MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. No. 143

September 1967

Stereo and Perspective Calculations

Marvin Minsky

A brief introduction to use of projective coordinates for hand-eye position computations. Some standard theorems. Appendix A reproduces parts of Roberts' thesis concerning homogeneous coordinates and matching of perspectively transformed objects. Appendix B by Arnold Griffith derives the stereo calibration formulae using just the invariance of cross-ratios on projections of lines, and he describes a program that uses this.

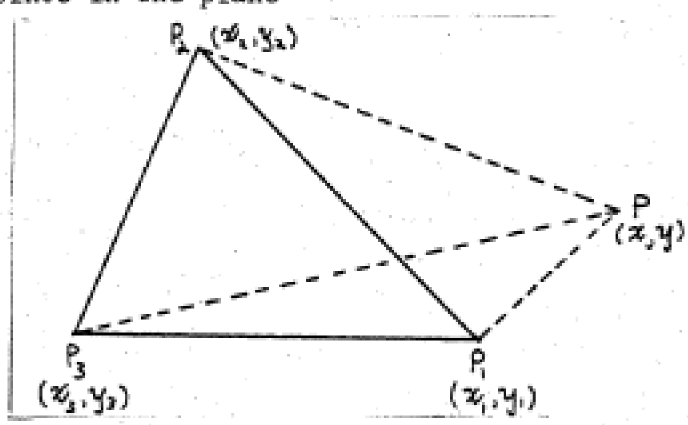
Perspective and stereo calculations

1.0 Redundant Coordinates

We are used to using linearly independent, e.g., Cartesian coordinates for analytic geometry. As you will see, there are sometimes advantages in using "redundant" coordinate systems, with an "extra" axis, because computations become simpler. Also, they can help with "numerical stability," in avoiding divisions by keeping numerators and denominators separate until the end of the computation.

1.1 Barycentric Coordinates

A well-known coordinate system is the barycentric system. Choose 3 non-collinear points in the plane



Then the barycentric coordinates of an arbitrary fourth point P is the ordered set of weights of mass-points at the P_i's that would have their center-of-gravity at P. Of course, this is not unique, because if (m_1, m_2, m_3) represents P, then so does (km_1, km_2, km_3). We could set $m_1 + m_2 + m_3 = 1$, or use the ratios m_1/m_3, m_2/m_3 as unique representors.

Given $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $P_3 = (x_3, y_3)$ in Cartesian coordinates, we find the (normalized) barycentric coordinates of an unknown $P = (x, y)$ by solving

$$m_1x_1 + m_2x_2 + m_3x_3 = x$$

$$m_1y_1 + m_2y_2 + m_3y_3 = y$$

$$m_1 \cdot 1 + m_2 \cdot 1 + m_3 \cdot 1 = 1$$

or

$$(m_1, m_2, m_3) \cdot \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

so if D is the determinant then

$$(Dm_1, Dm_2, Dm_3) =$$

$$\left(\begin{vmatrix} x & x_2 & x_3 \\ y & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}, \begin{vmatrix} x_1 & x & x_3 \\ y_1 & y & y_3 \\ 1 & 1 & 1 \end{vmatrix}, \begin{vmatrix} x_1 & x_2 & x \\ y_1 & y_2 & y \\ 1 & 1 & 1 \end{vmatrix} \right)$$

There is one coordinate system in which this comes out neatly:

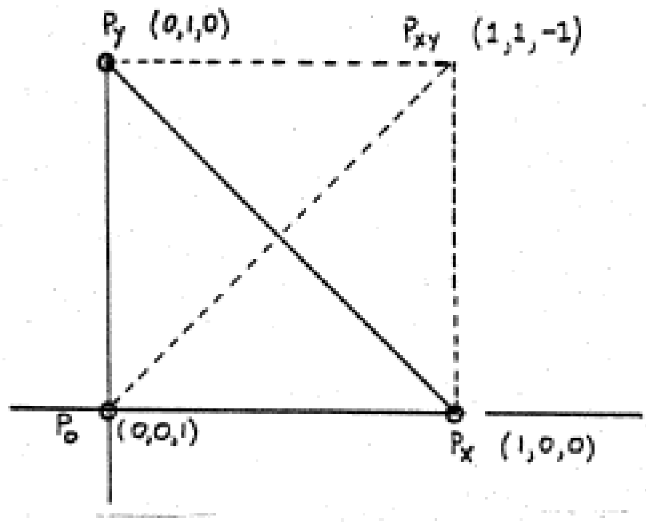


Figure 1.1

There the barycentric coordinates of a point (y, y) are (x, y, 1 - x - y) so that the first two components <u>are</u> the Cartesian coordinates, if the multiplier is chosen to make the three sum to unity.

1.2 Projective Coordinates

The trouble with using barycentric coordinates for Vision is that they aren't suitably invariant under perspective changes. But this can be fixed! Choose a fourth point P_{xy} and let (n_1, n_2, n_3) be its barycentric coordinates for P_x, P_y and P_0*. Now define the <u>projective</u> <u>coordinates</u> of a point P with respect to P_x, P_y, P_0 <u>and</u> P_{xy} to be

$$(p, q, r) = \left(\frac{m_1}{n_1}, \frac{m_2}{n_2}, \frac{m_3}{n_3}\right)$$

* Now P_x, P_y, and P_0 are <u>any</u> non-collinear triple.

where (m_1, m_2, m_3) are the barycentric coordinates of P with respect to P_x, P_y, P_0. This is invariant, under a projection of any quadrilaterial P_x, P_y, P_0, P_{xy} into another $P'_x, P'_y, P'_0, P'_{xy}$! Note that in projective coordinates the points

$$P_x \qquad P_y \qquad P_0 \qquad P_{xy}$$

always have components

$$(1,0,0) \qquad (0,1,0) \qquad (0,0,-1) \qquad (1,1,1).$$

For our standard system we will use a unit square.

