VISMEM: A bag of "robotics" formulae

Working Paper 34

by

B.K.P. Horn

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

December 1972

Abstract:

Here collected you will find a number of methods for solving certain kinds of "algebraic" problems found in vision and manipulation programs for our AMF arm and our TVC eye. They are collected here to avoid the need to regenerate them when needed and because I wanted to get rid of a large number of loose sheets of paper in my desk. Documented are various methods hidden in a number of old robotics and vision programs. Some are due to Tom Binford and Bill Gosper.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

Working Papers are informal papers intended for internal use.

TABLE OF CONTENTS:

LINES, LINE-SEGMENTS, REPRESENTATIONS AND CALCULATIONS:

There are many ways of representing a line by an appropriate equation. A minimum of two parameters is required. Unfortunately these parameters tend to become singular near certain angles. Special purpose tests must be made to handle these cases. The more redundant 3 and 4 parameter representations not only solve this problem but simplify the operations required to manipulate lines and points.

Let θ be the inclination of the line to the x-axis and ρ its perpendicular distance from the origin.

Let (x_0, y_0), (x_1, y_1) and (x_2, y_2) be points on the line. Then some equations for a line are:

$$y = mx + c \;\rightarrow\; \begin{cases} y - (\tan\theta)x - \dfrac{\rho}{\cos\theta} = 0 \\[2ex] x - (\cot\theta)y + \dfrac{\rho}{\sin\theta} = 0 \end{cases}$$

$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} \longrightarrow (x-x_1)(y_2-y_1) - (y-y_1)(x_2-x_1) = 0$$

Let $\ell = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$ $\quad \cos\theta = \dfrac{x_2-x_1}{\ell}$ $\quad \sin\theta = \dfrac{y_2-y_1}{\ell}$

$$(x-x_0)\sin\theta - (y-y_0)\cos\theta = 0$$

$$x\sin\theta - y\cos\theta + \rho = 0 \qquad\qquad \rho = -(x_0\sin\theta - y_0\cos\theta)$$

This last formulation is nice from a number of points of view. It is always non-singular, easy to use and allows the least-squares equations to be formulated neatly. Note that we have a choice of polarity, by multiplying the whole equation by -1. We can choose a preferred direction along or across the line in this fashion. This allows us to represent directed lines.

Now we get on to using this representation. First we note that the equation of a line at right angles through a point (x_3, y_3) is:

$$(x-x_3)\cos\theta + (y-y_3)\sin\theta = 0$$

Our line and one point (x_0, y_0) on it implicitly define a coordinate transformation:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{vmatrix} \begin{vmatrix} x-x_0 \\ y-y_0 \end{vmatrix}$$

The inverse is:

$$\begin{vmatrix} x-x_0 \\ y-y_0 \end{vmatrix} = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix} \begin{vmatrix} x' \\ y' \end{vmatrix}$$

The separation (perpendicular to the line) of two points:

$$(x_2-x)\sin\theta - (y_2-y_1)\cos\theta \qquad \textbf{(A)}$$

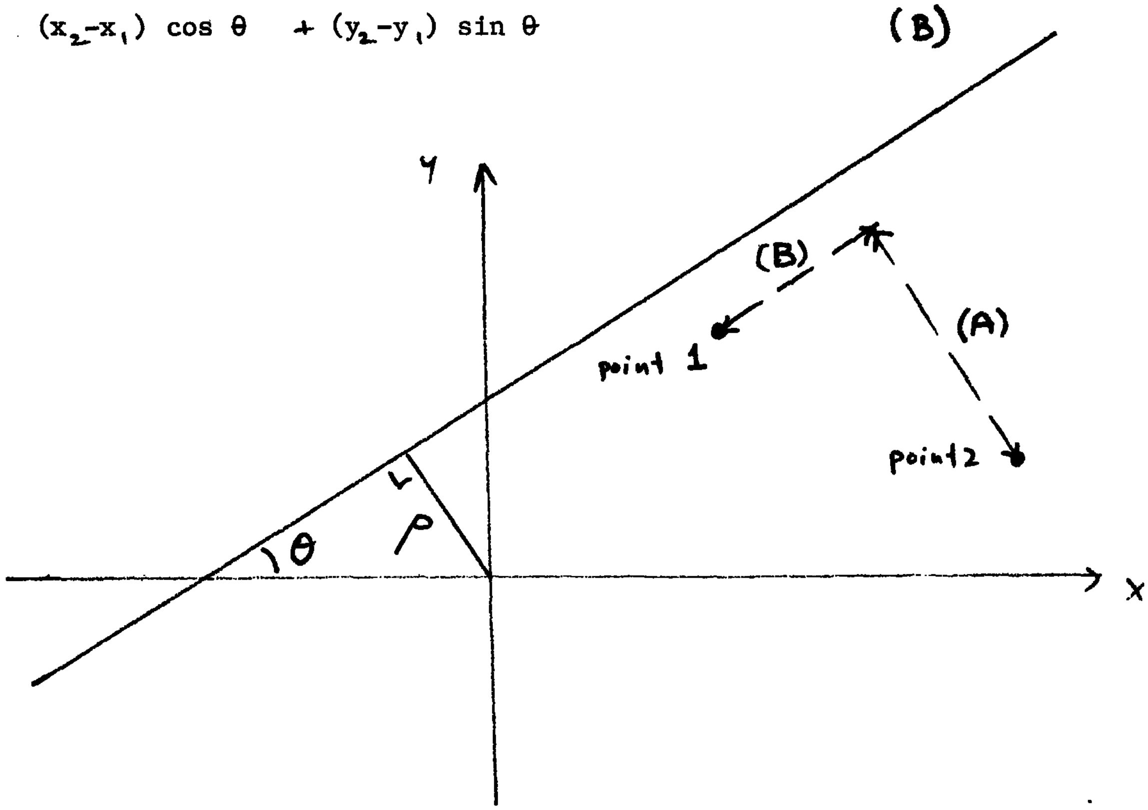
In particular the distance of a point from the line is :

$$x\sin\theta - y\cos\theta + \rho$$

Not surprisingly the equation shows the line to consist of points with zero distance from the line.

The separation (along the line) of two points:

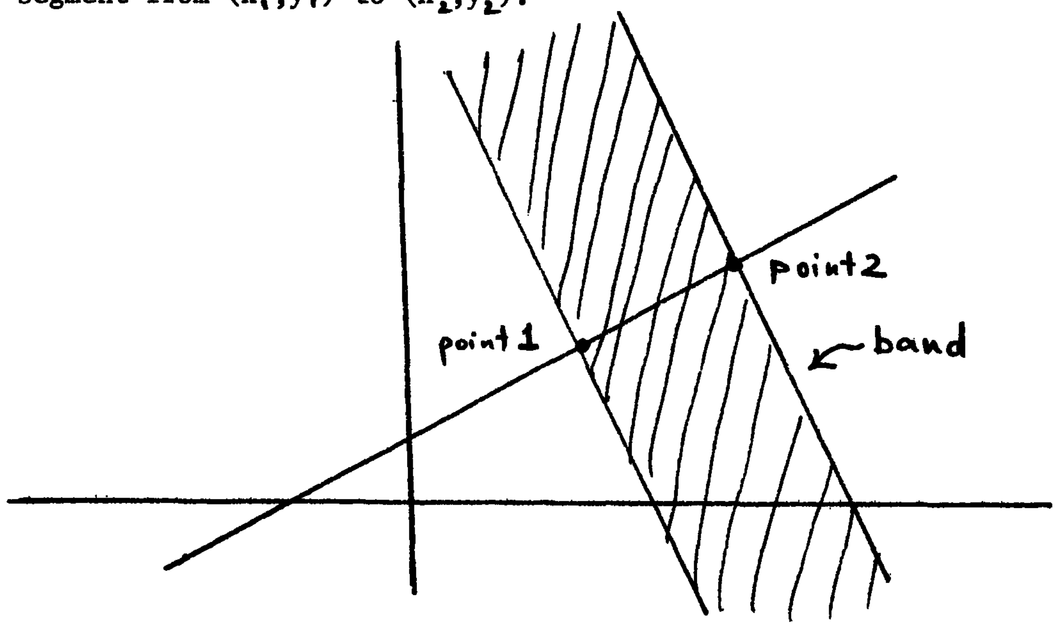
$$(x_2-x_1)\cos\theta + (y_2-y_1)\sin\theta \qquad \textbf{(B)}$$



In particular suppose that the end-points of a line segment are (x_1, y_1) and (x_2, y_2). Then a point lies in the band generated by projecting this line segment perpendicular to the line if:

$$[(x-x_1)\cos\theta + (y-y_1)\sin\theta]\,[(x-x_2)\cos\theta + (y-y_2)\sin\theta] < 0$$

This can be used to determine if a point on the line lies within the line segment from (x_1, y_1) to (x_2, y_2).



The sine of the angle between two lines:

$$\Delta = \sin(\theta_1-\theta_2) = \sin\theta_1 \cos\theta_2 - \cos\theta_1 \sin\theta_2$$

We use this in the equation for the intersection of two lines:

$$x = (\cos\theta_2 \rho_1 - \cos\theta_1 \rho_2)/\Delta$$

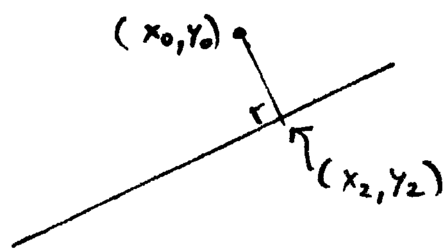
$$y = (\sin\theta_2 \rho_1 - \sin\theta_1 \rho_2)/\Delta$$

Naturally if $\Delta = 0$, the lines are parallel and we loose. To find out if two line-segments intersect, we use these equations to find the intersection of the corresponding lines, then apply the above "band" test twice to see if this point is inside both line-segments.

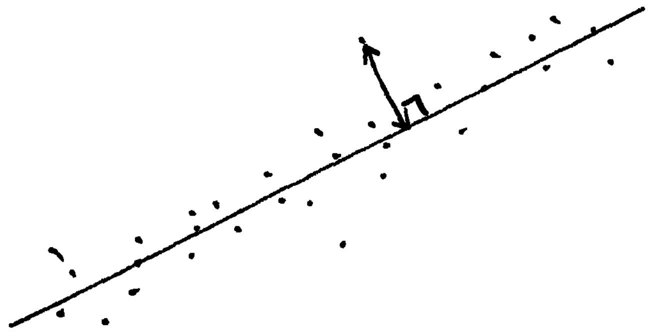
Next, to project a point perpendicularly onto the line we perform two opposite rotations about the origin:

$$\begin{cases} x_1 = x_0 \cos\theta + y_0 \sin\theta \\ y_1 = \rho \end{cases}$$

$$\begin{cases} x_2 = x_1 \cos\theta - y_1 \sin\theta \\ y_2 = x_1 \sin\theta + y_1 \cos\theta \end{cases}$$



Next we get to least-squares fitting a line to a set of points:



We minimise the sum of squares of perpendicular distances of the points from the line (moment of inertia):

$$e^2 = \sum_i (x_i \sin\theta - y_i \cos\theta + \rho)^2$$

$$\frac{d}{d\rho} e^2 = 2\Big[\sin\theta \sum_i x_i - \cos\theta \sum_i y_i + \rho \sum_i 1 \Big]$$

For this to be 0 we must have:

$$\rho = -(x_0 \sin\theta - y_0 \cos\theta) \qquad \text{where}$$

$$x_0 = \sum_i x_i \,/n \qquad y_0 = \sum_i y_i \,/n$$

That is, the line passes through the centre of gravity of the points. Changing coordinates to a system relative to this favoured point we get:

$$x_i' = x_i - x_0 \qquad y_i' = y_i - y_0$$

$$e^2 = \sum_i \big[(x_i - x_0) \sin\theta - (y_i - y_0) \cos\theta \big]^2$$

$$= \sum_i (x_i' \sin\theta - y_i' \cos\theta)^2$$

$$= (\sin\theta)^2 \sum_i x_i'^2 - 2(\sin\theta)(\cos\theta) \sum_i x_i' y_i' + (\cos\theta)^2 \sum_i y_i'^2$$

$$= 1/2 \Big[\Big(\sum_i x_i'^2 + \sum_i y_i'^2\Big) - \Big(\sum_i x_i'^2 - \sum_i y_i'^2\Big) \cos 2\theta - 2 \sum_i x_i' y_i' \sin 2\theta \Big]$$

Note that we can find some of these terms as follows:

$$\sum_i x_i' = \sum_i x_i^2 - n\, x_o^2 \qquad \sum_i y_i' = \sum_i y_i^2 - n\, y_o^2$$

$$\sum_i x_i' y_i' = \sum_i x_i y_i - n\, x_o y_o$$

For compactness let:

$$a = 2\sum_i x_i' y'_i \qquad b = \sum_i x_i'^2 - \sum_i y_i'^2 \qquad c = \sqrt{a^2 + b^2}$$

We get:

$$\frac{d}{d\theta} e^2 = b \quad \sin 2\theta \quad - \quad a \quad \cos 2\theta$$

For this to be zero we must have:

$$\tan 2\theta = a/b \text{ , i.e. } \sin 2\theta = a/c \qquad \cos 2\theta = b/c$$

Now $\cos\theta = \sqrt{\dfrac{1+\cos 2\theta}{2}} = \sqrt{\dfrac{b+c}{2c}}$

And $\sin\theta = \dfrac{\sin 2\theta}{2\cos\theta} = (a/2c)/\sqrt{\dfrac{b+c}{2c}}$

If $a = 0$, so is $\sin\theta$. Again we can choose to multiply the whole thing by -1 to decide the direction.

An equivalent way of getting this last result is the following:

$$\tan\theta = \frac{-1 + \sqrt{(\tan 2\theta)^2 + 1}}{(\tan 2\theta)} = \frac{-b + \sqrt{a^2 + b^2}}{a} = \frac{c-b}{a} = \frac{a}{c+b}$$

$$\cos\theta = 1/\sqrt{(1 + (\tan\theta)^2)} = \frac{a}{\sqrt{a^2 + (b-c)^2}} = \frac{a}{\sqrt{2c(c-b)}} = \sqrt{\frac{b+c}{2c}}$$

Different least-squares fits can be described (for example one which minimises the sum of squares of distance parrallel to the y-axis), but this one has the property that the fit does not depend on the coordinate system (invariant with rotation for example).

Note that while trigonometric functions appear all over these results, none ever get evaluated. Trigonometric functions are a most useful intellectual crutch, there is however seldom a need to actually use them in numeric calculations. One can us ually replace them using the well-known relationships amongst them and reduce the required operations to $+, -, *, /$ and $\sqrt{}$ only. In the above formulas for least-square fitting for example the only requirment is for two square root evaluations.

We ought to also check that we in fact have a minimum:

$$\frac{d^2}{d\rho^2} e^2 = n > 0$$

$$\frac{d^2}{d\theta^2} e^2 = 2\,b \cos 2\theta + 2a \sin 2\theta = 2\,\frac{a^2+b^2}{c} = 2\sqrt{a^2+b^2} > 0$$

So indeed we have a minimum. We might also want to know the average error. And the average error if instead we had chosen the worst line (one at right angles to the best line).

$$e_1^2 = (\sin\theta)^2 \sum_i x_i'^2 - 2\cos\theta \sin\theta \sum_i x_i' y_i' + (\cos\theta)^2 \sum_i y_i'^2$$

$$e_2^2 = (\cos\theta)^2 \sum_i x_i'^2 + 2\sin\theta\cos\theta \sum_i x_i' y_i' + (\sin\theta)^2 \sum_i x_i'$$

Let $d = \sum_i x_i'^2 + \sum_i y_i'^2$ then we can also write the above as:

$$e_1^2 = 1/2\ (d - c)$$

$$e_2^2 = 1/2\ (d + c)$$

The ratio can be used as a "form-factor".

Note that all of this line-fitting can be modified to handle weighted points by simple multiplying the coordinates (x_i, y_i) by the weights w_i, and using $\sum_i w_i$ instead of n.

