cone about the ray predicted in the triangulation. The intersection of these cones of uncertainty gives a region of uncertainty which indicates the spatial distribution of prediction errors. Figure 16(a) shows a cross-section of this error volume in the plane defined by the lens-centers and the point of interest. The case shown is for equal sized error cones with the point chosen equidistant from the lens-centers. The depth uncertainty $\Delta d$ is an inverse function of the ratio $b/d$ of the baseline $b$ to the depth $d$ of the point. The 'depth' in this case is defined as the normal distance to the baseline. Configurations having large (small) $b/d$ are designated
Figure 16 Estimates of Depth Uncertainty

(a) Stereo Uncertainty Relations

(b) Approximation for Off Axis Case
'wide (narrow)' stereo. An asymmetric configuration can be approximated by a symmetric one as shown in figure 16 (b).

In this experiment, we avoided the correlation problem by processing the second image to extract a bounding polygon of a calibration object. We used prior knowledge of the object to order the vertices for the proper correspondence. In the operational stereo system currently being implemented a 'corner-detector' designed by A. Gill (9) is used as the feature detector. It searches a small prespecified rectangle for a pair of intersecting edges. The intersection, if found, is recorded and called a corner. The detector is guided along the straight line calculated by the strategy of part II.
V. Centering

We can now consider the centering problem of part I in more detail. A system is currently being implemented at Stanford in which a wide-angle lens will be used for low-resolution large-area searches in conjunction with a facility for examining a subfield of view at higher magnification. The sequence of operation of this generalized viewing system is (see figure 17)

1. Find an area of interest in the wide-angle view.
2. Pan and/or tilt the camera so as to center the area about the principal-point of the image.
3. Zoom or change lenses to achieve the desired magnification.

The specification of an area of interest will probably come from context since this sequence will only be implemented when more resolution is needed in a certain area; for example, figure 17 depicts a situation where the error in alignment of two stacked blocks must be measured with more precision than is available in the wide-angle view. In this case, the amount of precision needed will be just that to resolve the largest tolerable error. In its most general form the specification will be implemented as a function of the form:

\[ \text{LOOK AT (OBJ,LOC,MAG)} \]

where

OBJ + description such as "the interior vertex of the big red block"

LOC + location designation such as "in the upper right corner"

MAG + a number specifying the magnification desired
Figure 17 Close-up Segmentation

(a) Magnification
(b) Centering
(c) Acquisition
The output of such a function will have to be \([h,v,f]\) where \([h,v]\) are the image-plane coordinates of the feature to be centered and \(f\) is the principal-distance needed for the desired magnification. 'f' implicitly specifies a lens focal length.

Prediction will be used as much as possible in centering. Optical-servoing - i.e., feature tracking and correcting aligment - will be used only to remove residual errors.

Prediction involves solving a pair of simultaneous equations which express the relation between image coordinates of a point and the settings of the PAN and TILT angles needed to center it in the field of view. If range is not known for the feature of interest, there will be one degree of freedom left in the solution. Although optical-servoing is generally applicable to problems of feature aligment, it is costly in terms of computation time. The image processing involved in feature tracking takes much longer than the solution of the prediction equations.

Let us now look at this situation in greater detail. In the discussion in part I, we assumed that we knew the location in reference coordinates of the point we wanted to center on. The more common situation is the one where the robot's strategy program decides to have a closer look at some as yet unidentified area of its field of view. We start by assuming the simpler first case and outline the solution for the desired PAN and TILT angles. We then show how to generalize this to the second case.
Let P and P' be respectively the reference and camera coordinates of the point of interest. Figure 18 shows a possible centering sequence of first panning and then tilting. The condition for centering can most easily be expressed by saying that the x and y components of P' must be zero. That is
\[ P_x' = P_y' = 0. \]

If we write the rotation [R] in terms of row-vectors R1, R2, R3 then we can expand the above condition to two equations by recalling that
\[ P' = [R] \cdot (P - C) \]

They are
\[ R_1 \cdot (P - C) = 0 \]
\[ R_2 \cdot (P - C) = 0 \]

Substituting the expression given previously, in part II(c)
\[ C = P_0 + [R]^t \cdot DP \]

for C in terms of [R] gives
\[ R_1 \cdot (P - P_0) = DP_x \]
\[ R_2 \cdot (P - P_0) = DP_y \]

where \( DP_x \) and \( DP_y \) are the x and y components of DP. The unknowns in this system are the vectors R1 and R2. [R] is determined by PAN, TILT, and SWING. SWING is known and fixed. Thus, we are left with two transcendental equations in two unknowns. They are messy and attempts to solve them analytically have, so far, not been successful. They have been solved quite rapidly using various iterative techniques. In addition, an approximate set has been formulated by A. Gill (9) at Stanford and solved analytically. In most cases, the approximate solution gives satisfactory accuracy.
After the pan-tilt head is moved as predicted, a feature correlator can be run to actually locate the point of interest and measure a prediction error. If the reference coordinates are accurately known for the point, this error can be used as an indication of the inaccuracy of the camera calibration. In addition, the reference and image coordinates for the point can be stored as data for later calibrations.

If $P$ is not completely known but has been identified as interesting in the original view, we can generate the equation for the ray through it. As was shown in part II, we use $[C_0L]^{-1}$ and $C$ to get the parametric ray equation $P(t)$ in terms of the unknown range parameter $t$. We can then substitute the expression for $P(t)$ into the above system and get $PAN(t)$ and $TILT(t)$. In practice, some range bound will already be available as in the stereo case discussed earlier. An initial range estimate $t_e$ can be the expected range of objects in the work space, or a focus ranging estimate (31). By using $t_e$ to compute an initial $PAN(t_e)$ and $TILT(t_e)$ we can limit prediction errors to tolerable values. After panning and tilting to the values specified by $t_e$, we can do a search of the straight line image of $P(t)$ to find the feature in the new view. The prediction routine can be called again with the new $[h,v]$ coordinates and the process repeated until the error is within acceptable bounds. This process will result in some final value $t_f$ for $t$ and $P(t_f)$ and can be used to refine information about the reference coordinates of the point.

Note here that we have just described a narrow-stereo system. Because the viewpoint displacement is small, range accuracy is limited. This is one instance of a general proposition in this field - everything is related to everything else.
In conclusion, we have presented a camera model, given some arguments for its general utility in a robot visual system, and shown specifically how it is used in stereo vision and centering tasks.
References