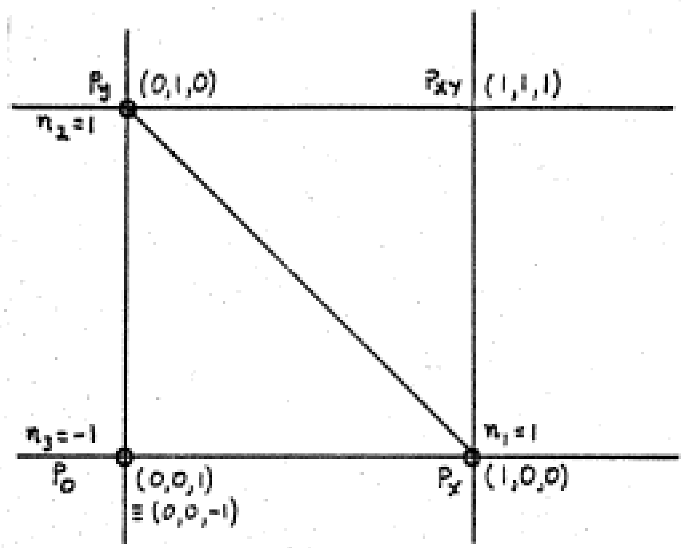
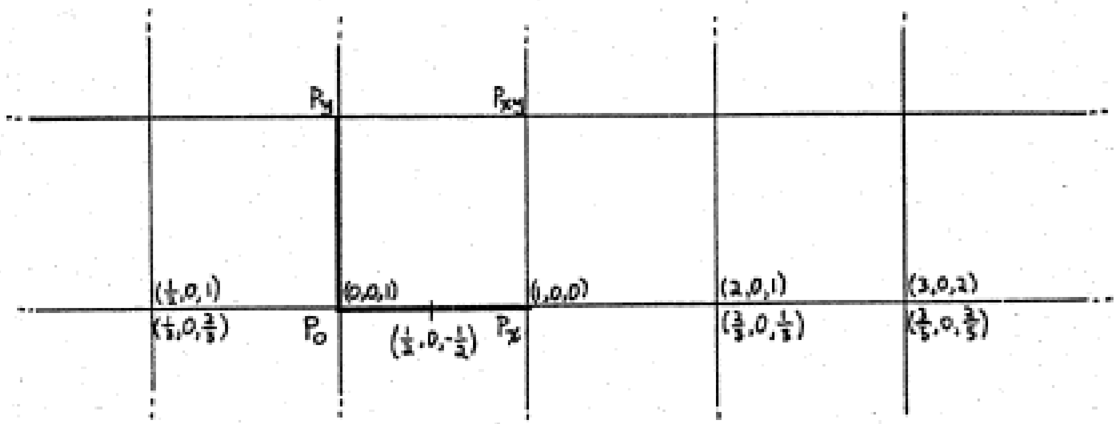


Figure 1.2

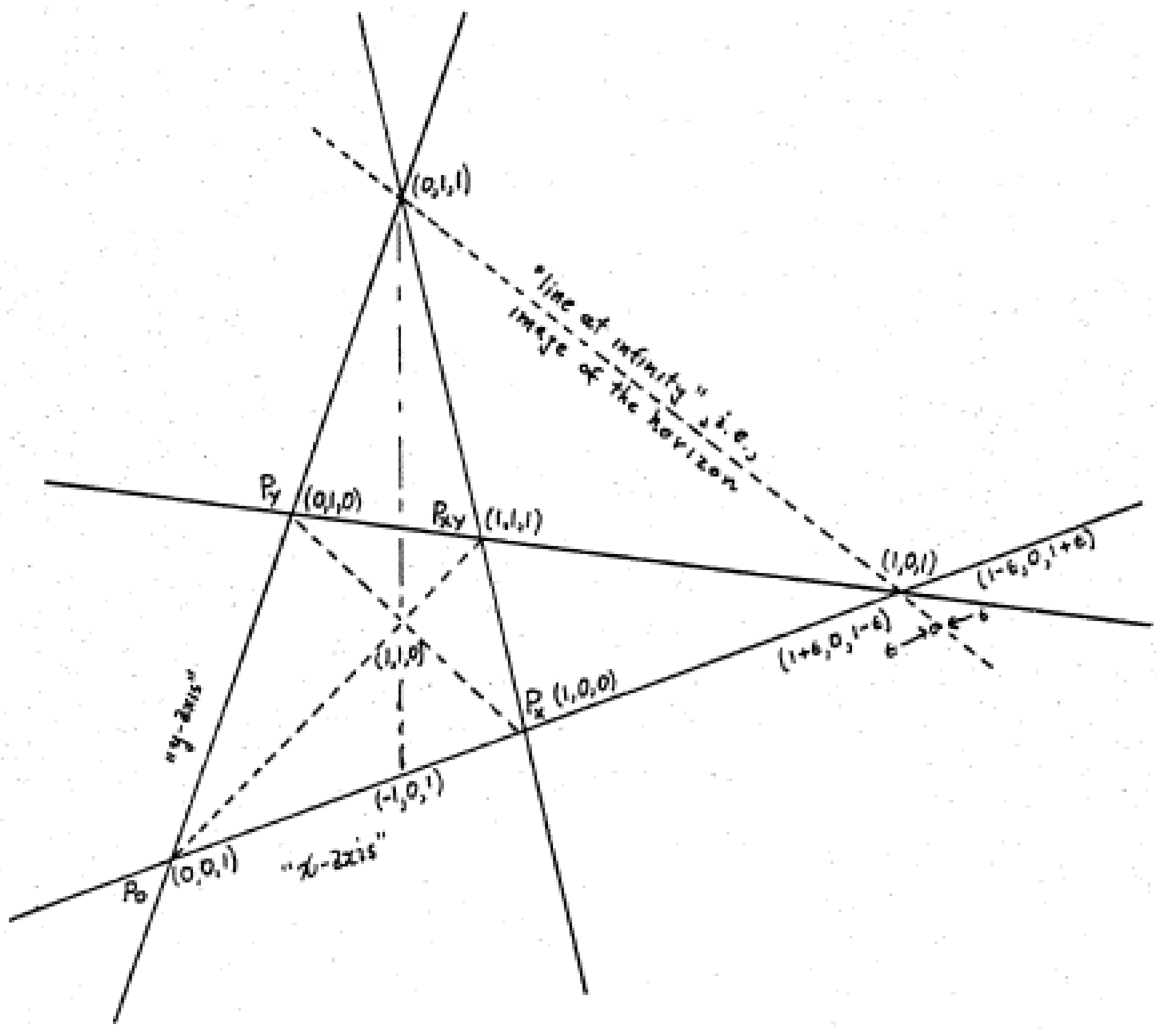
In this system the projective coordinates of a Cartesian point P = (x, y) are proportional to