Now suppose we are given several lines and are required to find a point with minimum sum of squares of perpendicular distance.

$$
\begin{aligned}
e^2 &= \sum_i (x \sin \theta_i - y \cos \theta_i + \rho_i)^2 \\
&= x^2 \sum_i (\sin \theta_i)^2 - 2xy \sum_i \sin \theta_i \cos \theta_i + y^2 \sum_i (\cos \theta_i)^2 \\
&\quad\ 2 x \sum_i \sin \theta_i \rho_i - 2 y \sum_i \cos \theta_i \rho_i + \sum_i \rho_i^2
\end{aligned}
$$

$$\frac{d}{dx} e^2 = 2\left(x \sum_i (\sin \theta_i)^2 - y \sum_i \sin \theta_i \cos \theta_i + \sum_i \rho_i \sin \theta_i \right)$$

$$\frac{d}{dy} e^2 = 2\left(-x \sum_i \sin \theta_i \cos \theta_i + y \sum_i (\cos \theta_i)^2 - \sum_i \rho_i \cos \theta_i \right)$$

Let $\Delta = \sum_i (\sin \theta_i)^2 \cdot \sum_i (\cos \theta_i)^2 - \left(\sum_i \sin \theta_i \cos \theta_i \right)^2$

This will only be zero if all the lines are parallel. Solving the above set of equations in x and y we get:

$$x = \left[-\sum_i (\cos \theta_i)^2 \sum_i \rho_i \sin \theta_i + \sum_i \sin \theta_i \cos \theta_i \sum_i \rho_i \cos \theta_i \right] / \Delta$$

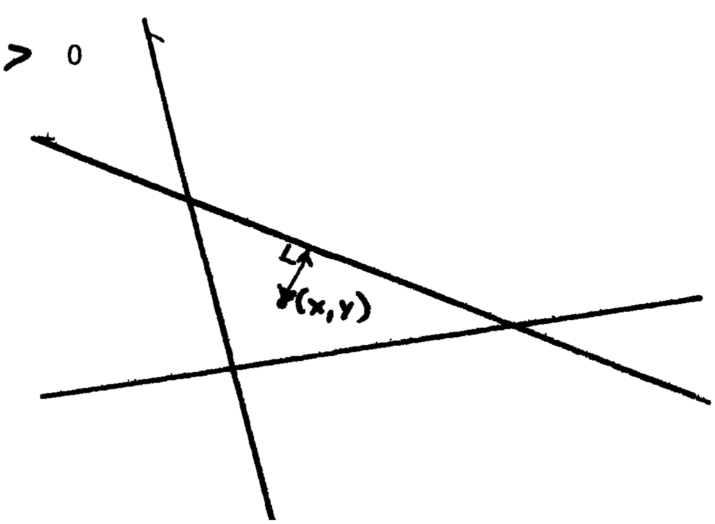
$$y = \left[-\sum_i (\sin \theta_i \cos \theta_i) \sum_i \rho_i \sin \theta_i + \sum_i (\sin \theta_i)^2 \sum_i \rho_i \cos \theta_i \right] / \Delta$$

We also ought to check whether this gives us a minimum:

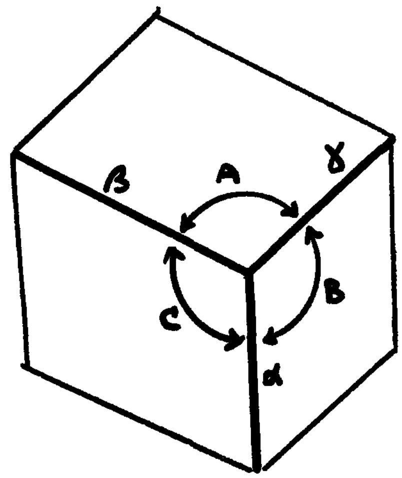
$$\frac{d^2}{dx^2} e^2 = 2 \sum_i (\sin \theta_i)^2 > 0$$

$$\frac{d^2}{dy^2} e^2 = 2 \sum_i (\cos \theta_i)^2 > 0$$

We can weight the lines by simply multiplying $\sin \theta_i$, $\cos \theta_i$, ρ_i by the weights w_i.



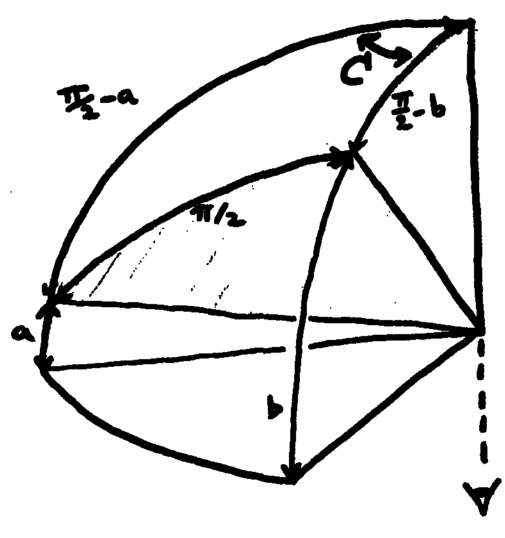
PROJECTION OF A RECTANGULAR CORNER:



Given that the tri-hedral vertex is formed by three planes meeting ar right angles, find the inclinations of the three lines α, β, γ relative to the image plane. Let these angles be a, b, c . Note that the angle between the plane containing β, γ relative to the view vector is also a and so on for the other sides of the object.

This information is useful in defining the elevation and rotation of the eye relative to the coordinate system implicitly defined by the rectangular object. The angles can also be used to correct the fore-shortening introduced by the inclination of the lines relative to the image plane.

Now consider the following spherical triangle:



Using a spherical trigonometry formula we get:

$$0 = \cos\frac{\pi}{2} = \cos(\frac{\pi}{2} - a)\cos(\frac{\pi}{2} - b) + \sin(\frac{\pi}{2} - a)\sin(\frac{\pi}{2} - b)\cos C$$

$$= \sin a \sin b + \cos a \cos b \cos C$$

$\cos C = -\tan a \tan b$ and by symmetry:

$\cos B = -\tan c \tan a$

$\cos A = -\tan b \tan c$

$$\tan^2 a = -\frac{\cos B \cos C}{\cos A}$$

$$\cos a = \frac{1}{\sqrt{1 + \tan^2 a}} = \sqrt{\frac{\cos A}{\cos A - \cos B \cos C}}$$

Where $a < 0$ if and only if $A > \pi$.

So we have the angle of the line α relative to the image plane. As mentioned this also gives us the inclination of the plane containing β and γ relative to the view vector. The others are found by symmetry.

To get the "unforshortened" length of the lines, that is the length in the image if they had been oriented at right angles to the view vector, we just divide by the cosine of the inclination angle:

$$\alpha_u = \alpha_f / \cos a$$

We also note that the cosines needed in the formulae can be obtained simply by using dot-products:

$$\cos A = (\underset{\sim}{\beta} \cdot \underset{\sim}{\gamma}) / \sqrt{(\underset{\sim}{\beta} \cdot \underset{\sim}{\beta})(\underset{\sim}{\gamma} \cdot \underset{\sim}{\gamma})}$$

So we don't need to use trig-functions at all, only +,-,*,/ and $\sqrt{}$.

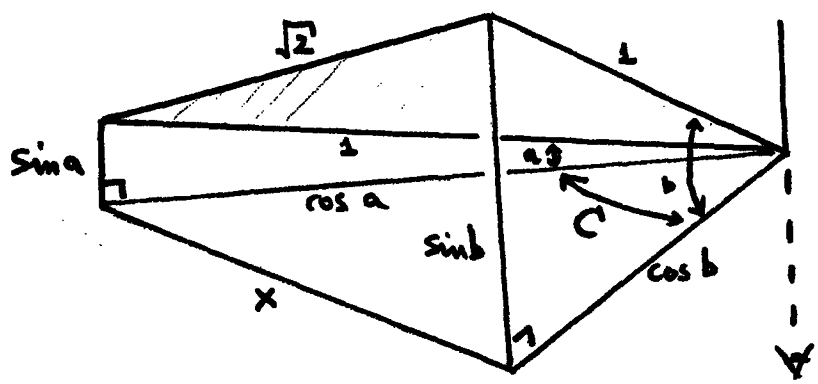
A few other random formulae in this relation:

Since $A + B + C = 2\pi$, $\cos A = \cos B \cos C - \sin B \sin C$

$$\cos a = \sqrt{-\frac{\cos A}{\sin B \sin C}} \qquad \text{because of that.}$$

$$\sin a = \sqrt{\frac{\tan^2 a}{1 + \tan^2 a}} = \sqrt{\frac{\cos B \cos C}{\cos B \cos C - \cos A}} = \sqrt{\frac{\cos B \cos C}{\sin B \sin C}}$$

Another derivation not involving spherical triangles is as follows:



Using the formulae for plane triangles:

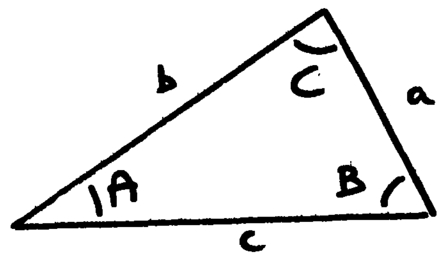
$$(\sin a - \sin b)^2 + x^2 = 2$$

$$x^2 = \cos^2 a + \cos^2 b - 2 \cos a \cos b \cos C$$

$$2 = (\sin^2 a + \cos^2 a) + (\sin^2 b + \cos^2 b) - 2 \sin a \sin b - 2 \cos a \cos b \cos C$$

$\cos C = - \tan a \tan b$ as before

RELATIONS BETWEEN SIDES AND ANGLES OF ANY PLANE TRIANGLE:

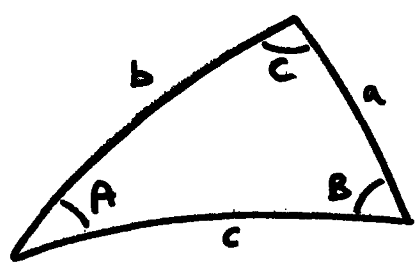


$$\frac{\sin A}{a} = \frac{\sin B}{b} = \frac{\sin C}{c} \quad (= \text{diameter of circumscribed circle})$$

$$a^2 = b^2 + c^2 - 2bc \cos A$$

$$a = b \cos C + c \cos B$$

RELATIONS IN ANY SPHERICAL TRIANGLE:



$$\frac{\sin A}{\sin a} = \frac{\sin B}{\sin b} = \frac{\sin C}{\sin c}$$

$$\cos a = \cos b \cos c + \sin b \sin c \cos A$$

$$\cos A = - \cos B \cos C + \sin B \sin C \cos a$$

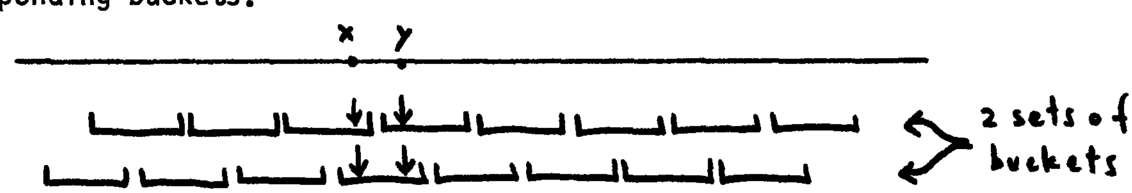
(a, b, c are the length of the arc on a unit sphere, alternatively one can think of them as the angle (in radians) subtended by the arc at the centre of the sphere)

PROXIMITY FINDING:

In the later phases of line-finding programs it is often necessary to repeatetly locate lines that are close together, lines that pass close to a given vertex and so on. To do this efficiently we require a fast access method to locate likely candidates for more sensitive tests. First consider the problem of deciding if two points are close together.

To tell if x and y are close together we can quantise both (by dividing by the quantisation interval size and truncating). If the two numbers $[x/d]$ and $[y/d]$ are the same we win, but it may be that x and y just straddle a boundary defined by our truncation algorithm. We need a second, interlaced set of boundaries and calculate $[x/d+.5]$, $[y/d+.5]$. Now if either pair of numbers matches we know that $|x-y|<d$. Conversly if $|x-y|<d/2$ we are guaranteed that at least one pair will match.

This method only comes into its own if we have large sets of points. We then simply find the two integer-codes for each one and add it to the appropriate buckets. To find which points are near a given point we determine its two integer-codes and collect the union of the two corresponding buckets.



This method can now be extended to n dimensions. We need at least n+1 sets of buckets if we use n-tetrahedrons. It may be more convenient to use 2^n sets of buckets if the unit cell is an n-cube. (d/n versus d/2 min sep)

Line-segments and curves can be handled by entering each point on them into the system. In practice one will only enter a set of points separated by the minimum distance guaranteed by the geometry used. Retrieval works in a symmetric way.

LEAST SQUARES SOLUTION OF AN OVERDETERMINED SET OF EQUATIONS:

Let us write the equations as follows:

$$A\,x = y + e$$

Where A is a given m by n matrix ($m > n$), x is the unknown n-vector, y is a given m-vector and e is an m-vector of errors which we are trying to minimise.

$$\begin{aligned} e^T e &= (A\,x - y)^T\,(A\,x - y) \\ &= (x^T\,A^T - y)\,(A\,x - y) \\ &= x^T\,A^T\,A\,x - y^T\,A\,x - x^T\,A^T\,y + y^T\,y \end{aligned}$$

$$\frac{d}{dx}\,e^T e = x^T\,A^T\,A + (A^T\,A\,x)^T - y^T\,A - (A^T\,y)^T + 0$$

So: $\quad x^T\,A^T\,A = y^T\,A \qquad$ for this to be 0

$$A^T\,A\,x = A^T\,y$$

$$x = (A^T\,A)^{-1}\,A^T\,y$$

$$\frac{d^2}{dx^2}\,e^T e = 2\,A^T A$$

The diagonal elements of this will clearly be postive so we do have a minimum.

LEAST SQUARES CURVE FITTING:

Suppose we have a function $g(x)$ defined at the n points x_1, x_2 etc., and that we are trying to fit a function $f(x)$ which depends on the m parameters a_1, a_2 etc. so as to minimise the sum of squares of errors at the points x_1, x_2 etc. Let e_i be the error at point x_i.

$$e_i = f(x_i) - g(x_i)$$

Let $\underline{e}$ be the n-vector of errors, $\underline{f}$ the n-vector of fitted values, $\underline{g}$ the n-vector of defined values. Then we are trying to minimise:

$$\underline{e}^T\underline{e} = (\underline{f} - \underline{g})^T (\underline{f} - \underline{g})$$

by varying the parameter m-vector $\underline{a}$. The derivative of $\underline{e}^T\underline{e}$ w.r.t. to this vector must be zero (and the second derivative positive).

$$(\underline{f} - \underline{g}) \frac{d\underline{f}}{d\underline{a}} = 0 \qquad \text{where } \frac{d\underline{f}}{d\underline{a}} \text{ is an n by m matrix}$$

So: $\underline{g} \dfrac{d\underline{f}}{d\underline{a}} = \underline{f} \dfrac{d\underline{f}}{d\underline{a}}$ or written out in full:

$$\begin{vmatrix} \frac{df(x_1)}{da_1} & \frac{df(x_2)}{da_1} & \vdots & \frac{df(x_n)}{da_1} \\ \frac{df(x_1)}{da_2} & \frac{df(x_2)}{da_2} & \vdots & \frac{df(x_n)}{da_2} \\ \cdots & & & \cdot \\ \frac{df(x_1)}{da_m} & \frac{df(x_2)}{da_m} & \cdot & \frac{df(x_n)}{da_m} \end{vmatrix} \begin{vmatrix} g(x_1) \\ g(x_2) \\ \\ g(x_n) \end{vmatrix} = \begin{vmatrix} \frac{df(x_1)}{da_1} & \frac{df(x_2)}{da_1} & \vdots & \frac{df(x_n)}{da_1} \\ \frac{df(x_1)}{da_2} & \frac{df(x_2)}{da_2} & \vdots & \frac{df(x_n)}{da_2} \\ \cdots & & & \cdot \\ \frac{df(x_1)}{da_m} & \frac{df(x_2)}{da_m} & \cdot & \frac{df(x_n)}{da_m} \end{vmatrix} \begin{vmatrix} f(x_1) \\ f(x_2) \\ \\ f(x_n) \end{vmatrix}$$

To be able to solve these equations we choose a particularly simple form for $f(x)$ namely $\sum a_j f_j(x_i) = f(x_i)$, that is a linear one. In this case the terms in the matrix $d\underline{f}/d\underline{a}$, namely $df(x_i)/da_j$ become:

$$\frac{df_i}{da_j} = f_j(x_i)$$

So the matrices become quite simple and let us denote them by F^T

$$F_{ji} = \frac{df_i}{da_j} = f_j(x_i)$$

We also note that because of the simple dependence of f(x) on the parameters a_i we get:

$$\underline{f} = F \; \underline{a}$$

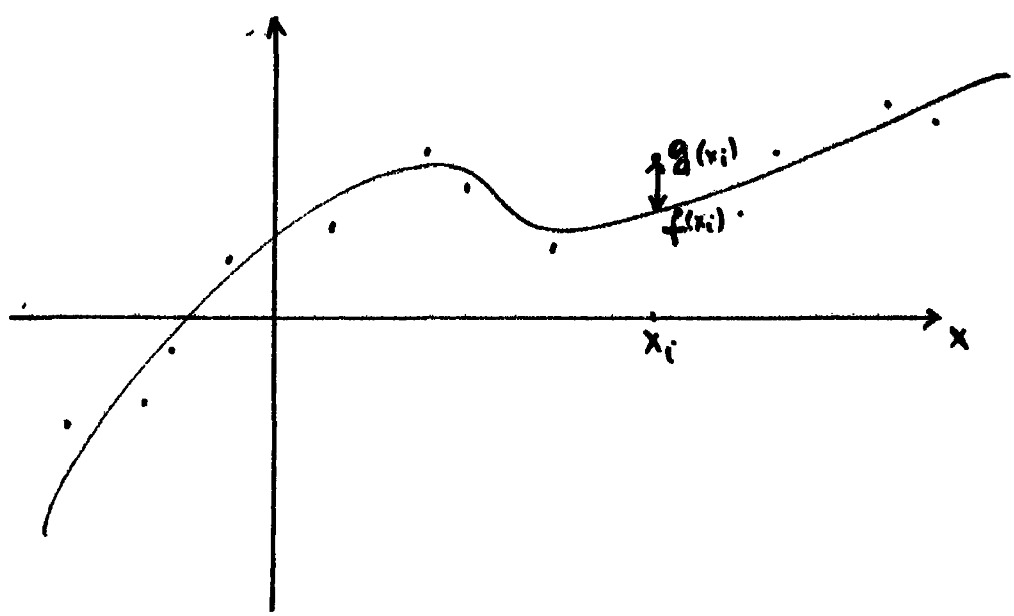
And we can rewrite the main equation as:

$$F^T\underline{g} = F^T F \; \underline{a}$$

Since $F^T F$ is square we can attempt to invert it and get:

$$\underline{a} = (F^T F)^{-1} \; F^T \; \underline{g}$$

So inverting the normal matrix allows us to solve for the parameters $\underline{a}$.