$$(x, y, x + y - 1).$$

Here are some coordinates of points, some "normalized to make their sum = unity.



With a skew quadrilaterial one see more clearly what is happening:



1.3 Transforming Back

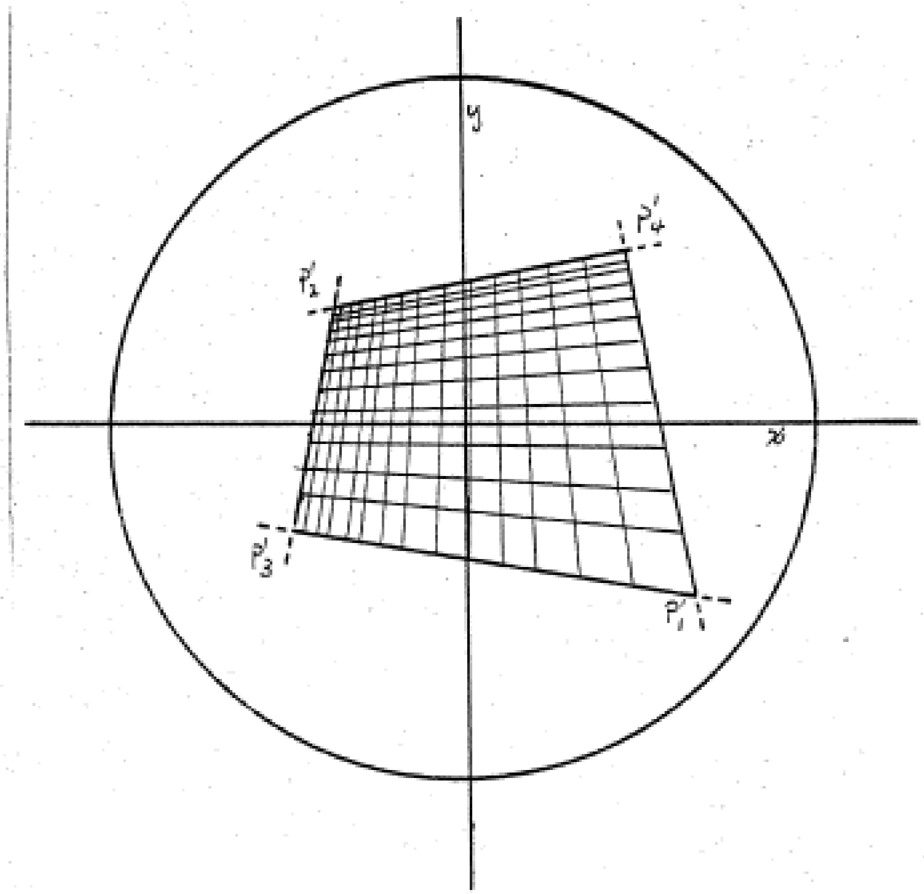
To transform back from projective to Cartesian coordinates, one assumes that one has a projective point $(p, q, r) = (kx, ky, k(x + y) - 1)$ hence

$$x = \frac{p}{p+q-r} \qquad \text{and} \qquad y = \frac{q}{p+q-r}$$

and this will work except when $(p+q-r)$ is dangerously small, numerically. This happens for very large x's and/or y's and means that the point cannot be in the "table-plane" or it would be too far away! The "bad points" lie near the "line at infinity" which has the equation $n_1 p + n_2 q + n_3 r = 0$, and we will discuss it later.

2.0 Transforming from Eye coordinates to Table coordinates: no z-axis

If we think of the table as a big square, whose corners are the vertices of Figure 1.2, its projection on the retina will look something like this:



and we want to find the point in table-coordinates corresponding to an arbitrary point seen on the retina. (We assume the point is projected from the table--we'll worry about the z-axis later.)

We need some notation. Let

> (u_i, v_i) be <u>retinal</u> (i.e., "vidissector") <u>coordinates</u>;
> (x_i, y_i) be <u>table</u> Cartesian <u>coordinates</u>;
> (p, q, r) be <u>projective</u> coordinates.

We don't have to say, for projective coordinates, whether they are <u>table</u> or <u>retinal</u>, because they're invariant! Now the four P-points (the table corners) have coordinates

	Projective	Table	Retinal
P_x	(1,0,0)	(1,0)	(u_1, v_1)
P_y	(0,1,0)	(0,1)	(u_2, v_2)
P_0	(0,0,1)	(0,0)	(u_3, v_3)
P_{xy}	(1,1,1)	(1,1)	(u_4, v_4).

Suppose that their retinal coordinates have in fact been determined--by experiment!--using, say, the program de veloped by Arnold Griffith, or by the Vision System.

Now we can take an arbitrary point (u, v) on the retina and transform it to projective coordinates, as follows.