EXAMPLE OF FITTING A STRAIGHT LINE:

$f(x) = a_1 + a_2 x$ and let $y_i = g(x_i)$

Here $\underline{a} = (a_1, a_2)$ $\underline{g} = (y_1, y_2, \ldots y_n)$ $f_1(x)=1$ $f_2(x)=x$

$$F^T = \begin{vmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \end{vmatrix} \qquad F^T F = \begin{vmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{vmatrix}$$

Let $\Delta = n\Sigma x_i^2 - (\Sigma x_i)^2$, then

$$(F^T F)^{-1} = \frac{1}{\Delta}\begin{vmatrix} \Sigma x_i^2 & -\Sigma x_i \\ -\Sigma x_i & n \end{vmatrix} \qquad F^T \underline{g} = \begin{vmatrix} \Sigma y_i \\ \Sigma x_i y_i \end{vmatrix}$$

$$\underline{a} = (F^T F)^{-1} F^T \underline{g} = \frac{1}{\Delta}\begin{vmatrix} \Sigma x_i^2 & -\Sigma x_i \\ -\Sigma x_i & n \end{vmatrix}\begin{vmatrix} \Sigma y_i \\ \Sigma x_i y_i \end{vmatrix}$$

$$a_1 = (\Sigma x_i^2 \Sigma y_i - \Sigma x_i \Sigma x_i y_i) / \Delta$$

$$a_2 = (-\Sigma x_i \Sigma y_i + n \Sigma x_i y_i) / \Delta$$

Note that this is an unsymmetrical method different from the one demonstrated elsewhere in this memo and not suited for fitting lines in a line-drawing for example.

APPLICATION TO FITTING A POLYNOMIAL:

$f(x) = \sum_{j=0}^{m-1} a_j \, x^j$ and let $y_i = g(x_i)$

$\underline{a} = (a_0, a_1 \dots a_{m-1})$ $\underline{g} = (y_0, y_1, \dots y_n)$

$f_j(x) = x^j$

$$F^T = \begin{vmatrix} 1 & 1 & & 1 \\ x_1 & x_2 & \dots & x_n \\ \cdot & & & \cdot \\ x_1^{m-1} & & \dots & x_n^{m-1} \end{vmatrix}$$

$$F^T F = \begin{vmatrix} n & \Sigma x_i & \Sigma x_i^2 & \dots & \Sigma x_i^{m-1} \\ \Sigma x_i & \Sigma x_i^2 & & & \vdots \\ \Sigma x_i^2 & & & & \vdots \\ \vdots & & & & \\ \Sigma x_i^{m-1} & & & \dots & \Sigma x_i^{2m-2} \end{vmatrix}$$ The "normal" matrix

$$F^T \underline{g} = \begin{vmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \vdots \\ \Sigma x_i^{m-1} y_i \end{vmatrix}$$

At this stage we obtain the parameters $\underline{a} = (F^T F)^{-1} F^T \underline{g}$

FITTING EXPONENTIALS:

$$f(x_i) = \sum_{j=0}^{m-1} a_j (s_j)^{x_i} \qquad \underline{g} = (y_0, y_1, \quad y_{n-1})$$

$$f_j(x) = (s_j)^x$$

$$F^T = \begin{vmatrix} s_0^{x_0} & s_0^{x_1} & & s_0^{x_{n-1}} \\ s_1^{x_0} & s_1^{x_1} & & s_1^{x_{n-1}} \\ s_{m-1}^{x_0} & s_{m-1}^{x_1} & & s_{m-1}^{x_{n-1}} \end{vmatrix}$$

$$(F^TF)_{ij} = \sum_{k=0}^{n-1} (s_i s_j)^{x_k}$$

Now suppose we have regular intervals $x_k = k\tau$

Let $s_i' = s_i^{\tau}$ and $s_j' = s_j^{\tau}$

$$(F^TF)_{ij} = \sum_{k=0}^{n-1} (s_i s_j)^{k\tau} = \sum_{k=0}^{n-1} (s_i' s_j')^k = (1 - (s_i' s_j')^n)/(1 - s_i' s_j')$$

Next take the special case: $s_a' = e^{(2\pi i/n)a}$, $w = e^{-(2\pi j/n)}$

$(F^TF)_{ij} = 0$ for $i+j \neq 0$ or n $\qquad$ $(F^TF)_{ij} = n$ for $i+j = 0$ or n

$$(F^TF) = n \begin{vmatrix} 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 1 \\ 0 & 0 & 0 & & 1 & 0 \\ & & & \iddots & & \\ 0 & 1 & 0 & & 0 & 0 \end{vmatrix} \qquad (F^TF)^{-1} = \frac{1}{n} \begin{vmatrix} 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 1 \\ 0 & 0 & 0 & & 1 & 0 \\ & & & \iddots & & \\ 0 & 1 & 0 & & 0 & 0 \end{vmatrix}$$

$$\underline{a} = (F^TF)^{-1} \times \begin{vmatrix} \sum w^{-0k} y_k \\ \sum w^{-1k} y_k \\ \\ \sum w^{-(n-1)k} y_k \end{vmatrix} = \frac{1}{n} \begin{vmatrix} \sum w^{0k} y_k \\ \sum w^{1k} y_k \\ \\ \sum w^{(n-1)k} y_k \end{vmatrix}$$

Where we used $w^{(n-a)k} = w^{-ak}$. Note: $\underline{a}$ is the discrete F.T. of $\underline{g}$.

APPLICATION TO FITTING NON HARMONICALLY RELATED SINES AND COSINES:

Assume regular sample intervals: $x_i = i\tau$. Let $s_i = 2\pi f_i \tau$, where f_i are the frequencies of the various components. The s_i should be non-zero, positive and unique.

$$f(x_i) = a_0 + \sum_{j=1}^{m} a_j \cos(s_j\, i) + \sum_{j=1}^{m} b_j \sin(s_j\, i)$$

$$F^T = \begin{vmatrix} 1 & 1 & 1 & & 1 \\ 1 & \cos(s_1) & \cos(2s_1) & \cdots & \cos((n-1)s_1) \\ 1 & \cos(s_2) & \cos(2s_2) & \cdots & \cos((n-1)s_2) \\ & \cdots & & & \cdots \\ 1 & \cos(s_m) & \cos(2s_m) & \cdots & \cos((n-1)s_m) \\ 0 & \sin(s_1) & \sin(2s_1) & \cdots & \sin((n-1)s_1) \\ 0 & \sin(s_2) & \sin(2s_2) & \cdots & \sin((n-1)s_2) \\ & \cdots & & & \cdots \\ 0 & \sin(s_m) & \sin(2s_m) & \cdots & \sin((n-1)s_m) \end{vmatrix}$$

Now note that

$$2 \cos A \cos B = \cos(A+B) + \cos(A-B)$$
$$2 \sin A \cos B = \sin(A+B) + \sin(A-B)$$
$$2 \sin A \sin B = \cos(A-B) - \cos(A+B)$$

and let $s_0=0$

$$(F^TF)_{i,j} = \sum_{k=0}^{n-1} \cos(s_i k)\cos(s_j k) = (1/2) \sum \cos((s_i+s_j)k) + \sum \cos((s_j-s_i)k)$$
for $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant m$

$$(F^TF)_{i,j+m} = \sum_{k=0}^{n-1} \cos(s_i k)\sin(s_j k) = (1/2) \sum \sin((s_i+s_j)k) + \sum \sin((s_j-s_i)k)$$
for $0 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m$

$$(F^TF)_{i+m,j+m} = \sum_{k=0}^{n-1} \sin(s_i k)\sin(s_j k) = (1/2) \sum \cos((s_j-s_i)k) - \sum \cos((s_i+s_j)k)$$
for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m$

The other terms can be found using the symmetry of (F^TF).

Now $$\sum_{k=0}^{n-1} e^{jwk} = (1 - e^{jwn})/(1 - e^{jw})$$

$$= (e^{jwn/2}/e^{jw/2})\ (e^{jwn/2} - e^{-jwn/2})/(e^{jw/2} - e^{-jw/2})$$

$$= e^{jw(n-1)/2}\ \sin(nw/2) / \sin(w/2)$$

Now since $\cos(A) = \mathrm{Re}\,(e^{jA})$ and $\sin(A) = \mathrm{Ig}\,(e^{jA})$:

$$\sum_{k=0}^{n-1} \cos(wk) = \cos(w(n-1)/2) \sin(nw/2) / \sin(w/2)$$

$$= (1/2)\ (\sin(w/2) + \sin((2n-1)w/2))/ \sin(w/2)$$

$$= (1/2)\ (1 + \sin((2n-1)w/2) / \sin(w/2)\ )$$

unless $w = 0$ in which case the sum is n.

$$\sum_{k=0}^{n-1} \sin(wk) = \sin(w(n-1)/2) \sin(nw/2) / \sin(w/2)$$

$$= (1/2)\ (\cos(w/2) - \cos((2n-1)w/2))/ \sin(w/2)$$

unless $w = 0$ in which case the sum is 0.

$$(F^TF)_{i,j} = \frac{1}{4}\left(2 + \frac{\sin((2n-1)(s_i+s_j)/2)}{\sin((s_i+s_j)/2)} + \frac{\sin((2n-1)(s_i-s_j)/2)}{\sin((s_i-s_j)/2)}\right)$$

for $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant m$ and $i \neq j$. If $i=j$, the third term is $2n-1$.

for $i = j = 0$, the second and third term become $2n-1$.

$$(F^TF)_{i,j+m} = \frac{1}{4}\left(\frac{\cos((s_i+s_j)/2) - \cos((2n-1)(s_i+s_j)/2)}{\sin((s_i+s_j)/2)} + \frac{\cos((s_j-s_i)/2) - \cos((2n-1)(s_j-s_i)/2)}{\sin((s_j-s_i)/2)}\right)$$

for $0 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m$ and $i \neq j$. If $i=j$, the second term is 0.

$$(F^TF)_{i+m,j+m} = \frac{1}{4}\left(\frac{\sin((2n-1)(s_j-s_i)/2)}{\sin((s_j-s_i)/2)} - \frac{\sin((2n-1)(s_j+s_i)/2)}{\sin((s_i+s_j)/2)}\right)$$

for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m$ and $i \neq j$. If $i=j$ the first term is $2n-1$.

The first element in this array is n, the other diagonals are near n/2, while most of the other terms are small, of the order of 1. The only large elements will be the result of two narrowly separated frequencies. This makes for good numerical stability when inverting $(F^T F)$ by the simplest methods.

When the frequencies are harmonically related , we have $s_i = (2\pi\ i/n)$. Then all off-diagonal terms will be 0, and those on the diagonal will be exactly n/2 except the first which will be n. The inverse of $(F^T F)$ then is also diagonal with the first element 1/n and the rest 2/n. We are back to discrete fourier transforms in this case.

Note that if we use:

$$\sin(A+B) = \sin A \cos B + \cos A \sin B$$
$$\sin(A-B) = \sin A \cos B - \cos A \sin B$$
$$\cos(A+B) = \cos A \cos B - \sin A \sin B$$
$$\cos(A-B) = \cos A \cos B + \sin A \sin B$$

we can obtain all the entries in the array using only a few operations on $\sin(s_i/2)$, $\cos(s_i/2)$ and $\sin((2n-1)s_i/2)$, $\cos((2n-1)s_i/2)$.

SOLVING SETS OF SIMULTANEOUS LINEAR DIFFERENCE EQUATIONS:

$$(x_1)_{n+1} = a_{11}\ (x_1)_n + a_{12}\ (x_2)_n \quad \cdots \quad + a_{1m}(x_m)_n$$

$$(x_2)_{n+1} = a_{21}\ (x_1)_n + a_{22}\ (x_2)_n \quad \cdots \quad + a_{2m}(x_m)_n$$

$$\cdots$$

$$(x_m)_{n+1} = a_{m1}\ (x_1)_n + a_{m2}\ (x_2)_n \quad \cdots \quad + a_{mm}(x_m)_n$$

$\underline{x}_{n+1} = A\ \underline{x}_n$ where A is the given coefficient set

Assume a solution of the form r^n for each x_i:

$\underline{x}_n = \underline{a}\ r^n$ where $\underline{a}$ is a m-vector of parameters

$r(\underline{a}r^n) = A(\underline{a}r^n)$ $(A-Ir)\underline{a} = 0$ since $r^n \neq 0$

A non-zero solution for $\underline{a}$ requires that det(A-Ir) = 0. The possible $\underline{a}$'s are eigenvectors, the possible r's are eigenvalues of the matrix A. The determinant is a polynomial of degree m in r and will usually have m solutions, possible complex. We get the usual problems if two roots coincide and have to introduce additional solutions of the form nr^n, n^2r^n and so on. Having found r, we can solve for $\underline{a}$ using some normalising conditions (since any multiple will also be a solution). Then using the linearity of the set of equations we can add up the solutions into a more general one:

$\underline{x}_n = \sum_{j=1}^{m} \underline{a}_j\ (r_j)^n$ where r_j are the various solutions

Often we are only interested in stability and just check the roots:

$$|r_j| < 1$$

EXAMPLE FOR A TWO VARIABLE SYSTEM:

$$(x_1)_{n+1} = a_{11}\ (x_1)_n + a_{12}(x_2)_n$$

$$(x_2)_{n+1} = a_{21}\ (x_1)_n + a_{22}(x_2)_n$$

$$\det \begin{vmatrix} (a_{11} - r) & a_{12} \\ a_{21} & (a_{22} - r) \end{vmatrix} = (a_{11}-r)(a_{22}-r) - a_{12}a_{21} = 0$$

$$(a_{11}a_{22} - a_{12}a_{21}) - (a_{11} + a_{22})r + r^2 = 0$$

$$r = (1/2)\left((a_{11}+a_{22}) \pm \sqrt{(a_{11}+a_{22})^2 - 4(a_{11}a_{22}-a_{12}a_{21})} \right)$$

$$r = (1/2)\left((a_{11}+a_{22}) \pm \sqrt{(a_{11}-a_{22})^2 + 4\ a_{12}a_{21}} \right)$$

Stability: When is $\left|(a \pm \sqrt{a^2 - 4b})\right| < 2$?

Case 1: $4b > a^2$ Complex roots $|a| < 2$ for stability

Case 2: $4b < a^2$ Real roots $a > 0.$ $a+\sqrt{a^2-4b} < 2$

$a < b+1$

Case 3: $4b < a^2$ Real roots $a < 0.$ $-a+\sqrt{a^2-4b} < 2$

$-a < b+1$

Case 2 &3 $4b < a^2$ Real roots $|a| < b+1$ for stability

Substituting:

Case 1: $4(a_{11}a_{22}-a_{12}a_{21}) > (a_{11}+a_{22})^2$

$-4\ a_{12}a_{21} > (a_{11}-a_{22})^2$ $|a_{11}+a_{22}| < 2$

Case 2 & 3: Opposite of above relations $|a_{11}+a_{22}| < 1+(a_{11}a_{22}-a_{12}a_{21})$

MULTI-DIMENSIONAL NEWTON-RAPHSON ZERO-FINDING:

Suppose wehave n functions F_i each of n parameters a_j . We are trying to find values for the a_j's such that the F_i's are all zero.

Assume we have the value $\underline{a}_n$ at step n for the parameter vector.
This gives us the value for the function vector $\underline{F}_n = F(\underline{a}_n)$.
Now consider a small change $d\underline{a}$ in $\underline{a}$. To a first approximation we get:

$$F(\underline{a}_n + d\underline{a}) = F(\underline{a}_n) + F'(\underline{a}_n)\, d\underline{a}$$

Where $F'(\underline{a}_n)$ is the matrix of derivatives:

$$\begin{bmatrix} \frac{dF_1}{da_1} & \frac{dF_1}{da_2} & \cdots & \frac{dF_1}{da_n} \\ \frac{dF_2}{da_1} & \frac{dF_2}{da_2} & \cdots & \frac{dF_n}{da_2} \\ \cdots & & & \cdots \\ \frac{dF_n}{da_1} & \frac{dF_n}{da_2} & \cdots & \frac{dF_n}{da_n} \end{bmatrix}$$

For $F(\underline{a}_n + d\underline{a}) = 0$ we need $\quad -F(\underline{a}_n) = F'(\underline{a}_n)\, d\underline{a}$

So $\quad d\underline{a} = -(F'_n)^{-1}\, F_n$

$$\underline{a}_{n+1} = \underline{a}_n - (F'_n)^{-1}\, F_n$$

So we iterate to a solution, requiring one matrix inversion per step. There are better methods, but few simpler. For bad hill-climbing type problems one can use the method of conjugate directions and various variations such as the so-called mixed method which is also fairly rapid.

SIMPLE INTERPOLATION OF A FUNCTION FROM A STORED GRID:

Rectangular grid: Suppose the origin is x_0, y_0 and the spacing d.