First--once and for all--we find the barycentric coordinates of P_{xy} with respect to P_x, P_y, P_0 by using the formulae (see §1.1):

$$n_1 = \begin{vmatrix} u_4 & u_2 & u_3 \\ v_4 & v_2 & v_3 \\ 1 & 1 & 1 \end{vmatrix} \qquad n_2 = \begin{vmatrix} u_1 & u_4 & u_3 \\ v_1 & v_4 & v_3 \\ 1 & 1 & 1 \end{vmatrix} \qquad n_3 = \begin{vmatrix} u_1 & u_2 & u_4 \\ v_1 & v_2 & v_4 \\ 1 & 1 & 1 \end{vmatrix}$$

and these are saved for future use. Now the projective coordinates (p, q, r) for the point (u, v) are given by (see §1.2):

$$p = \frac{1}{n_1} \begin{vmatrix} u & u_2 & u_3 \\ v & v_2 & v_3 \\ 1 & 1 & 1 \end{vmatrix} \qquad q = \frac{1}{n_2} \begin{vmatrix} u_1 & u & u_3 \\ v_1 & v & v_3 \\ 1 & 1 & 1 \end{vmatrix} \qquad r = \frac{1}{n_3} \begin{vmatrix} u_1 & u_2 & u \\ v_1 & v_2 & v \\ 1 & 1 & 1 \end{vmatrix}$$

Observe, for programming speed, that these can be expanded as

$$p = u \left[\frac{v_2 - v_3}{n_1} \right] + v \left[\frac{u_3 - u_2}{n_1} \right] + \left[\frac{u_2 v_3 - v_2 u_3}{n_1} \right]$$

$q =$ etc.

$r =$ etc.

where the bracketed quantities can be precomputed once and for all!!

Each retinal → projective conversion requires only two multiplications and two additions per component. One doesn't have to worry about numerical stability at this point because the n's will be large enough. (That is why we should use the whole table: so that P_4 can be maximally far from the other P's and thus have large barycentric coordinates.)

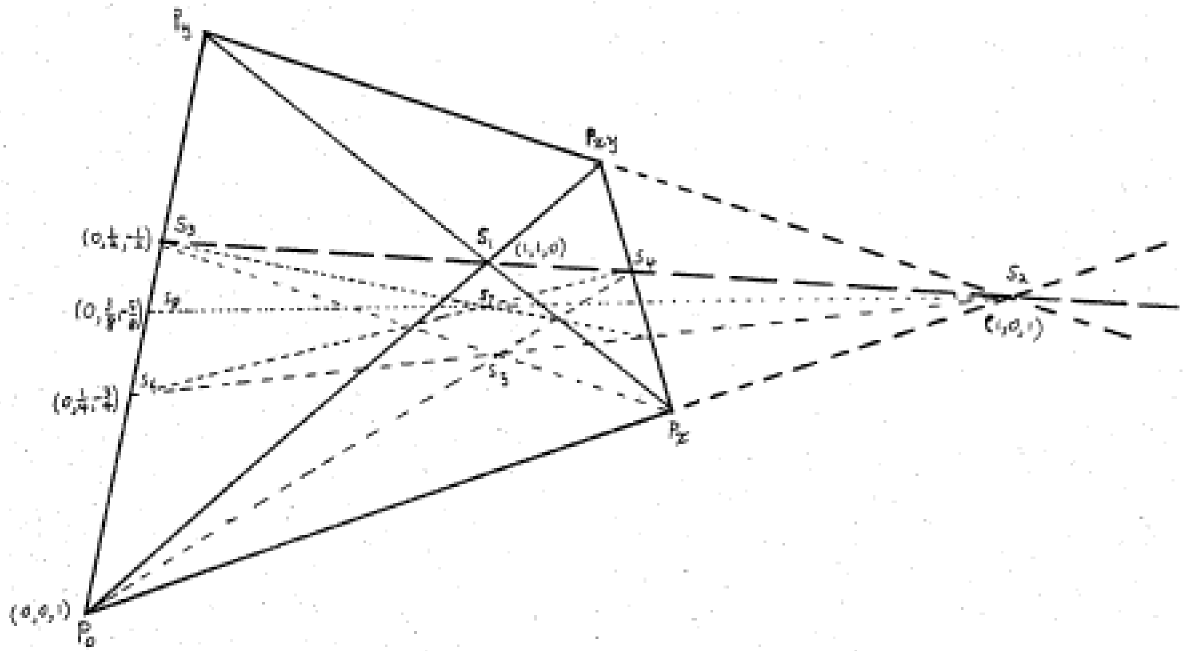
Finally one has to Transform Back to Cartesian table coordinates (see §1.3). This need not be done immediately. We should watch this carefully, in fact, because often it need never be done!

3.0 Some Theorems

3.1 Quadrangles

You might wonder why a plane quadrangle is sufficient, or why a triangle is insufficient. The triangle isn't enough because there are 9 parameters in the transformation, with one of them redundant in the ratios, leaving 8. Various degree-of-freedom analyses will show this. You might prefer the following constrictive argument to the usual kind of axiomatic proofs found in projective-geometry books. (By the way, I am a novice about the theory; this is just my initial way of understanding it, and it may be "immature.") We will accept that projection of a picture through a point (the lens center approximation) onto another plane carries lines into lines, because pairs of planes intersect in lines.

Now, given an arbitrary quadrilateral, observe that one can geometrically construct the "projective" midpoint of any side



by first finding its center S_1 by intersecting diagonals, then finding the vertex S_2 by intersecting opposite sides, and finally finding the "midpoint" of P_0, P_y by intersecting with the line through S_1 and S_2. It is clear that this is the invariant point with projective coordinates (0,-1,1). Now observe that we have a new, known, quadrilateral P_0, P_x, S_4, S_3 so we can do it again, invariantly bisecting S_3, P_0 or, similarly, S_3, P_y. Repeating this indefinitely often, we thus can construct all binary fractions along the line P_0, P_y!

Thus we obtain, by "closure," or continuity, or Dedekind Cuts...whatever you prefer, a real coordinate system along $[P_0, P_y]$ like the picture in §2 where we see the projection of a Cartesian coordinate system.

We have thus shown, by arguments using only intersections of lines, that this image is uniquely constructable from the vertices of our arbitrary quadrilaterial--hence it must be "projectively invariant."

3.2 Useful theorems about homogeneous coordinates

In addition to the Cartesian coordinates (x, y) we are interested in homogeneous coordinates, defined as the equivalence-classes of triples under multiplication of each component by the same factor k. The types we will treat are

H: (Homoscalar) $(x, y, 1)$

B: (Barycentric) $(x, y, 1-x-y)$

P: (Projective) $(x, y, x+y-1)$.

Scale Change

In all systems if

$$(x, y) \equiv (u, v, w)$$

then

$$(kx, ky) \equiv (u, v, \frac{w}{k}).$$

Transforming Back to Cartesian

H: $(u, v, w) \to \frac{1}{w}(u,v)$

B: $(u, v, w) \to \frac{1}{u+v+w}(u, v)$

P: $(u, v, w) \to \frac{1}{u+v-w}(u, v)$.

Lines

We will represent a line as a column vector $[u, v, w]$. The Cartesian line $ax + by + c = 0$ becomes

H: (a, b, c)

B: $(a+c, b+c, c)$

P: $(a+c, b+c, -c)$.

(The simplicity of the form for H recommends it highly.)

The point (u, v, w) is on the line [p, q, r] if $(u, v, w)\cdot[p, q, r] = 0$, that is, if $up + vq + wr = 0$. The sign of the scalar product tells on which side of the line is the point, and even its distance away, provided the vectors are normalized to standard form. For example, in B:

$$(u,\ v,\ 1-u-v)[p+r,\ q+r,\ r] = up + vq + r.$$

The line through two points (u_1, v_1, w_1) and (u_2, v_2, w_2) is given by

$$\begin{vmatrix} u & u_1 & u_2 \\ v & v_1 & v_2 \\ w & w_1 & w_2 \end{vmatrix} = 0$$

that is,

$$u(v_1w_2 - v_2w_1) + v(w_1u_2 - w_2u_1) + w(u_1v_2 - u_2v_1) = 0$$

because the determinant will vanish only when the first column equals the second, or the third, or a linear combination of them.

Two lines $[p_1, q_1, r_1]$ and $[p_2, q_2, r_2]$

intersect in the point

$$(q_1r_2 - q_2r_1,\ r_1p_1 - r_2p_1,\ p_1q_2 - p_2q_1).$$

Useful fact: the three points

$$(0,1,1) \quad (1,0,1) \quad (1,1,0)$$

are never collinear. (This is known as Fano's Axiom.)

<u>Transformations</u>

To translate $P = (x,y,w)$ to the origin, i.e., to transform $(x + \alpha,\ y + \beta) \to (\alpha, \beta)$ apply the matrix: $P' = PT$.

$$T(P) = \begin{pmatrix} w & & \\ & w & \\ -x & -y & w \end{pmatrix}$$

This works in any system, provided w is normally defined.

The line through a point P normal to the line L is given by

$$T(P) \cdot K \cdot L$$

where T(P) is as above and K is

$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The line parallel to L through P is given by $T(P)K^2L$ where

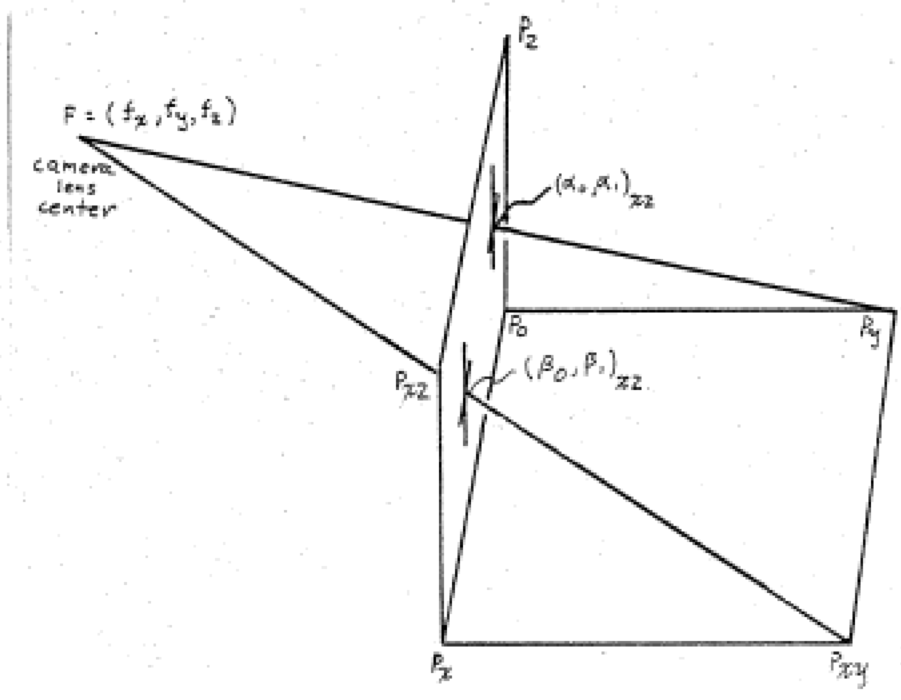
$$K^2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

4.0 Finding where the eye is

There are any number of ways to do this and the choice depends on (1) convenience of a "calibration object" and (2) avoidance of numerical degeneracy--loss of precision. The larger the calibration object, the better --Gosper's use of a large imaginary object whose vertices are traced out by the arm allows larger spans than would be convenient for real objects in the laboratory, as well as providing the important direct hand-eye-relation coefficients. Also, this dynamic calibration-object could be programmed to provide favorable positions as a function of the observations, to avoid degeneracy for a wider class of hand-eye position relations.