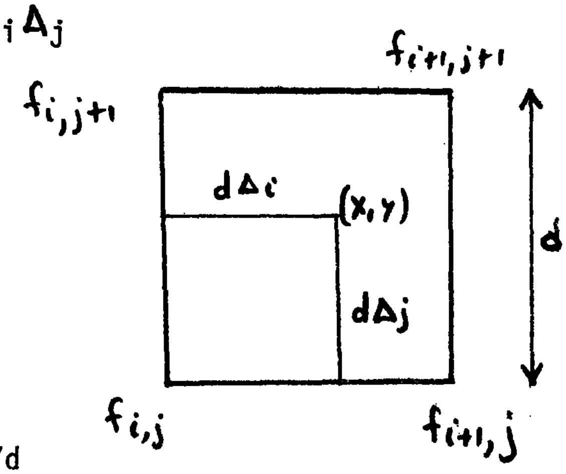
To find an interpolated value at a point x, y calculate as follows:

$x' = (x - x_0)/d \qquad y' = (y - y_0)/d$

$i = [x'] \qquad j = [y']$

$\Delta i = x' - i \qquad \Delta j = y' - j$

$$f(x,y) \cong f_{i,j}(1-\Delta i-\Delta j-\Delta i\,\Delta j)+f_{i+1,j}\Delta i(1-\Delta j)$$
$$+f_{i,j+1}(1-\Delta i)\Delta j + f_{i+1,j+1}\Delta i\,\Delta j$$



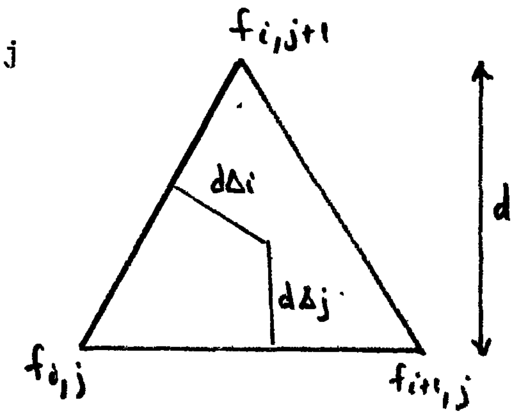
Triangular grid: Again origin at x_0, y_0

$x' = \left(\frac{\sqrt{3}}{2}(x-x_0)-\frac{1}{2}(y-y_0)\right)/d \qquad y'=(y-y_0)/d$

$i = [x'] \qquad j = [y']$

$\Delta i = x' - i \qquad \Delta j = y' - j$

$$f(x,y) \cong f_{i,j}(1-\Delta i-\Delta j) + f_{i+1,j}\Delta i + f_{i,j+1}\Delta j$$



WHAT ELLIPSE IS IT:

Given an ellipse in the form:

$$A\,x^2 + B\,xy + C\,y^2 + D\,x + E\,y + F = 0$$

Determine its center, angular orientation and major and minor axes.

$$x_0 = (BE - 2CD)/(B^2 - 4AC) \qquad y_0 = (2EA - DB)/(B^2 - 4AC)$$

x_0, y_0 are the center because we can expand as follows:

$$A(x-x_0)^2+B(x-x_0)(y-y_0)+C(y-y_0)^2+F'=0$$

$$\begin{aligned} A\,x^2 + B\,xy + C\,y^2 \quad & + (-2Ax_0-By_0)x + (-2Cy_0-Bx_0)y \\ & + (F'+Ax_0^2+Bx_0y_0+Cy_0^2) \qquad = 0 \end{aligned}$$

So we have:

$$\begin{aligned} 2Ax_0 + By_0 &= -D \\ Bx_0 + 2Cy_0 &= -E \end{aligned}$$

Solving this set of equations we get the above expression for x_0, y_0.

We also now have a useful new quantity:

$$F' = F - (Ax_0^2 + Bx_0y_0 + Cy_0^2)$$

The orientation of the ellipse is found as follows:

$$\tan 2\,\theta = B/(A-C)$$

And the major and minor axes can be found as well:

$$a^2 = -2F'/((A+C) - \sqrt{B^2 + (A-C)^2})$$

$$b^2 = -2F'/((A+C) + \sqrt{B^2 + (A-C)^2})$$

These last results follow from expansion after change of coordinates:

$$\left(\frac{x \cos\theta + y \sin\theta}{a}\right)^2 + \left(\frac{-x \sin\theta + y \cos\theta}{b}\right)^2 = 1$$

$$\left(\frac{\sin\theta}{b^2} + \frac{\cos\theta}{a^2}\right) x^2 + 2 \sin\theta \cos\theta\left(\frac{1}{a^2} - \frac{1}{b^2}\right) xy + \left(\frac{\sin\theta}{a^2} + \frac{\cos\theta}{b^2}\right) y^2 = 1$$

Identifying the appropriate terms with A, B, C and F' we get:

$$B/(-F') = \sin 2\theta \left(\frac{1}{a^2} - \frac{1}{b^2}\right)$$

$$(A-C)/(-F') = \cos 2\theta \left(\frac{1}{a^2} - \frac{1}{b^2}\right)$$

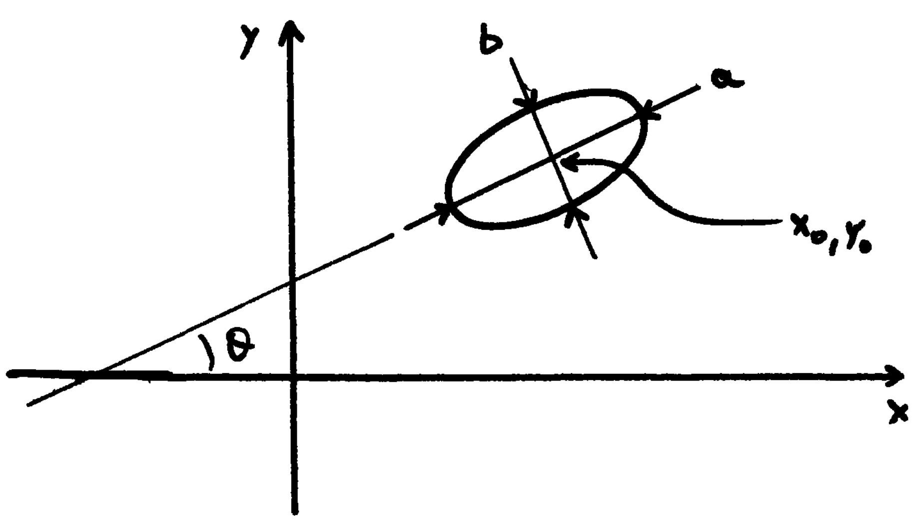
Since $2 \sin\theta \cos\theta = \sin 2\theta$ and $(\cos\theta)^2 - (\sin\theta)^2 = \cos 2\theta$

$$(A+C)/(-F') = \left(\frac{1}{a^2} + \frac{1}{b^2}\right)$$

Its also clear that:

$$\left(\frac{1}{a^2} - \frac{1}{b^2}\right) = \sqrt{B^2 + (A-C)^2}/(+F') \quad \text{(Assuming } a > b\text{)}$$

The rest follows from these simple equations.



APPROXIMATION TO n!

Stirling's formula: $n! \cong \sqrt{2\pi n}\,(n/e)^n$

Better approximation: $n! \cong \sqrt{2\pi n}\,(n/e)^n\, e^{(1/12n)}$

The fractional error of the latter is much smaller. For n=10 for example it is .27E-5 versus .8E-2 and for n=50 it is .22E-7 versus .1E-3. This is useful in calculating large binomial coefficients.

OBTAINING A NORMALLY DISTRIBUTED RANDOM VARIABLE FROM ONE UNIFORMLY DISTRIBUTED:

Suppose x_i is the output of our random (pseudo ...) number generator.

1. $\left(\sum_{i=n}^{n+11} x_i - 6\right)/6$

2. $\sqrt{-2 \log x_i}\ \sin(2\pi x_{i+1})$

3. Let f(x) be the distribution we are aiming for. Now integrate it:

$$F(x) = \int_{-\infty}^{x} f(t)\, dt$$

A random variable distributed as desired will be $(F^{-1})(x_i)$

MULTIPLICATIVE RANDOM GENERATORS:

$x_{i+1} = A^k x_i \pmod{p}$ p a large prime

A a primitive root of p, k not a factor of p-1.

Example: $p = 2^{35} - 31$ $A = 5$ $k = 5$

$p = 2^{31} - 1$ $A = 7$ $k = 5$

Additive congruential generators are better though.

FAST IN-POSITION MATRIX INVERSE:

```
Do  i = 0 ( 1 ) n-1
com ← a(i,i)
a(i,i) ← 1.

     Do  j = 0 ( 1 ) n-1
     a(i,j) ← a(i,j) / com
     End

     Do  k = 0 ( 1 ) n-1  and  i ≠ k
     com ← a(k,i)
     a(k,i) ← 0.

          Do  j = 0 ( 1 ) n-1
          a(k,j) ← a(k,j) - a(i,j) * com
          End

     End

End
```

Note that rows and columns are never shuffled and that there will be matrices which while not singular will cause this procedure to fail.

The matrix is n by n and stored in the array a(i,j) where i and j range from 0 to n-1.

GENERATING A BIT-REVERSING TABLE:

```
b(0) ← 0
m ← 1
Do i = 0 ( 1 ) ln-1
     Do j = 0 ( 1 ) m-1
     b(j) ← b(j)*2
     b(j+m) ← b(j)+1
     End
m ← m*2
End
```

SOME FOURIER TRANSFORM METHODS FOR IMAGES:

Fourier transforms for images are two-dimensional and two-sided. In this they differ from time-series type transforms which are one-dimensional and often pertain to one-sided functions (Impulse responses must be 0 for negative values of time).

The general formula for n-dimensions is: (Note: f and g are complex)

$$g(\underline{u}) = \frac{1}{(2\pi)^{n/2}} \int_V f(\underline{x})\, e^{-i\, \underline{u}\cdot\underline{x}}\, d\underline{x}$$

$$f(\underline{x}) = \frac{1}{(2\pi)^{n/2}} \int_V g(\underline{u})\, e^{+i\, \underline{u}\cdot\underline{x}}\, d\underline{u}$$

Where $\underline{x}$ is the n-dimensional source-space vector, $\underline{u}$ is the n-dimensional transform-space vector and g is the transform of f. For two dimensions:

$$g(u,v) = \frac{1}{2\pi} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i(ux+vy)}\, dx\, dy$$

$$f(x,y) = \frac{1}{2\pi} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v)\, e^{+i(ux+vy)}\, du\, dv$$

Many functions of interest are rotationally symmetric and can be dealt with by use of the one-dimensional integrals obtained after introducing the polar coordinates (r, θ) for (x, y) and (ρ, ϕ) for (u, v).

$$g(\rho) = \int_0^{\infty} f(r)\, r\, J_0(r\rho)\, dr$$

$$f(r) = \int_0^{\infty} g(\rho)\, \rho\, J_0(r\rho)\, d\rho$$

Where J_0 is the zeroth order Bessel function.
Note that f and g are now real-valued.

This follows from:

$$\frac{1}{2\pi}\int_0^\infty\int_0^{2\pi} f(r)\, e^{-i\,(r\rho\cos\phi\cos\theta + r\rho\sin\phi\sin\theta)}\,d\theta\,dr$$

$$= \frac{1}{2\pi}\int_0^\infty f(r)\int_0^{2\pi} e^{-i\,r\rho\,\cos(\phi-\theta)}\,d\theta\,dr$$

$$= \frac{1}{2\pi}\int_0^\infty f(r)\;2\pi r\;J_0(r\rho)\;dr$$

We can apply these results to a few useful examples:

The pillbox: $f(r) = 1$ for $r \leq R$, 0 otherwise. This is the point-spread function produced by defocusing.

$$g(\rho) = \int_0^R r\,J_0(r\rho)\,dr = \frac{1}{\rho^2}\int_0^{R\rho} x\,J_0(x)\,dx = \frac{R\rho}{\rho^2}\,J_1(R\rho)$$

$$g(\rho) = R^2\,\frac{J_1(R\rho)}{(R\rho)} \qquad \text{Since } \int_0^\infty xJ_0(x) = xJ_1(x)$$

So the function $J_1(x)/x$ plays the role here that Sin x/x plays for one-dimensional systems.

The gaussian:

$$f(r) = e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2}$$

$$g(\rho) = \int_0^\infty e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2}\,r\,J_0(r\rho)\,dr = \sigma^2\,e^{-\frac{1}{2}(\sigma\rho)^2}$$

$$\text{Since} \int_0^\infty e^{-\alpha x^2}\,J_n(bx)\,x^{n+1}\,dx = \frac{b}{(2a)^{n+1}}\,e^{-\frac{b^2}{4\alpha}}$$

The gaussian has some interesting properties. First it is the only rotationally symmetric function that can be factored into a product of a function of x and a function of y. Secondly it is the only function that "transforms into itself".

The gaussian is also a good first approximation to point-spread functions in some devices (at least for small r).

A scatter function: $f(r) = e^{-\sigma r} / r$

Analysis of total reflections in the face-plate of an imaging device leads to an equation of this form (at least for large r).

$$g(\rho) = \int_0^\infty e^{-\sigma r} J_0(r\rho) \; r/r \; dr$$

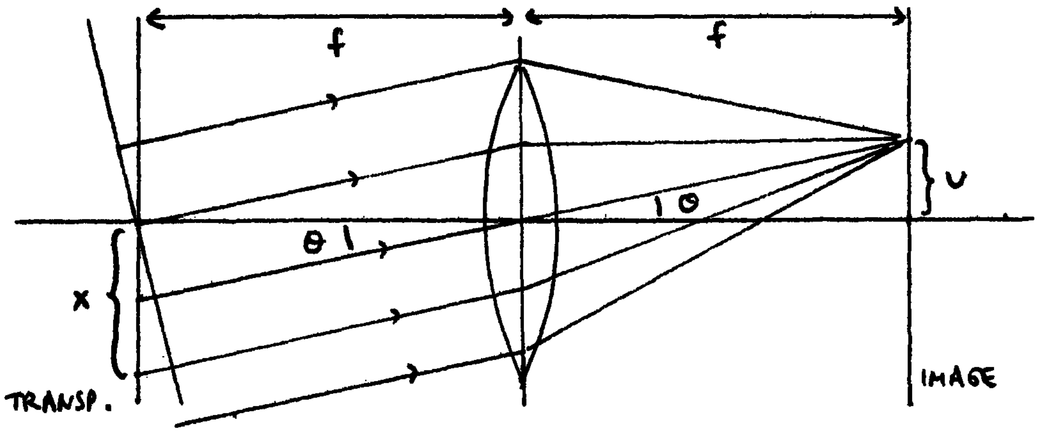
$$= \int_0^\infty e^{-\sigma r} J_0(r\rho) \, dr \qquad = \frac{1}{\sqrt{\sigma^2 + \rho^2}}$$

Note on scaling: For the gaussian we have the following relationship:

$$r_H \rho_H = 1.386 \qquad (r_H = 1.177\sigma \;, \rho_H = 1.177/\sigma)$$

where r_H is the half-intensity radius in the source space, and ρ_H is the half-intensity radius in the transform space.

HOW COHERENT MONOCHROMATIC LIGHT AND A LENS DO FOURIER TRANSFORMS:



Let f be the focal length of the lens and λ the wavelength of light. Plane monochromatic coherent light enters from the left and passes through the transparency, being then focused by the lens on the image plane. We assume that x and u are relatively small compared to f, so that θ will be small. We then have for the distance that the ray has to travel from the point x on the source plane to the point u on the image plane:

$$f/\cos\theta + f\cos\theta + x\sin\theta \quad = \quad f(2 + \theta^4/4 + \theta^6/120 \ldots) + x(\theta - \theta^3/6 \ldots)$$

For small θ this is approximately: $2f + x\,\theta$

The phase-shift in radians is then: $(2f + x\,\theta) * 2\pi/\lambda$

We can ignore the constant part of this and considering that light will arrive at the point u from all over the transparancy we get:

$$g(u) = \int_{-\infty}^{+\infty} f(x)\, e^{2\pi i \frac{x\,u}{\lambda\,f}}\, dx$$

Where f(x) is the amount of light passed through at the point x. Now extend this to two dimensions and we finally have:

$$g(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{2\pi i\left(\frac{xu+yv}{\lambda f}\right)}\, dx\, dy$$

Note that g(u,v) is complex. We can get an idea of scaling from this equation.

SOME HEURISTICS FOR TELLING WHAT HAPPENS WHEN YOU TRANSFORM A FUNCTION

Source domain	Transform domain
Periodic	Discrete (non-zero only for some f)
Symmetric (about 0)	Real
Non-zero for finite distance	Non-zero out to infinity
Compact	Spread-out
Sharp transitions	Lots of high frequency components
Sample of f(t)	Periodic copies of F(w)
Sum of f(t) and g(t)	Sum of F(w) and G(w)
Convolution of f(t) and g(t)	Product of F(w) and G(w)
Time shift of f(t) by	Multiply F(w) by e^{jw}
Integral of f(t)	Divide by jw
Differential of f(t)	Multiply by jw

These rules apply going either way in the transformation and may be used simultaneously. Discrete fourier transforms for example are both periodic and discrete in both domains.

MEASURING MODULATION TRANSFER FUNCTION USING SQUARE WAVES:

It is very hard to produce images in which the intensity varies sinusoidally. Yet such images are required in the traditional determination of frequency response or modulation transfer function. An alternative is the use of simple-to-produce square wave intensity modulated images. Then however we have to recover the transfer function from the measured results.

Let t be one of the spacial dimensions and $\omega = (2\pi)/T$, where T is the repetition interval. The input can be analysed into:

$$f(t) = 1/2 + \sum_{1}^{\infty} b(n) \cos n\omega t \qquad \text{where } b(n) = (2/\pi n)(-1)^{\frac{n-1}{2}}$$

for n odd, 0 otherwise

Let the transfer function be $a(\omega)$. Then the output will be:

$$g(t) = (1/2)a_0 + \sum_{1}^{\infty} b(n)\, a(n\omega) \cos n\omega t$$

We can easily normalise to let $a_0 = 1$. Let $c(\omega) = \sum_{1}^{\infty} b(n)a(n\omega)$.
The problem is to recover $a(\omega)$ from $c(\omega)$. In the case of square-waves:

$$c(\omega) = (2/\pi)\left(a(\omega) - \tfrac{1}{3}a(3\omega) + \tfrac{1}{5}a(5\omega) - \tfrac{1}{7}a(7\omega) + \tfrac{1}{9}a(9\omega) - \tfrac{1}{11}a(11\omega) \ldots\right)$$

$$c(3\omega) = (2/\pi)\left(a(3\omega) - \tfrac{1}{3}a(9\omega) + \tfrac{1}{5}a(15\omega) \ldots\right)$$

$$c(5\omega) = (2/\pi)\left(a(5\omega) - \tfrac{1}{3}a(15\omega) \ldots\right)$$

$$c(7\omega) = (2/\pi)\left(a(7\omega) \ldots\right)$$

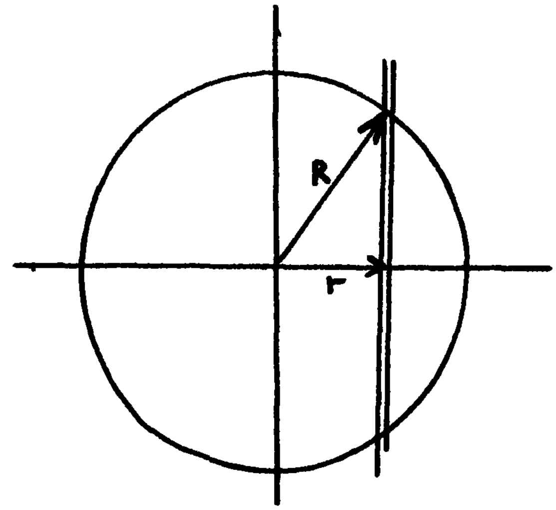
$$c(9\omega) = (2/\pi)\left(a(9\omega) \ldots\right)$$

Now add appropriate high-order terms to $c(\omega)$ to cancel out high-order terms of $a(\omega)$ and get:

$$a(\omega) \cong (\pi/2)\left(c(\omega) + \tfrac{1}{3}c(3\omega) - \tfrac{1}{5}c(5\omega) + \tfrac{1}{7}c(7\omega) + \tfrac{1}{11}c(11\omega) - \tfrac{1}{13}c(13\omega) \ldots\right)$$

CONVOLUTIONS OF PILL-BOXES:

With a line:



Clearly the convolution is $2\sqrt{R^2 - r^2} = 2R\sqrt{1 - (\frac{r}{R})^2}$ for $|r|<R$

This then gives us the intensity profile of a defocused line.

Convolution of a pill-box with a step:

We simply integrate the above:

$$\int_{-R}^{r} 2\sqrt{R^2 - x^2}\,dx$$

$$= \left[x\sqrt{R^2 - x^2} + R^2 \sin^{-1}\left(\frac{x}{R}\right)\right]_{-R}^{r}$$

$$F(r) = \pi R^2 \left(\frac{1}{2} + \frac{1}{\pi}\sin^{-1}\left(\frac{r}{R}\right) + \frac{1}{\pi}\left(\frac{r}{R}\right)\sqrt{1 - \left(\frac{r}{R}\right)^2}\right)$$

So we have the intensity profile of a defocused edge.

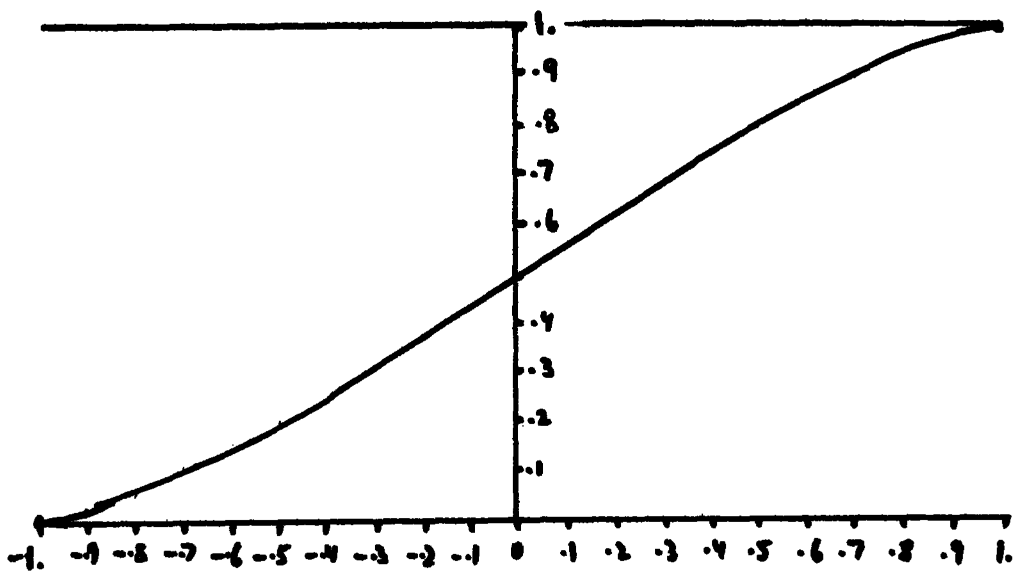
This can be rewritten in a slightly different form using:

$$\sin^{-1} x + \cos^{-1} x = \frac{\pi}{2}$$

We can also use this to find the convolution of two pillboxes as

$$2\,F\left(-\frac{|r|}{2}\right)$$

WHAT A DEFOCUSED EDGE LOOKS LIKE:



Vertical:relative intensity, horizontal: (distance from edge/defocus radius)

Central slope: $2/(\pi R)$, 10% to 90% distance = 1.38 R

Derivative is $(2/\pi R)\sqrt{1 - (r/R)^2}$

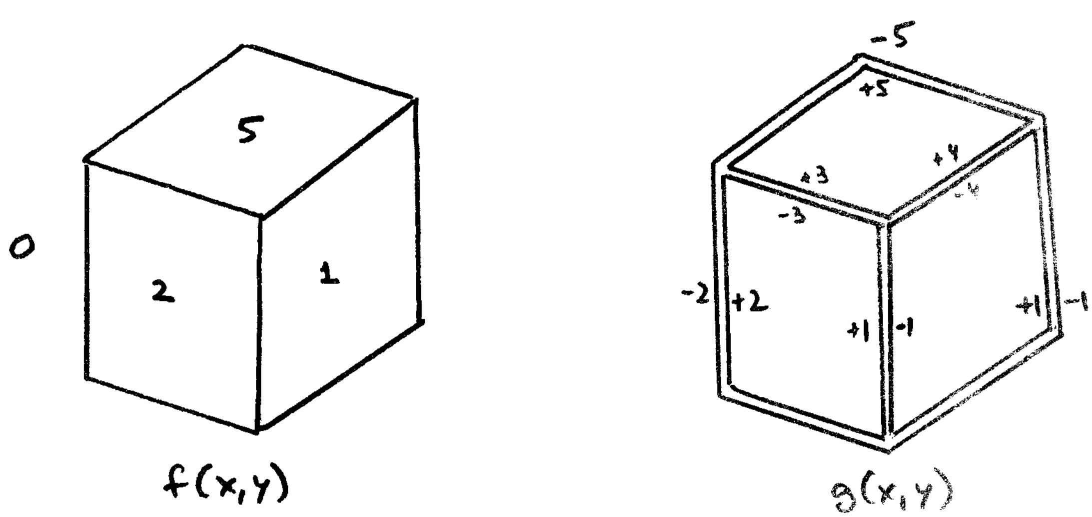
A LINEAR THEORY OF FEATURE POINT MARKING

A first step in many line-finding programs is a process for determining which points in the image are likely to be on an edge. This is usually done by locating areas of rapid intensity variations. Various ad hoc linear and non-linear techniques of varying support in the image are brought into play. It would be useful to have an anchor point on this spectrum of possible procedures. Since a lot is known about linear methods we might ask what linear method applied to a somewhat idealised image would do the job.

Given a function f(x,y) which is constant within polygonal areas in the image, we are looking for a convolution function h(x,y) which when applied to f(x,y) will be zero everywhere except on the edges.

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-x',y-y')\ h(x',y')\ dx'\ dy'$$

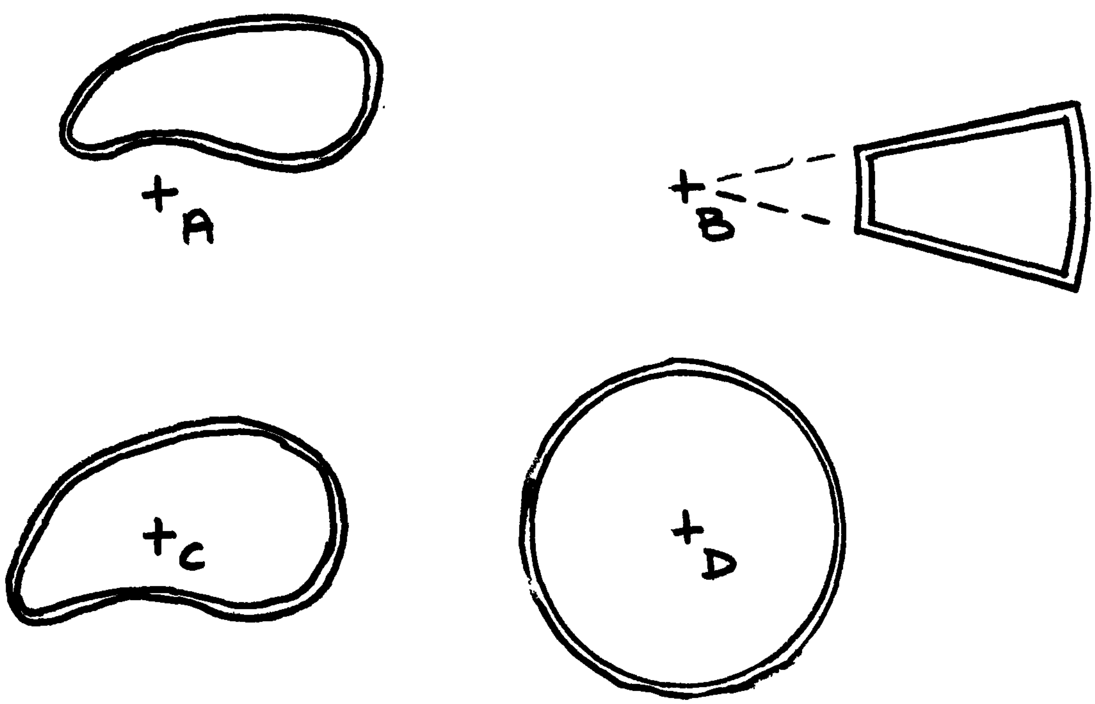
To attempt to answer this question we might start by asking what values we expect g(x,y) to take on the edges. Linearity considerations dictate that it somehow be proportional to the intensity step. In addition it must reflect the orientation of the step, to insure that superposition will work. A combination of a negative and a positive pulse will do the trick, provided the area under each is equal to the intensity step. Since the image is actually two-dimensional we will have two pulse walls running along each edge.



Note by the way that the regions of uniform intensity dont have to be polygonal. Now it is pretty hard to guess what form h(x,y) will take. A way to get a handle on this is to ask the inverse question: what h'(x,y) when convolved with g(x,y) will produce f(x,y) ?

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-x',y-y')h'(x',y')\,dx'\,dy'$$

Well it helps to look at some simple cases first. In particular if we only have one contour (one closed curve made of the double pulsed wall) we expect to get 0 if the convolution is about a point outside this contour, and the intensity step if the point is inside the contour.



A and C illustrate the above statements, while B and D are special cases useful for deriving equations. From D in particular we find that $2\pi r\ (d/dr)\ h'(r) = 1$. This is assuming that h must be rotationally symmetric which is clear from the other examples. We also noted that convolving with the double pulse wall is just like taking the derivative.

$$h'(r) = (1/2\pi) \int (1/r)\,dr = -(1/2\pi)\ \log r$$

Since this function also does the right thing for example B we have the desired result. Now we need to find h(x,y) from this. We do this by finding the fourier transform of h'(x,y) and noting that it must be the algebraic inverse of the transform of h(x,y). Since the functions are rotationally symmetric we get:

$$H'(\rho) = -(1/2\pi)\int_0^{\infty} \log r \;\; r\, J_0(r\rho)\, dr$$

Integrating by parts and using $\int xJ_0(x)\,dx = xJ_1(x)$ as well as $\int J_1(x)\,dx = J_0(x)$ we obtain:

$$H'(\rho) = (1/2\pi\rho^2)$$

To obtain the transform of h(x,y) we just invert this:

$$H(\rho) = 1/H'(\rho) = 2\pi\rho^2$$

When we try to inverse transform this we get into convergence difficulties and soon discover that we have to expand our universe to that of generalised functions if we expect to win, even if we use convergence factors. It then also becomes reasonable to guess at the answer. Consider the sequence of "functions" obtained by repeatetly differentiating a unit step. The first is a pulse at the origin, the second two pulses of opposite sign. This function corresponds to (d/dx) in the following sense: if we convolve it with a function f(x) we obtain the derivative f'(x). Similarly the next member of this sequence consists of a negative, a double height positive and another negative pulse and corresponds to (d^2/dx^2) and so on.

When we try to transform these "functions" we obtain the following: $T(d/dx) = iu$, $T(d^2/dx^2) = -u^2$, $T(d/dy) = iv$, $T(d^2/dy^2) = -v^2$. And:

$$T(\, d^2/dx^2 + d^2/dy^2 \,) = -u^2 - v^2 = -\rho^2 \quad !$$

So our h(x,y) is some multiple of the laplacian $(\, d^2/dx^2 + d^2/dy^2)$.

One can amuse oneself by showing that the convolution of $h(x,y)$ and $h'(x,y)$ is in fact zero everywhere except at the origin as it ought to be:

$$h(x,y) \circledast h'(x,y) = (d^2/dx^2+d^2/dy^2)(-1/2\pi) \log r$$

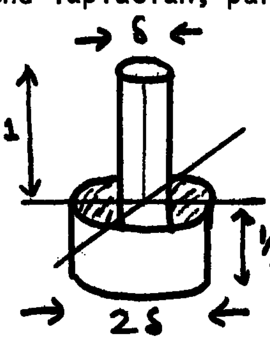
$$d/dx \log r = d/dx\ (1/2) \log(x^2+y^2) = x/(x^2+y^2)$$

$$d^2/dx^2 \log r = -(x^2-y^2)/(x^2+y^2)$$

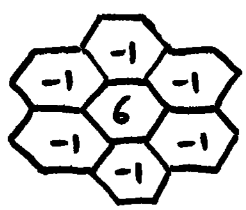
$$d^2/dy^2 \log r = (x^2-y^2)/(x^2+y^2) \quad \text{by symmetry}$$

$$(d^2/dx^2+d^2/dy^2) \log r = 0 \text{ except for } x=y=0$$

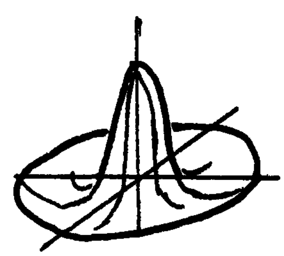
So log r is the function which has the surprising property of having a curvature at each point which is exactly opposite to the curvature at right angles. Next we might be interested in a discreet approximation to the laplacian, particularly a rotationally symmetric one(**B**):



A



B



C

Now since we have all this nice linear theory ala Wiener available we might as well mention that if the image is corrupted by gaussian spatially independent noise we can apply his results to produce a least squares approximation to $g(x,y)$. We then find our convolution functions more spread out than the laplacian and in fact they will contain a central peak surrounded by a larger negative depression (**C**). The only problem is that only some part of the noise in the image satisfies the criterion, a great deal of it not being spatially independent and whats worse, there is no reason to suppose that a least squares approximation to our pulse walls would be at all useful. Anyway, here it is, our anchor point for the spectrum of feature point (or inhomogeneous) finders.

FAST FOURIER TRANSFORM

Once we have bit-reversed the complex array x containing the function to be transformed we proceed as follows, assuming $ln = \log_2 n$.

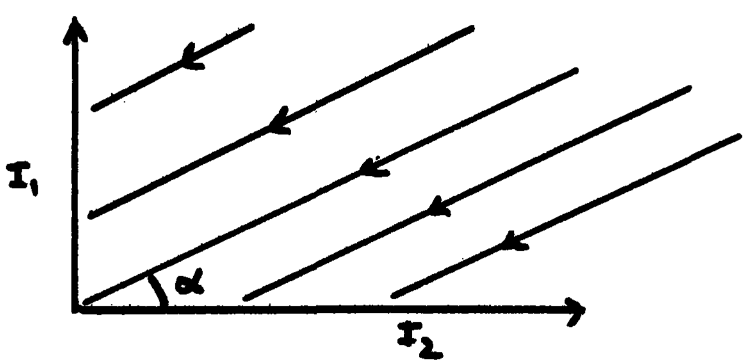
```
n   ← 2↑ln
itn ← n/2
igr ← n/2
iga ← 2
is  ← 1

Do  i = 1 ( 1 ) ln

    Do  ist = 0 ( iga) n-1

        Do  k = ist ( 1 ) ist+is-1 , iwb = 0 ( igr )

        a ← x(k+is)*w(iwb)+x(k)
        b ← x(k+is)*w(iwb+itn)+x(k)

        x(k) ← a
        x(k+is) ← b

        End

    End

    igr ← igr/2
    is ← iga
    iga ← iga*2

End
```

Where $w(a) = e^{\frac{2\pi a i}{n}}$

Note that the arrays x and w are complex valued and dimension n.

CONTRAST IN A RECTANGULAR CORNER:

One of the problems in generating line-drawings from complex scenes is that in addition to the contrast-reduction due to scatter in the imaging device there is also a great reduction in contrast due to mutual illumination. To get a handle on this problem, consider the simple case of two semi-infinite planes meeting at right-angles. The light is incident at an angle α w.r.t. one of the planes. The surface is such that ρ of the incident light is reflected. Clearly for any point on one of the half-planes one half is reflected into empty space, the rest onto the other surface. Light incident at any point is a sum of the light from the source and that reflected from the other plane. If both planes are semi-infinite the intensity on each one will be uniform since a point receives an amount of light from the other plane that does not depend on the position of the point.



$$I_1 = (\rho/2)\, I_2 + a \cos\alpha$$

$$I_2 = (\rho/2)\, I_1 + a \sin\alpha$$

$$I_1 = (\cos\alpha + (\rho/2) \sin\alpha)\, a \,/\, (1 - (\rho/2)^2)$$

$$I_2 = (\sin\alpha + (\rho/2) \cos\alpha)\, a \,/\, (1 - (\rho/2)^2)$$

$$\text{Contrast} = \left|\frac{I_1 - I_2}{I_1 + I_2}\right| = \left|\frac{((\rho/2-1)\cos\alpha - (\rho/2-1)\sin\alpha)}{((\rho/2+1)\cos\alpha + (\rho/2+1)\sin\alpha)}\right| = \frac{2-\rho}{2+\rho}\left|\frac{\cos\alpha - \sin\alpha}{\cos\alpha + \sin\alpha}\right|$$

$$\text{Contrast} = \left|\frac{I_1-I_2}{I_1+I_2}\right| = \frac{2-\rho}{2+\rho} \cdot \left|\tan(\alpha - \pi/4)\right|$$

In the absence of reflection this will be $|\tan(\alpha - \pi/4)|$, so the contrast is reduced by a factor

$$(2-\rho)/(2+\rho)$$

This factor ranges from 1/3 to 1 as ρ ranges from 1 to 0. So for objects that reflect most of the incident light, such as our white cubes this effect is worst, reducing the contrast by a factor 3.