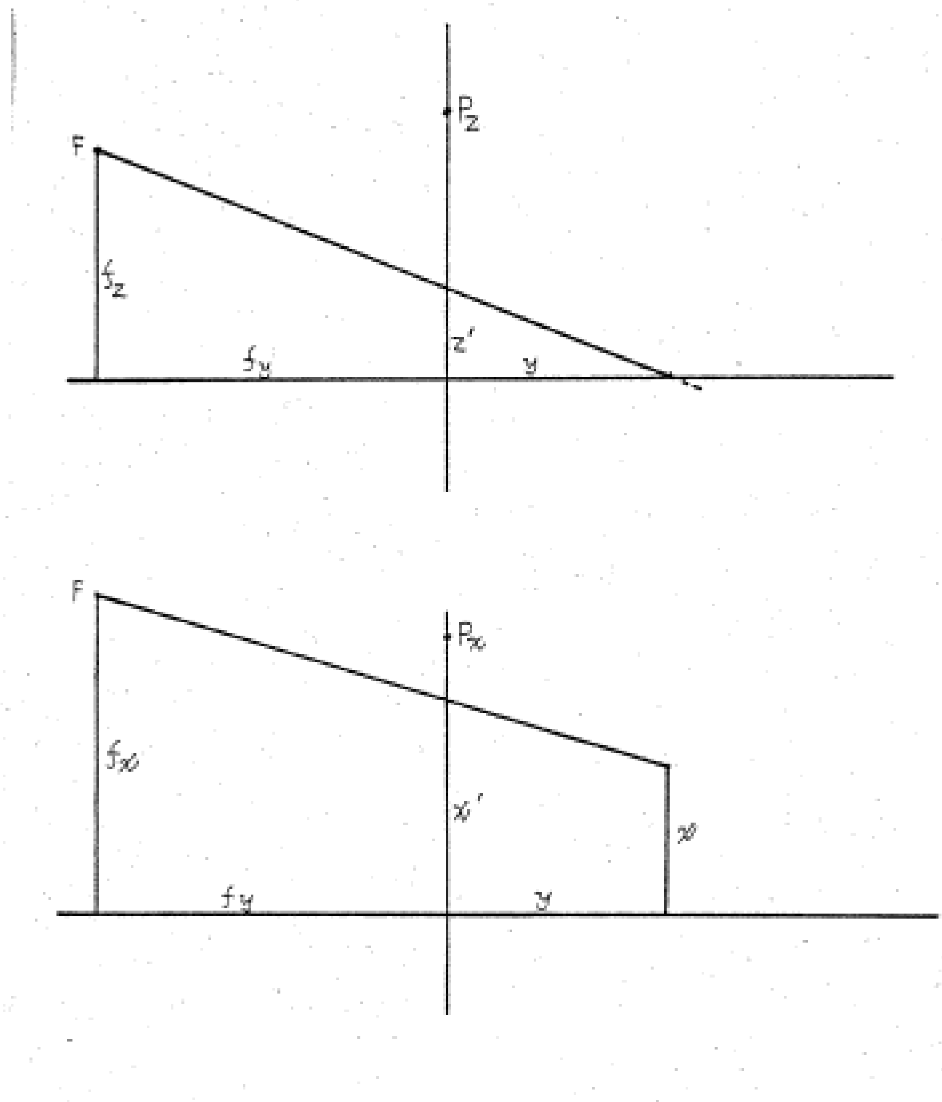
4.1 A Reference Cube

We will analyze a particularly straightforward test-object--six of the vertices of a cube. These form two squaresat right angles.



We will find the point F of the lens center by observing the projections (α_0, α_1) and (β_0, β_1), of the points P_y and P_{xy}, in the xz plane.

The analysis is simple when we use the projective coordinates in both planes. First, observe the geometry of the projections along the x and z axes, for a point (x, y) in the xy plane: we want to find its projection (x', z') in the xz plane.



By similar triangles we have

$$z' = \frac{yf_z}{y + f_y} \qquad x' = \frac{yf_x + xf_y}{y + f_y}\ .$$

For the two points $P_y = (0,1)_{xy} = (\alpha_0, \alpha_1)_{xz}$ and $P_{xy} = (1,1)_{xy} = (\beta_0, \beta_1)_{xz}$ we then obtain

$$\alpha_0 = \frac{f_z}{y + f_y} \qquad \beta_0 = \frac{f_z}{y + f_y}$$

$$\alpha_1 = \frac{f_x}{y + f_y} \qquad \beta_1 = \frac{f_x + f_y}{y + f_y}$$

Now α_0, α_1, β_0, β_1 are computed as the normalized projective coordinates of the images of P_y and P_{xy} in the quadrangle P_0, P_x, P_z, P_x so we can assume that they are available. But then we obtain, simply:

$$f_x = L\alpha_0$$

$$f_y = (\alpha_1 - \alpha_0)/L$$

$$f_z = L\beta_0$$

$$L = \frac{1}{1 - (\alpha_1 - \alpha_0)}$$

Two points should be noticed:

(1) α_0 and β_0 should be equal. Any deviation is a measurement error: if large, the observation is suspect. If small, one probably should use $\alpha_0' = \frac{\alpha_0 + \beta_0}{2}$.

(2) The quantity $L = \frac{1}{1 - (\alpha_1 - \alpha_0)}$ is critical for accuracy. If the eye is relatively far from the scene, the projection will be nearly parallel; the $\alpha_1 - \alpha_0 \cong 1$ and L will be small, killing precision--as one would expect. For a large 2-foot object with the Eye 6 feet away, we would have L = 1/3 so that the precision of the F measurements is 1/3 that of the plane measurements, which is fine.