If we consider finite half-planes things get more hairy and the intensity on a given plane is no longer independent of position, falling off as one goes outward from the corner. In the corner itself however the situation is unchanged in the limit. So as far as the contrast across the edge in the image is concerned we can still use the above formula. Note that with finite half-planes a rigorous analysis would require knowledge of the distribution of reflected light with angle which was not needed in the above.

If we consider other angles we find that the problem increases as the angle gets smaller.

Suppose the angle between the two planes is π/k instead of $\pi/2$. Then instead of $(\rho/2)$ we have $(1-1/k)\rho$. The reduction in contrast then is:

$$\frac{1 - (1 - 1/k)\rho}{1 + (1 - 1/k)\rho}$$

And when $\rho = 1$, the worst case, we have a reduction of $1/(2k-1)$.

It is clear that gray blocks are very much better in this respect than white ones. For example if $\rho = .5$ instead of 1.0, the reduction is only 3/5 instead of 1/3 for the contrast.

SCATTER IN OUR IMAGE DISSECTOR (TVC):

A considerable reduction in contrast in our image dissector is caused by scatter of the incident light. This scatter goes undetected when one concerns oneself with the point-spread function because it corresponds to a very low, very wide skirt around the central blob. The size of the central blob is determined by the resolution of the device (or visa versa) and in our case has a half-intensity radius of around .09 mm (in the centre of the field of view). The scatter skirt however extends easily to the edge of the field of view 38 mm away. It is so low that it would go undetected due to dim-cutoff if we are looking at point sources. Only when it is integrated over large areas is its effect noticable. It turns out that about 33 % of the incident light is scattered in this way. This causes a dramatic reduction of contrast.

Several causes can be traced for this phenomenan. The lens contributes some small amount of scatter but the major defects occur because of multiple reflections in the face-plate and reflection from the aperture plate at the end of the drift-tube. It is not known whether any electron optic effects come into this as well. The light enters the face plate and is partially absorbed by the photocathode; some light is however reflected and may bounce repeatetly inside the face-plate. Some fraction of the light also passes right through the photo-cathode and strikes the shiny nickel (!) arperture plate only to be reflected onto the back of the photo-cathode.

These problems could be ameliorated by optically coating the face-plate to avoid multiple reflections or to use a fiber-optic front-plate. The arperture plate clearly ought to be made of some more reasonable material (to avoid the magnetic problems) and should be fairly non-reflective.

(We might expect by the way that the front-plate scatter is worse for larger iris diameter (lower f-stops) because the light will be entering the face-plate from larger angles relative to the optical axis)

As pointed out this phenomena only occurs when we are integrating signals over large areas. To measure the effect then we have to illuminate large areas. One method involves the use of a series of white discs on a black background to be viewed by the image dissector. For each size disc one records the intensity at the centre. This method suffers from the fact that it is hard to find paper surfaces that have a high reflectivity ($>$ 50%) and others having a low reflectivity ($<$ 10%). The observed effect is then considerably lower than expected, in addition the scatter in the lens is included.

A better method is that of removing the lens, using a point source of light (such as a distant lamp reflected in a metal sphere) and using the iris to allow variable diameter circles of light to fall on the photocathode. We observe in this way the integral of this scatter function. Let the point-spread function be rotationally symmetric, f(r).

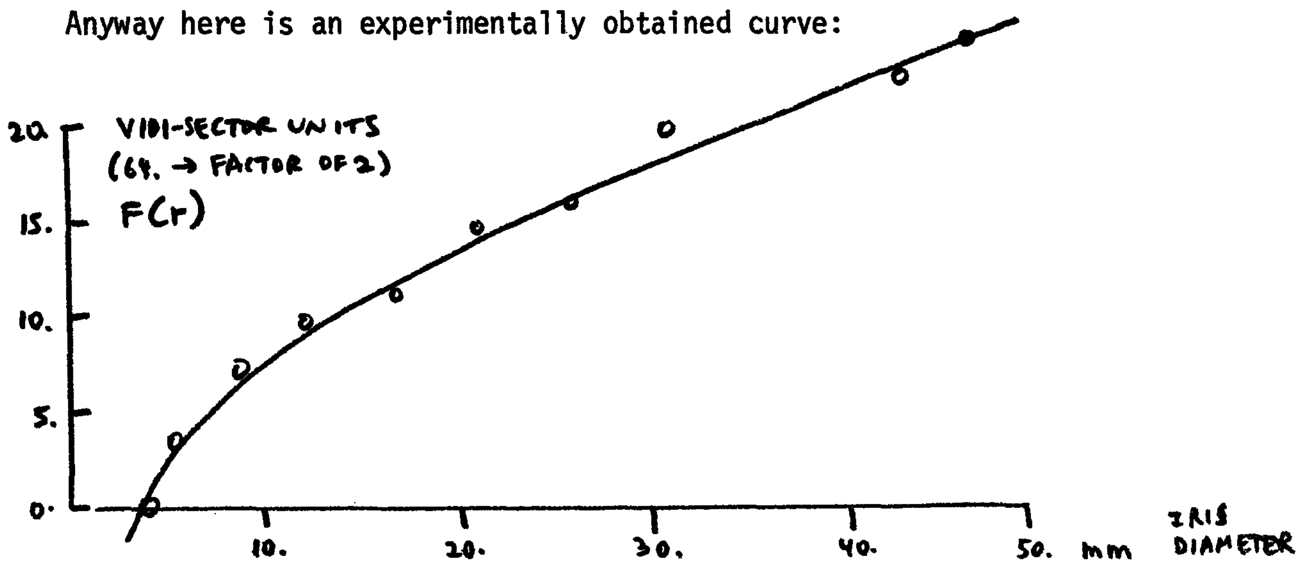
$$F(r) = 2\pi\int_0^r f(t)\, t\, dt$$

If we wish we can differentiate the observed function and get:

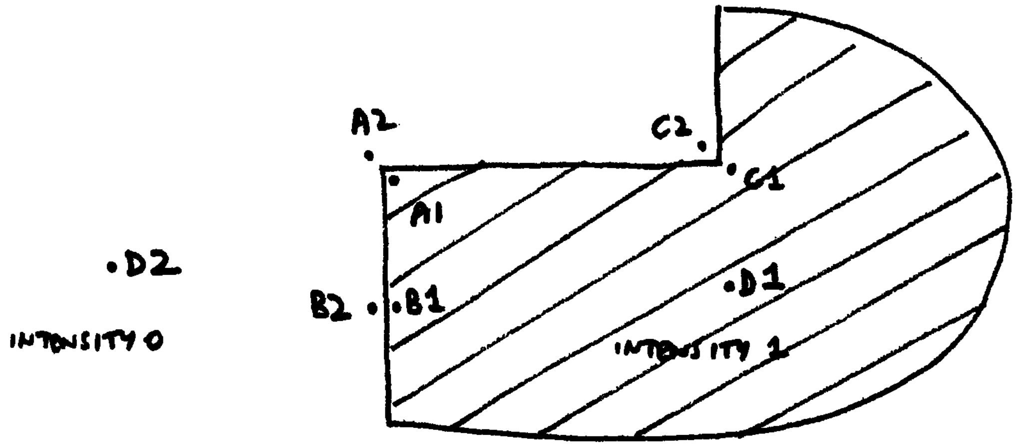
$$2\pi\, f(r)\, r$$

There is some reason to suppose that f(r) can be approximated by $e^{-\sigma r}/r$

Anyway here is an experimentally obtained curve:



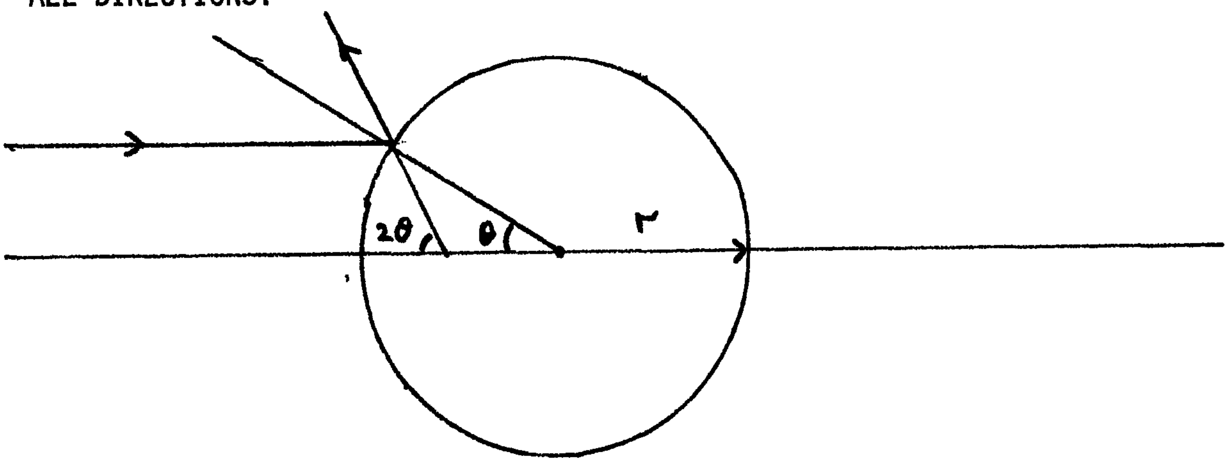
We can use our results to estimate the intensity at various points in an image consisting of large polygonal areas of uniform intensity.



Lets look at the intensity at the points A1, A2, B1, B2, C1, C2 assuming 33 % spill-over:

A1 (90° out of 360° illuminated)	1. - .33 (3/4)	= .75
A2 ( )	0. + .33 (1/4)	= .08
B1 (180° out of 360° illuminated)	1. - .33 (1/2)	= .84
B2 ( )	0. + .33 (1/2)	= .18
C1 (270° out of 360° illuminated)	1. - .33 (1/4)	= .92
C2 ( )	0. + .33 (3/4)	= .25
D1		1.
D2		0.

WHY THE VIRTUAL IMAGE OF A POINT-SOURCE LOOKS EQUALLY BRIGHT FROM ALL DIRECTIONS:



Consider the small surface ring where the light is incident at an angle θ w.r.t to the surface normal.

The incident area is: $$2\pi r^2 \sin\theta \cos\theta \, d\theta = \pi r^2 \sin 2\theta \, d\theta$$

Light falling into this ring is reflected at an angle 2θ w.r.t to the incident ray and with a spread 2 $d\theta$. At the distance R, the light reflected from the ring is spread into an area 2 $R^2 \sin 2\theta \; 2 \, d\theta$. The intensity per unit area at distance R is:

$$I \, (\pi \, r^2 \sin 2\theta \, d\theta)/(2\pi R^2 \sin 2\theta \; 2 \, d\theta) = I \, (r/R)^2 \, / \, 4$$

So its independent of what angle one is looking at it from.

This has implications for reflectivity models of surfaces made of spherical particles. It is also useful in producing point-sources with very small source areas.

(We are assuming both source and observer distant from the sphere). Note that the factor of 4 comes from the fact that the incident area is πr^2, while the light is reflected into an area $4\pi R^2$.

GLOB-TRACKING:

Suppose we have an intensity glob such as a ping-pong ball against a dark background. The object is to track it using the random access camera. Define a two-dimensional pattern of points. The spread and position of this pattern will be servoed using the intensities read.

At each step input the intensities, find their maximum and minimum, IMAX and IMIN. If IMAX is too small, go into search mode, otherwise calculate the following sums

$$\Sigma X_i I_i \qquad \Sigma Y_i I_i \qquad \Sigma I_i$$

Then adjust the position:

$$\overline{X_{n+1}} = \overline{X_n} + \theta_1 \left(\frac{\Sigma X_i I_i}{\Sigma I_i} - \overline{X_n} \right)$$

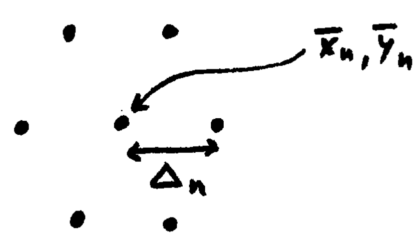
$$\overline{Y_{n+1}} = \overline{Y_n} + \theta_1 \left(\frac{\Sigma Y_i I_i}{\Sigma I_i} - \overline{Y_n} \right)$$

Then adjust the size of the pattern:

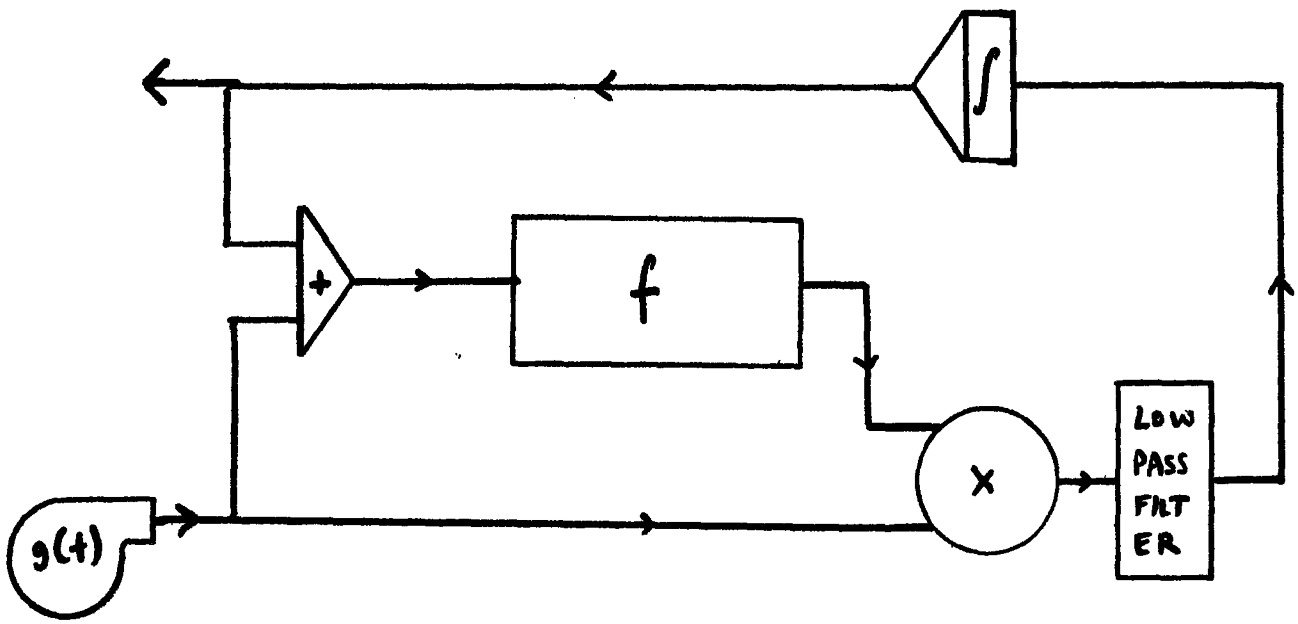
$$\Delta_{n+1} = \Delta_n \left(1 + \theta_2 \left(\frac{\text{IMAX}' - \text{IMIN}'}{\text{IMAX} - \text{IMIN}} - 1 \right) \right)$$

Where (IMAX' - IMIN') is the desired state of intensity range.

Usually $\theta_1 > \theta_2$ eg $\theta_1 = 1/2$ $\theta_2 = 1/8$



A RELATED ANALOG TYPE GLOB-TRACKER

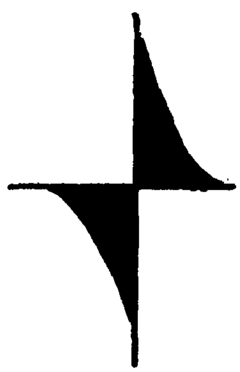


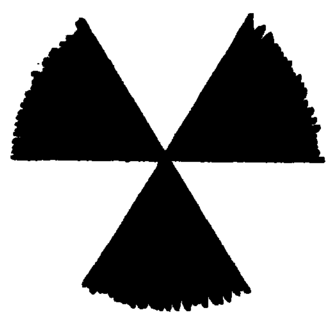
Here g(t) is some test function like cos (wt) for example and f is the external function such as intensity. An interesting case is obtained if we combine two of these circuits, one for x and one for y coordinates in an image dissector camera. We then have a star-tracker. The two g(t)'s will need to be "orthogonal" then, cos(wt) and sin(wt) for example.

A similar circuit or equivalent program can be used for light-pen tracking. Interesting variations address the question of whether the low pass filter can be eliminated or replaced by some other device and whether g(t) can be removed or "self-generated". In other words one aims at a system that is self-contained and samples the image in a way dependend on what is in the image rather than some fixed predetermined pattern.

THE SURVEYORS MARK AND FRIENDS:

To track an object using the image dissector camera it is desirable to have to read the intensity at as few steps as possible at each time interval. The pattern to look at must also be designed for three conflicting requirments: ease of acquisition, ease of tracking in fast motion and accuracy of locating when stationary. The first two cause the object to be fairly large, the last requires that some point on it be well defined. The program should have no difficulty in processing the intensities read and should be fairly independent of distance and orientation of the pattern. A radially symmetric pattern with black and white areas seems suitable. In particular one consisting of a number of intersecting lines with alternate segments filled in black and white seems a winner. The one used by surveyors uses two lines, our robotics calibration programs use three-line patterns.





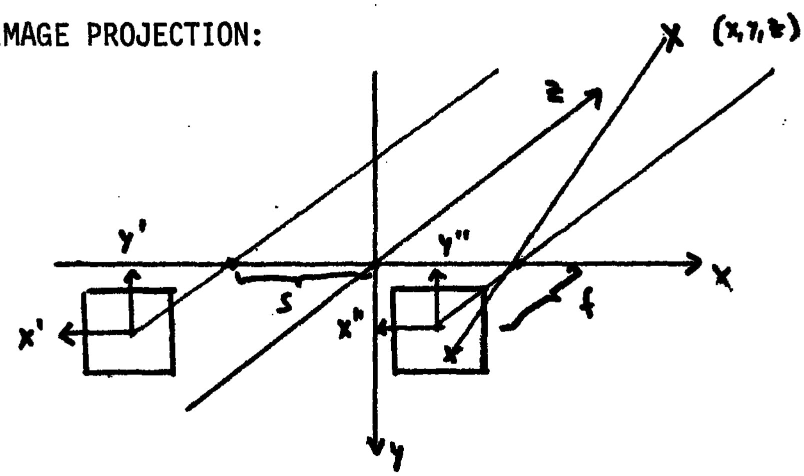
The image processing is simple. One reads the intensity at a number of points on the circumference of a circle, finds the maximum and minimum and sets up hysteresis thresholds. The lines are detected at the points where the intensity crosses both thresholds in sequence. The six points define three lines. The centre is then estimated to be near the point of minimum sum of squares of perpendicular distances to the three lines. Image motion between succesive scans can be almost the radius of the pattern, while its centre can be located extremely accuratly by shrinking the sampling circle.

OBJECT ROTATION MATRIX:

Consider an object rotated first about the x-axis (pitch, p), then about the y-axis (yaw, y) and finally about the z-axis (roll, r). We are interested in the corresponding transformation matrix:

$$\begin{vmatrix} \cos r \cos y & (\cos r \sin y \sin p - \sin r \cos p) & (\cos r \sin y \cos p + \sin r \sin p) \\ \sin r \cos y & (\sin r \sin y \sin p + \cos r \cos p) & (\sin r \sin y \cos p - \cos r \sin p) \\ -\sin y & \cos y \sin p & \cos y \cos p \end{vmatrix}$$

STEREO IMAGE PROJECTION:



Left eye: $x' = (x+s)f/z$ $y' = (y)f/z$

Right eye: $x'' = (x-s)f/z$ $y'' = (y)f/z$

Projection of point (x,y,z)

f is the distance the resulting images are to be viewed from.
2s is the eye separation.

EXPOSURE GUIDE FOR OUR DEC 340 DISPLAY

I	$(1+I)^{3/2}$
0	1
1	2.8
2	5
3	8
4	11
5	15
6	18
7	23

$$f = \frac{k}{r_1+r_2}\sqrt{\frac{A\ N\ P}{F}}\ (1+I)^{3/2}$$

f - f-number indicated on lens.

k - empirically found to be about 1/125 (gives rise to density of about 2 in negative; i.e. almost overexposed).

r_1 - Half-intensity radius of spot on DEC 340, varies somewhat with I. use .5 mm unless you have good reason to suspect other value.

r_2 - Half-intensity radius of blur in camera projected back onto display surface - varies with lens and film used. use .5 mm unless you have good reason to suspect other value.

r_3 - Spacing of points in image you are displaying. Use ∞ if all the points can be resolved in the image. Use .25 mm * 2^s for vectors, increments and characters of scale s.

s - scale send to DEC 340, 0-3.

A -- ASA rating of film.

For polaroid B/W: 3000
For 35mm TRI-X : 300
For 16mm TRI-X : 200

N - Argument to .NDIS ; i.e. number of times points are displayed.

P - Packing factor.

1 for resolved points.

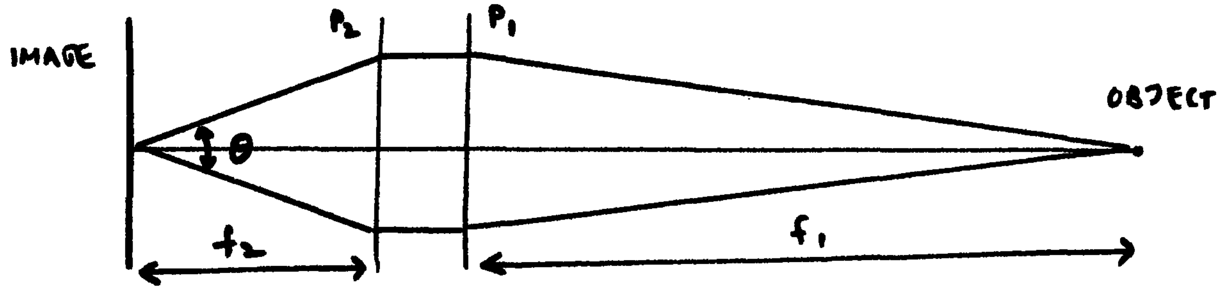
$\max(1, \frac{r_1+r_2}{r_3})$ for one-dimensional sets of points (vectors,increments, characters)

$\max(1, (\frac{r_1+r_2}{r_3})^2)$ for two-dimensional sets of points (rasters).

F - Filter factor. 1 for no filter, 2 for Wratten 15 (afterglow only), 8 for Wratten 47 (flash only).

I - Intensity parameter send to scope. If varying intensities are to be recorded, use I=5 - highlights will be slightly overexposed but the dark-areas will not be completely under-exposed. 0-7.

SOME LENS FORMULAE:



Let P_1 be the front principal plane, be P_2 the rear principal plane. Let f be the focal length and the media on the two sides of the lens be the same. Let f_1 be the object-lens distance and f_2 the lens-image distance.

The de-magnification of the image is then:

$$M = \frac{f_1}{f_2}$$

We know that: $\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f}$ i.e. $(f_1 - f)(f_2 - f) = f^2$

$$f_1 = (1 + M)f$$

$$f_2 = (1 + 1/M)f$$

Let d be the object to image distance (ignoring thick lens effect):

$$d = f_1 + f_2 = f\,(M + 2 + 1/M) = f\,\frac{(1+M)^2}{M}$$

Let $x = \left(\frac{d}{2f} - 1\right)$ then $M^2 - 2\,x\,M + 1 = 0$

$$x = \frac{M^2 + 1}{2M} = \frac{1}{2}(M + 1/M)$$

$$M = (x-1) \pm \sqrt{x^2-1}$$

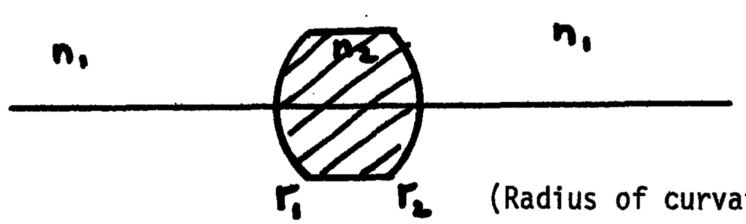
$$\frac{dd}{dM} = f\left(1 - \frac{1}{M^2}\right) = f\left(\frac{M^2 - 1}{M^2}\right) \qquad \frac{df_1}{dM} = f \qquad \frac{df_2}{dM} = -f/M^2$$

$$\frac{df_1}{dd} = \frac{M^2}{M^2 - 1} \qquad \frac{df_2}{dd} = \frac{-1}{M^2 - 1} \qquad \frac{df_1}{df_2} = -M^2$$

These formulae are useful for calculating focusing accuracy for example.

Numerical arperture is defined as $n\sin(\theta/2)$, the f-stop as $\dfrac{1}{2\sin(\theta/2)}$.

Where θ is the angle subtended by the lens at the centre of the image. The intensity at the image is proportional to $1/(\text{f-stop})^2$.



(Radius of curvature)

Then we have the lens-makers equation:

$$\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f} = (n_2 - n_1)\left(\frac{1}{r_1} - \frac{1}{r_2}\right)$$

Optimal pin-hole radius (for diffraction to equal hole spread):

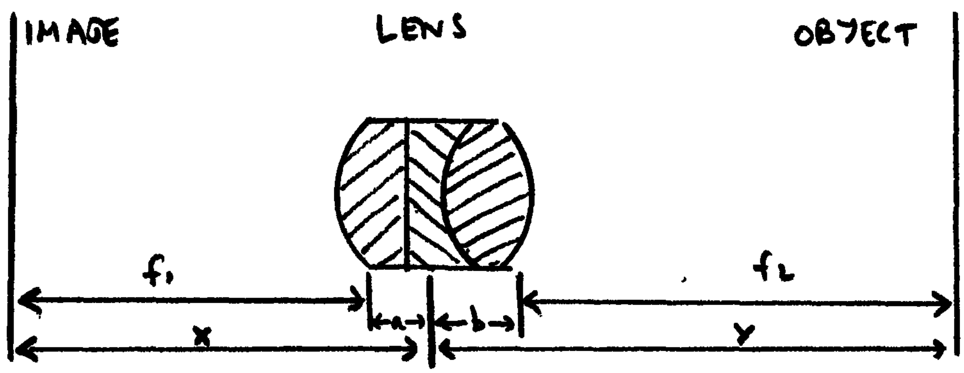
$r = \sqrt{d\lambda}$ where d is the hole-image distance, λ the wavelength

Airy radius: $2\,(3.8317/(2\pi))\,\lambda\,(\text{f-stop}) = 1.22\,\lambda\,(\text{f-stop})$

(Since 3.8317 is the first zero of $J_1(x)/x$)

FOCAL LENGTH CALIBRATION:

For accurate camera models one needs good measurements of the focal length and the position of the principal planes.



Now $x = f_1 + a$, $y = f_2 + b$ and $(1/f_1) + (1/f_2) = (1/f)$

We measure several combinations of x_i and y_i and attempt to find a, b and most important, f. We can assume that a and b are relatively small relative to f and that f is known approximately. We clearly require three such sets of measurements and could use least-squares methods if we had more. Unfortunatly the equations are non-linear. We can make them into polynomials in a, b and f however:

$$1/(x_i-a) + 1/(y_i-b) = 1/f$$

$$((x_i+y_i) - (a+b))\, f - (x_i-a)(y_i-b) = 0$$

$$-(ab+bf+fa) + (x_i(f+b) + y_i(f+a)) - x_i y_i = 0$$

We could solve this set of second order polynomials in 3 variables in a number of ways. Perhaps the easiest is multi-dimensional Newton-Raphson iteration. We consider this last expression as a function F of the parameters a, b and f and are aiming for F(a,b,f) = 0. For this we require the derivatives:

$$dF/da = y_i - b - f,\quad dF/db = x_i - a - f,\quad dF/df = (x_i+y_i)-(a+b)$$

We can also use guessing or a least squares method. If we can select the x_i and y_i we can also simplify the problem.

Special Case: If we can choose x_i and y_i we might try the following:

$$y_1 = \infty \qquad x_1 = f+a \qquad a = x_1 - f$$

$$x_2 = \infty \qquad y_2 = f+b \qquad b = y_2 - f$$

We need one more measurement:

$$1/(x_3 - x_1 + f) + 1/(y_3 - y_2 + f) = 1/f$$

Let $x_3 - x_1 = x'$, $y_3 - y_2 = y'$

$$(y'+f + x'+f)\, f - (x' + f)(y' + f) = 0$$

$$(x'+y')f + 2f^2 - x'y' - (x'+y')f - f^2 = 0$$

$$-x'y' + f^2 = 0 \qquad f = \sqrt{x'y'}$$

$$f = \sqrt{(x_3 - x_1)(y_3 - y_2)}$$

For accuracy we want both differences large, this implies that we want x_3 about the same magnitude as y_3.

DETERMINING THE TRANSFORM FROM ARM TO EYE SPACE:

Being a rotation and translation we expect:

$$\begin{vmatrix} x_v \\ y_v \\ z_v \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} a_{14} \\ a_{24} \\ a_{34} \end{vmatrix}$$

And the matrix ought to be orthogonal (i.e. $A^T A = I$). The coordinates with a-subscripts are arm coordinates, those with a v-subscript are eye coordinates. By allowing the matrix to be non-orthogonal we can absorb some of the distortions and non-linearities. In any case forcing it to be orthogonal introduces a non-linear constraint that messes up the mathematics ! We then have to use iterative methods well-known in the art of reducing aerial photographs.

Next we have to consider the projection into the image plane:

$$u = (x_v/z_v)\alpha + u_0$$
$$v = (y_v/z_v)\beta + v_0$$

α and β are normally the same more or less and depend on the focal length and the translation from image coordinates to deflection units. u_0 and v_0 are zero if we choose the image origin on the optical axis which may at times be convenient. It is not hard to show that α, β, u_0 and v_0 can be absorbed into our first transform and we can consider the simpler case:

$$u = x_v/z_v \qquad \text{and} \qquad v = y_v/z_v$$

This does make the matrix non-orthogonal however. Clearly multiplying all the a_{ij}'s by any factor causes no change in the image coordinates and we can therefore choose a fixed value for one of them, say $a_{34} = 1$. We then have the problem of determining the values of the other 11 terms.

We need at least 11 equations then and prefer ably more so as to allow a least squares solution. One method of determining the transformation matrix depends on moving the arm into n known positions and recording the corresponding x_{ai}, y_{ai}, z_{ai} and image coordinates u_i and v_i. It is convenient to track a special object held in the hand as it moves around rather than to blindly move the hand and try and locate it in the image.

For each such measurement we get 2 equations:

$$x_v - z_v\, u_i = 0 \qquad \text{and} \qquad y_v - z_v\, v_i = 0$$

$$a_{11}x_{ai}+a_{12}y_{ai}+a_{13}z_{ai}+a_{14} - a_{31}u_i x_{ai} - a_{32}u_i y_{ai} - a_{33}u_i z_{ai} - u_i a_{34} = 0$$

$$a_{21}x_{ai}+a_{22}y_{ai}+a_{23}z_{ai}+a_{24} - a_{31}v_i x_{ai} - a_{32}v_i y_{ai} - a_{33}v_i z_{ai} - v_i a_{34} = 0$$

For n such measurments we get 2n such equations which can be separated into two groups and written in matrix form as follows:

$$\begin{vmatrix}
x_{a1} & y_{a1} & z_{a1} & 1 & 0 & 0 & 0 & 0 & -u_1x_{a1} & -u_1y_{a1} & -u_1z_{a1} & -u_1 \\
x_{a2} & y_{a2} & z_{a2} & 1 & 0 & 0 & 0 & 0 & -u_2x_{a2} & -u_2y_{a2} & -u_2z_{a2} & -u_2 \\
 & & & & & & & & & & & \\
x_{an} & y_{an} & z_{an} & 1 & 0 & 0 & 0 & 0 & -u_nx_{an} & -u_ny_{an} & -u_nz_{an} & -u_n \\
0 & 0 & 0 & 0 & x_{a1} & y_{a1} & z_{a1} & 1 & -v_1x_{a1} & -v_1y_{a1} & -v_1z_{a1} & -v_1 \\
0 & 0 & 0 & 0 & x_{a2} & y_{a2} & z_{a2} & 1 & -v_2x_{a2} & -v_2y_{a2} & -v_2z_{a2} & -v_2 \\
 & & & & & & & & & & & \\
0 & 0 & 0 & 0 & x_{an} & y_{an} & z_{an} & 1 & -v_nx_{an} & -v_ny_{an} & -v_nz_{an} & -v_n
\end{vmatrix} * \begin{vmatrix} a_{11} \\ a_{12} \\ \\ \\ \\ \\ \\ a_{34} \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ \\ \\ \\ \\ \\ 0 \end{vmatrix}$$

Setting a_{34} to 1 and taking the resulting constant terms $(u_1 \ldots u_n, v_1 \ldots v_n)$ to the right hand side we obtain 2n equations in 11 unknowns. We can make do with 5 1/2 experimental measurements or attempt a least squares solution for $n \geq 6$ points. Not more than 3 points should be in any one plane in the first instance to avoid degeneracy. It is convenient to use the points at the tips of an octahedron.

Notes: 1. A slightly different formulation leads to 18 equations in 18 unknowns, the same results are obtained. (This corresponds to the homogeneous representation).

2. If we had assumed orthogonality we would have introduced 3 more constraints and needed only 8 equations, that is 4 experimental points which could conveniently be the corners of a tetrahedron.

RELATION BETWEEN THE SIMPLIFIED AND THE REAL IMAGE COORDINATES:

Given:
$$\begin{vmatrix} x_v \\ y_v \\ z_v \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} a_{14} \\ a_{24} \\ a_{34} \end{vmatrix}$$

and $\quad u = x_v/z_v \quad$ and $\quad v = y_v/z_v$

Where x_a, y_a, z_a are coordinates relative to the arm coordinate system. x_v, y_v, z_v are coordinates relative to the eye and u and v are image coordinates in the simplified system.

Now we introduce the real image coordinates:

$$u' = u\,\alpha + u_0 \quad \text{and} \quad v' = v\,\beta + v_0$$

$$\alpha x_v - (u'-u_0)z_v = 0 \quad \text{and} \quad \beta y_v - (v'-v_0) = 0$$

$$(\alpha a_{11}+u_0a_{31})x_a+(\alpha a_{12}+u_0a_{32})y_a+(\alpha a_{13}+u_0a_{33})z_a+(\alpha a_{14}+u_0a_{34})$$
$$-u'(a_{31}x_a+a_{32}y_a+a_{33}z_a+a_{34})=0$$

$$(\beta a_{21}+v_0a_{31})x_a+(\beta a_{22}+v_0a_{32})y_a+(\beta a_{23}+v_0a_{33})z_a+(\beta a_{24}+v_0a_{34})$$
$$-v'(a_{31}x_a+a_{32}y_a+a_{33}z_a+a_{34})=0$$

So finally:

$$\begin{vmatrix} x'_v \\ y'_v \\ z'_v \end{vmatrix} = \begin{vmatrix} (\alpha a_{11}+u_0a_{31}) & (\alpha a_{12}+u_0a_{32}) & (\alpha a_{13}+u_0a_{33}) \\ (\beta a_{21}+v_0a_{31}) & (\beta a_{22}+v_0a_{32}) & (\beta a_{23}+v_0a_{33}) \\ (\quad a_{31} \quad) & (\quad a_{32} \quad) & (\quad a_{33} \quad) \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} (\alpha a_{14}+u_0a_{34}) \\ (\beta a_{24}+v_0a_{34}) \\ (\quad a_{34} \quad) \end{vmatrix}$$

A VERTICAL PREDICATE FOR IMAGE LINES:

In a perspective transformation of the world a set of parallel lines will project into a bundle of lines passing through one point, the vanishing point x_f, y_f .