4.2 Using a Matrix

The geometric argument above is perfectly valid because the first two components of the projective coordinates are equal to the appropriate Cartesian coordinates. Let's just compute the xy → xz projective transformation for fun. Given $(x, y, x + y - 1)_{xy}$ we want to get $(x', z', x' + z' - 1)_{xz}$. Since,

$$x' = \frac{yf_x + xf_y}{y + f_y} \qquad \text{and} \qquad z' = \frac{yf_z}{y + f_y}$$

we have

$$x' + z' - 1 = \frac{yf_z + yf_x + xf_y + f_y - y}{y + f_y}$$

and the required triplet is obtained by

$$\frac{1}{y+f_y}\begin{pmatrix} f_y & f_x & 0 \\ 0 & f_z & 0 \\ 2f_y & f_x+f_y+f_z-1 & -f_y \end{pmatrix}\begin{pmatrix} x \\ y \\ x+y-1 \end{pmatrix} .$$

Substituting $P_y = (0,1,0)$ and $P_{xy} = (1,1,1)$ we have

$$(\alpha_0, \beta_0, \sim) \equiv \frac{1}{1+f_y}(f_x, f_z, \sim)$$

$$(\alpha_1, \beta_1, \sim) \equiv \frac{1}{1+f_y}(f_x+f_y, f_z, \sim)$$

which leads, of course, to the same solution as in §4.1. There is no particular reason, here, to use the matrix formulation; there would, if one used more complicated coordinates, e.g., tetrahedra. The term $f_x + f_y + f_z - 1$ suggests that one might find value in using a spatial projective system in which the lens-point (f_x, f_y, f_z) is a reference vertex, and Lennon is said to be writing a memo about this. The barycenters of F in the $P_0\ P_x\ P_y\ P_z$ tetrahedra are $(f_x, f_y, f_z, 1 - f_x - f_y - f_z)$.

To summarize the procedure to find where the eye is:

(1) Measure the six points' positions.

(2) Find the barycentric coordinates of P_{xz} with respect to P_0, P_x P_z. §1.1.

(3) Express P_y, P_{xy} in the xz projective system. §1.2.

(4) Use the formulae in §4.1 to find f_x f_y and f_z.

5.0 Finding a space point, using Stereo

Suppose that two points (x, y) and (x', y') are found in the two eyes, and are supposed to be images of the same point (x^*, y^*, z^*). Then they must lie on the intersection of the two lines defined by (see diagram on p. 20)

$$(x, y, 0) \leftrightarrow (f_x, f_y, f_z)$$

and

$$(x', y', 0) \leftrightarrow (f_x, f_y, f_z).$$

If we parametrize the lines by $0 \le t \le 1$ and eliminate z we obtain from the x equation

$$t = \frac{1}{1 - \dfrac{f_x f'_z - f'_x f_z}{x f'_z - x' f_z}}$$

which is the fractional distance of the intersection from the table to the first eye. Then the intersection is

$$((1-t)x + tf_x, (1-t)y + tf_y, tf_z) = (x^*, y^*, z^*).$$

Of course, one might encounter an error because the two lines don't intersect! To check this, one might also compute the other version of t:

$$t = \frac{1}{1 - \frac{f_y f'_z - f'_y f_z}{y f'_z - y' f_z}}$$

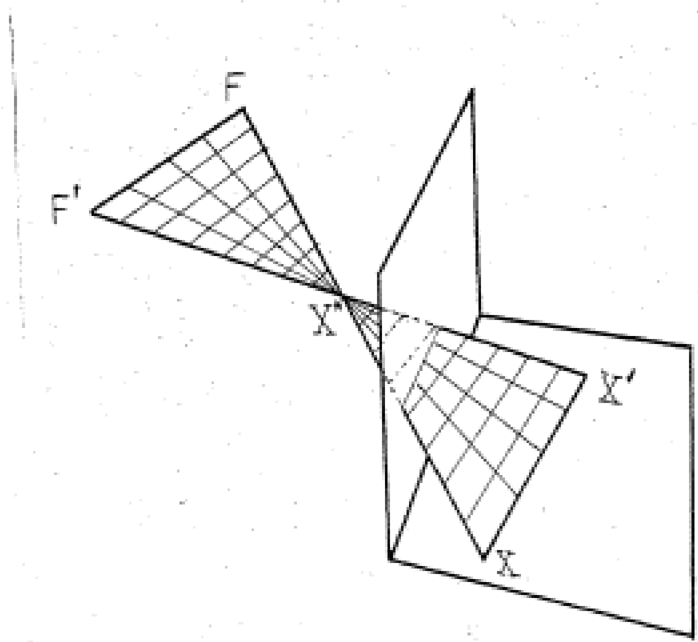
and if it doesn't agree closely something is wrong. I hope someone else will do a more thoughtful error analysis!

Dimensional Analysis and Stability

It would be valuable for someone to analyze the general precision question about all of this. Consider the following questions:

(1) Do I need six points? Clearly not, in principle. But what about practice? Can we do just as well with 5 or even 4 points?

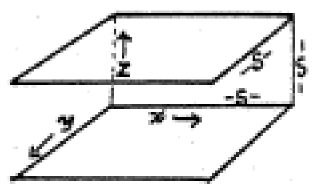
(2) To look at it another way, if we have redundant information, can we use it better? We used just two rays in our analysis, but we could also use the projections of P_z in the xy plane, P_0 in the P_z, P_{zx}, P_y, P_{xy}, etc. So one could use all six available rays and average, or something, to reduce errors. ("Something" might just be selection of the best pair.)



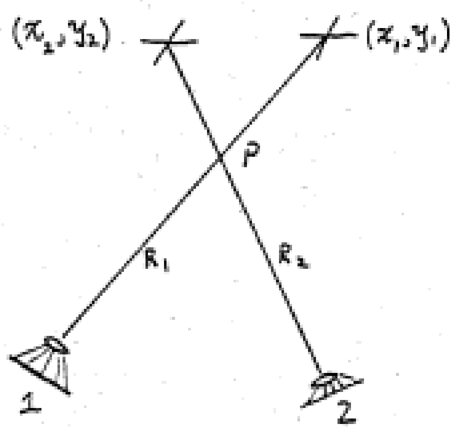
Appendix B. by Arnold Griffith

General Approach

A calibration object consisting of two parallel squares of edge length and separation s, defines a coordinate system (x, y, z) in the real world:



A point P in space gives rise to two rays R_1 and R_2; both incident with P, and respectively incident with the "points of view" of the cameras in the positions designated one and two:



Let (x_i, y_i) and (x_i', y_i') be the intersections of R_i with the XY plane and the XY' planes respectively*. The basis of the stereo routine here described

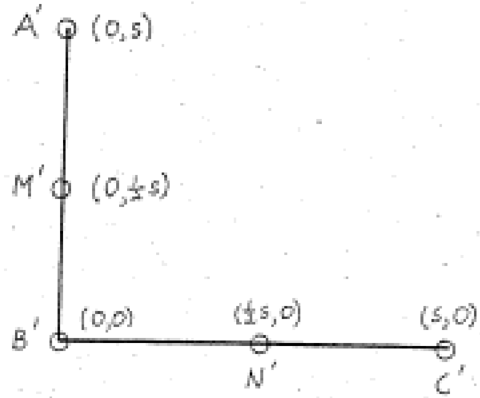
---

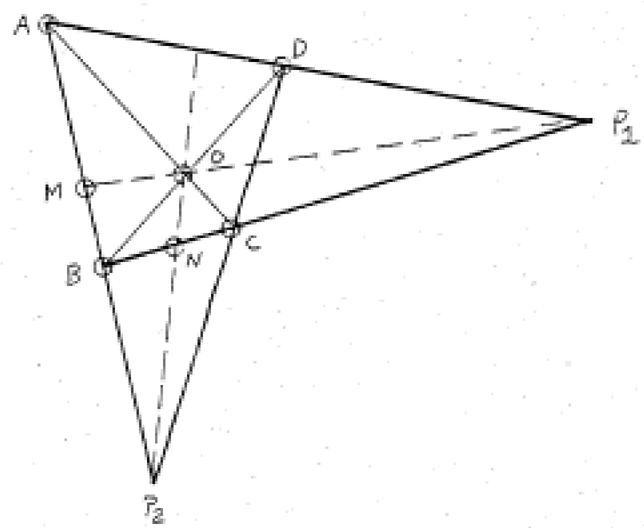
* The XY' plane may be seen from the diagram to be parallel to and with similar orientation, as the XY plane, but displaced a distance s.

is the determination of these points. Once these are determined, the location of P in space follows easily.

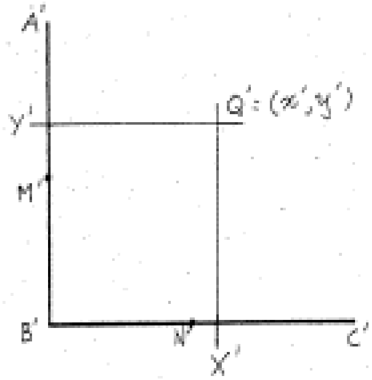
The Determination of an Intersection Point, an Example

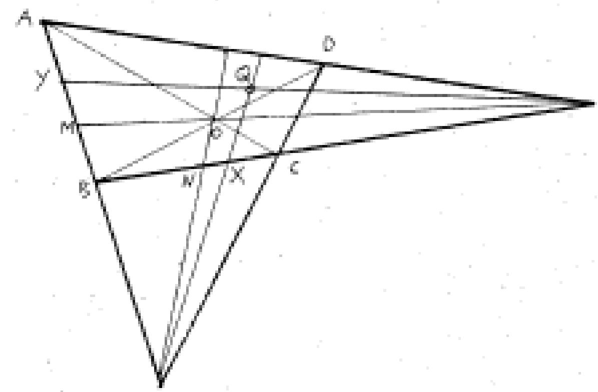
The image on the retina, i.e., the focal plane, of the square which defines the XY plane is a quadrilaterial ABCD.





Note that O must be the image of the center of the square A'B'C'D', and hence M and N are the images of the edge midpoints M' and N'. Let Q be the image of some point in space. Since Q is the image of any point along R_1, it is the image of (x_1, y_1). It is easy to see that Y and X on the diagram are the images of the points $(0,y_1)$ and $(x_1,0)$:





Consider the computation of, e.g., y_1. By a theorem of projective geometry we have that

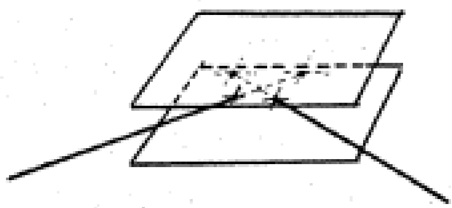
$$\frac{A'M' \cdot Y'B'}{Y'A' \cdot M'B'} = \frac{AM \cdot YB}{YA \cdot MB} \qquad (1)$$

where, e.g., AM stands for an (unsigned) distance between A and M. It is easy to see that this equation is still valid when AM is interpreted as a vector from A to M, and $\cdot$ is dot-product. Hence Eq. (1) becomes, eventually:

$$z_1 = s \cdot \frac{MA \cdot BY'}{MB \cdot AY + MA \cdot BY} . \qquad (2)$$

<u>The Determination of Position</u>

The rays R_1 and R_2 may not intersect because they came from two points which are not the same, or because of measurement errors. It is important to have a criterion for closeness such that rays satisfying the criterion are to be considered to be from the same object. An easily computed criterion is the difference in z-values of two points, one on each line, which have the same x and y coordinates. The computation involves an intersection of two lines in 2-space.



If the lines are sufficiently close, i.e., the z-values at (x, y) on the two lines differ by a sufficiently small amount, then the z value for the point is the average of the two z values on the lines.

<u>Availability</u>: The following programs exist and will be described in further detail in a forthcoming Vision Memo.

(CAL) Does the calibration; i.e., looks at the real world and finds A, B, C, D, O, P_1, P_2, M and N for both eyes.

(WHERE1 P C) P is a point, C is one or two, depending on which camera position P is seen at. Returns $((x_C, y_C)(x_C, y'_C))$.

(WHERE2 X Y) X is the value of (WHERE1 P 1), Y is (WHERE1 P2). Value is ((x, y, z) d), where (x, y, z) is the position of the point, P, and d is the z-value disparity described above.