If we simply assume that vertical means parallel to the arm's z-axis:

Then as $z_a \to \infty$, $x_v \to a_{13}z_a$, $y_v \to a_{23}z_a$, $z_v \to a_{33}z_a$

And so $\quad x_f = a_{13}/a_{33} \quad$ and $\quad y_f = a_{23}/a_{33}$

In practice vertical means perpendicular to the table. Suppose the table equation is given by:

$$p_1x + p_2y + p_3z + p_4 = 0$$

Then let $x_a = \alpha p_1$, $y_a = \alpha p_2$, $z_a = \alpha p_3$ and let $\alpha \to \infty$.

$$x_v \to \alpha(a_{11}p_1 + a_{12}p_2 + a_{13}p_3)$$
$$y_v \to \alpha(a_{21}p_1 + a_{22}p_2 + a_{23}p_3)$$
$$z_v \to \alpha(a_{31}p_1 + a_{32}p_2 + a_{33}p_3)$$

$$x_f = \frac{a_{11}p_1 + a_{12}p_2 + a_{13}p_3}{a_{31}p_1 + a_{32}p_2 + a_{33}p_3}$$

$$y_f = \frac{a_{21}p_1 + a_{22}p_2 + a_{23}p_3}{a_{31}p_1 + a_{32}p_2 + a_{33}p_3}$$

To test if a line is vertical or near vertical we calculate the angle it makes with the line connecting it to the vanishing point:

$$x_{12} = x_1 - x_2 , \quad x_{1f} = x_1 - x_f , \quad y_{12} = y_1 - y_2 , \quad y_{1f} = y_1 - y_f$$

$$(\sin \theta)^2 = (x_{1f}y_{12} - x_{12}y_{1f})^2 / (x_{1f}^2 + y_{1f}^2)(x_{12}^2 + y_{12}^2)$$

GOING FROM IMAGE COORDINATES TO ARM SPACE COORDINATES:

Clearly we need some extra information to make up for the lack of one dimension. But first lets look at what we have:

$$u = x_v/z_v \quad \text{and} \quad v = y_v/z_v$$

$$x_v - u\, z_v = 0 \quad \text{and} \quad y_v - v\, z_v = 0$$

$$(a_{11}-ua_{31})x_a+(a_{12}-ua_{32})y_a+(a_{13}-ua_{33})z_a=(a_{14}-ua_{34})$$

$$(a_{21}-va_{31})x_a+(a_{22}-va_{32})y_a+(a_{23}-va_{33})z_a=(a_{24}-va_{34})$$

We need a third equation in x_a, y_a and z_a to be able to solve. We could for example be given any one of these three coordinates. More likely is the case where we have some relation to the table. Let the equation of the table be given by:

$$p_1x + p_2y + p_3z + p_4 = 0$$

If the point is on the table we can simple use this equation.

If the point is in the same plane parallel to the table as some other known point x_1, y_1, z_1 then we use the equation:

$$p_1x_a+ p_2y_a+ p_3z_a = p_1x_1 + p_2y_1 + p_3z_1$$

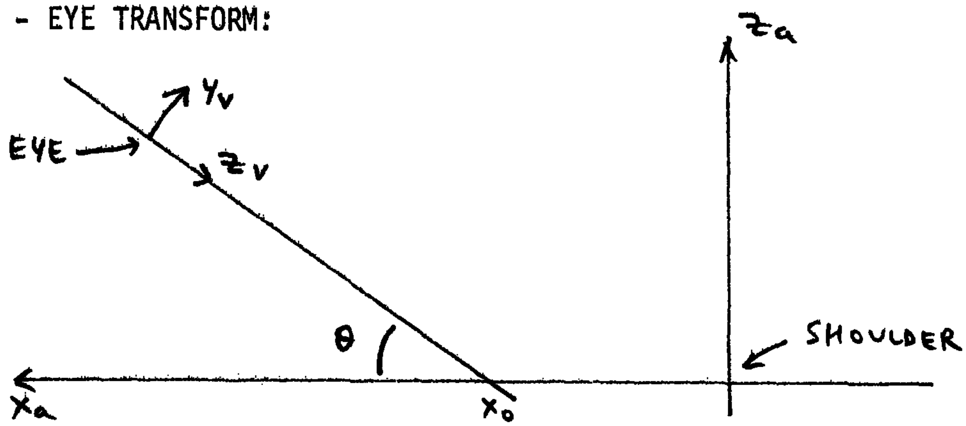
If the point is directly above (along a line normal to the table) some other known point x_2, y_2, z_2 then we have:

$$(x_a-x_2)\, p_3 - (z_a-z_2)\, p_1 = 0$$

$$(y_a - y_2)\, p_3 - (z_a-z_2)\, p_2 = 0$$

We only need one of these and will use the first since it comes out more accurate with the eye-arm geometries we use.

TYPICAL ARM - EYE TRANSFORM:



Suppose we have the above simple geometry. Then:

$$\begin{vmatrix} x_v \\ y_v \\ z_v \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \\ -\cos\theta & 0 & -\sin\theta \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} 0 \\ \sin\theta\, x_0 \\ \cos\theta\, x_0 + \ell \end{vmatrix}$$

Now we change to real image coordinates:

$$\begin{vmatrix} x_v' \\ y_v' \\ z_v' \end{vmatrix} = \begin{vmatrix} (\alpha a_{11}+u_0 a_{31}) & (\alpha a_{12}+u_0 a_{32}) & (\alpha a_{13}+u_0 a_{33}) \\ (\beta a_{12}+v_0 a31) & (\beta a_{22}+v_0 a_{32}) & (\beta a_{23}+v_0 a_{33}) \\ (\quad a_{31} \quad) & (\quad a_{32} \quad) & (\quad a_{33} \quad) \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} (\alpha a_{14}+u_0 a_{34}) \\ (\beta a_{24}+v_0 a_{34}) \\ (\quad a_{34} \quad) \end{vmatrix}$$

So we get:

$$\begin{vmatrix} x_v' \\ y_v' \\ z_v' \end{vmatrix} = \begin{vmatrix} -u_0\cos\theta & \alpha & -u_0\sin\theta \\ -\beta\sin\theta - v_0\cos\theta & 0 & \beta\cos\theta - v_0\sin\theta \\ -\cos\theta & 0 & -\sin\theta \end{vmatrix} \begin{vmatrix} x_a \\ y_a \\ z_a \end{vmatrix} + \begin{vmatrix} u_0(\cos\theta\, x_0+\ell) \\ \beta\sin\theta\, x_0+v_0(\cos\theta\, x_0+ \\ \cos\theta\, x_0+\ell \end{vmatrix}$$

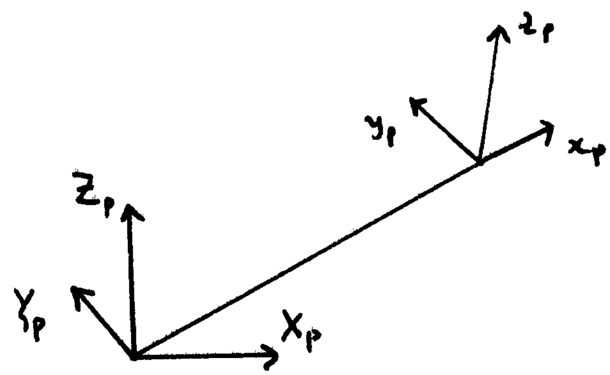
Now suppose $\theta = 30^o$, $\cos\theta = .86...$, $\sin\theta = .5$, $u_0=v_0=512$. (Center of coordinates for the image dissector on a scale of 0 - 1024.)
With a lens of 10" focal length we find that α =3150 units/radian.
With a lens of 6.5 " focal length α =2000 units/radian.
(Assuming about 12.5 units per mm on the photocathode)
Next suppose x_0 = 30.0", ℓ = 50.0" and use of the 6.5" lens.

$$\begin{vmatrix} -440. & 2000. & -256. \\ -1440. & 0. & 1464. \\ -.86 & 0. & -.5 \end{vmatrix} \qquad \begin{vmatrix} 39800. \\ 69800. \\ 76. \end{vmatrix}$$

Next we normalise by setting a_{34} = 1:

$$\begin{vmatrix} -5.8 & 26.3 & -3.38 \\ -19.0 & 0. & 19.3 \\ -.0113 & 0. & -.0066 \end{vmatrix} \qquad \begin{vmatrix} 525. \\ 918. \\ 1. \end{vmatrix}$$

ABSOLUTE ORIENTATION:



$$\begin{vmatrix} X_p \\ Y_p \\ Z_p \end{vmatrix} = R \begin{vmatrix} \alpha\, x_p \\ \beta\, y_p \\ \gamma\, z_p \end{vmatrix} + \begin{vmatrix} x_o \\ y_o \\ z_o \end{vmatrix}$$

Where R is an orthogonal rotation matrix. α, β, γ are scale factors, often equal to one another. x_o, y_o, z_o is a displacment vector. There is one of these equations for each point in the object.

Let $X_{ij} = X_i - X_j$ and $x_{ij} = x_i - x_j$, then:

$$\begin{vmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{vmatrix} = R \begin{vmatrix} \alpha\, x_{ij} \\ \beta\, y_{ij} \\ \gamma\, z_{ij} \end{vmatrix}$$

Multiplying this equation by its transpose we get:

$$\left[X_{ij}\ Y_{ij}\ Zij \right] \begin{vmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{vmatrix} = \left[\alpha x_{ij}\ \beta y_{ij}\ \gamma z_{ij} \right] R^T R \begin{vmatrix} \alpha x_{ij} \\ \beta y_{ij} \\ \gamma z_{ij} \end{vmatrix}$$

Now noting that $R^T R = I$ we get:

$$(X_{ij}^2 + Y_{ij}^2 + Z_{ij}^2) = (\alpha^2 x_{ij}^2 + \beta^2 y_{ij}^2 + \gamma^2 z_{ij}^2)$$

Now suppose we are given four points in each coordinate system:

$$\begin{vmatrix} (X_{12}^2 + Y_{12}^2 + Z_{12}^2) \\ (X_{23}^2 + Y_{23}^2 + Z_{23}^2) \\ (X_{34}^2 + Y_{34}^2 + Z_{34}^2) \end{vmatrix} = \begin{vmatrix} x_{12}^2 & y_{12}^2 & z_{12}^2 \\ x_{23}^2 & y_{23}^2 & z_{23}^2 \\ x_{34}^2 & y_{34}^2 & z_{34}^2 \end{vmatrix} \begin{vmatrix} \alpha^2 \\ \beta^2 \\ \gamma^2 \end{vmatrix}$$

It is now easy to solve for α, β, γ . Let $x'_{ij} = \alpha x_{ij}$ and so on:

$$\begin{vmatrix} X_{12} \\ X_{23} \\ X_{34} \end{vmatrix} = \begin{vmatrix} x'_{12} & y'_{12} & z'_{12} \\ x'_{23} & y'_{23} & z'_{23} \\ x'_{34} & y'_{34} & z'_{34} \end{vmatrix} \begin{vmatrix} r_{11} \\ r_{12} \\ r_{13} \end{vmatrix}$$

Let X be the vector $[X_{12}\ X_{23}\ X_{34}]^T$ and similarly for Y and Z.

Let x' be the vector $[x'_{12}\ \ x'_{23}\ \ x'_{34}]^T$ and similarly for y' and z'.

Combinig three equations like the above:

$$(X\ Y\ Z) = (x'\ y'\ z')\ R^T$$

$$R^T = (x'\ y'\ z')^{-1}\ (X\ Y\ Z)$$

$$= \begin{vmatrix} x'_{12} & y'_{12} & z'_{12} \\ x'_{23} & y'_{23} & z'_{23} \\ x'_{34} & y'_{34} & z'_{34} \end{vmatrix}^{-1} \begin{vmatrix} X_{12} & Y_{12} & Z_{12} \\ X_{23} & Y_{23} & Z_{23} \\ X_{34} & Y_{34} & Z_{34} \end{vmatrix}$$

Due to measurment inaccuracies R determined this way may not be orthogonal, one can if one whishes adjust it iteratively using Newton-Raphson:

$$R_{n+1} = R_n - .5 \; (R_n^T R - I)$$

or $$R_{n+1} = .5 \; ((R_n^T)^{-1} + R_n)$$

Fin ally we have to find the displacment vector:

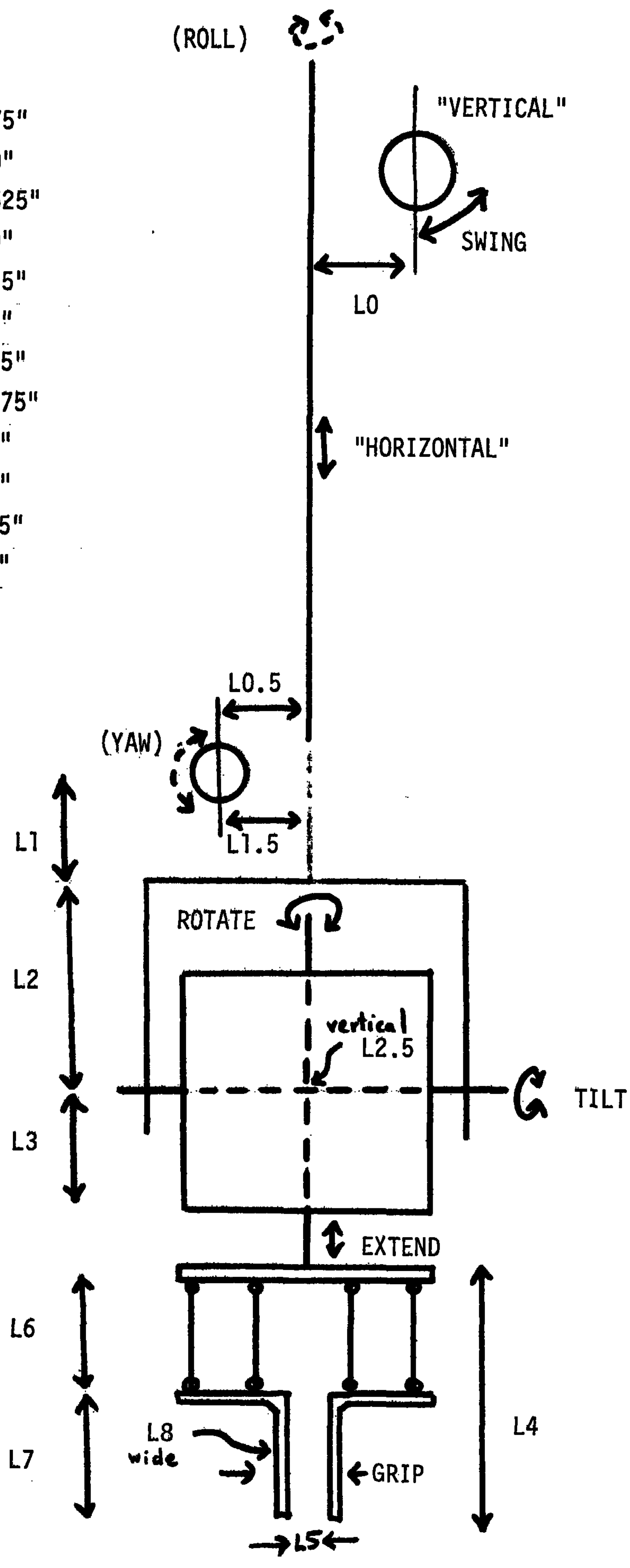
$$\begin{vmatrix} x_o \\ y_o \\ z_o \end{vmatrix} = \begin{vmatrix} X_p \\ Y_p \\ Z_p \end{vmatrix} - R \begin{vmatrix} \alpha \; x_p \\ \beta \; y_p \\ \gamma \; z_p \end{vmatrix}$$

We can get four estimates from this which we can average if neccessary.

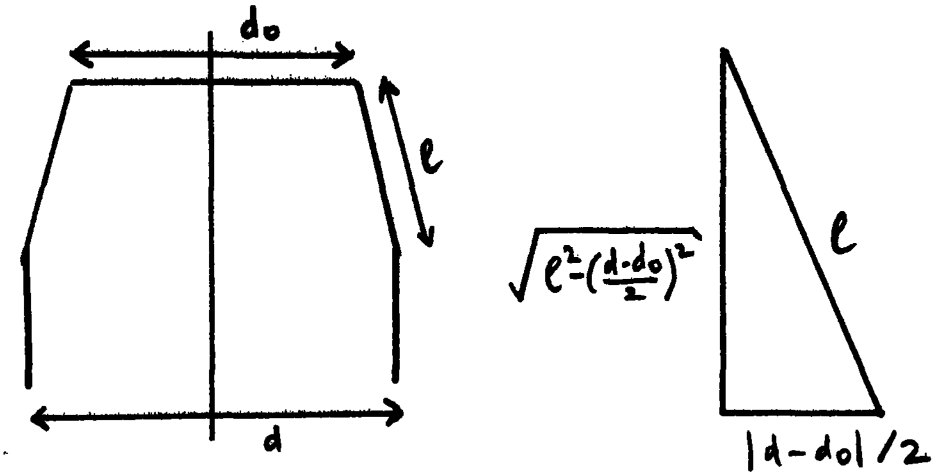
If $\alpha = \beta = \gamma$ we can get away with only three points.

GEOMETRY OF THE AMF-VERSTRAN ARM WITH THE ALLES HAND:

L0	2.75"
L0.5	1.0"
L1	3.625"
L1.5	1.0"
L2	10.5"
L2.5	.75"
L3	4.75"
L4	6.375"
L5	.25"
L6	2.0"
L7	2.75"
L8	.56"



COMPENSATION FOR GRIP MOTION:



The geometry of the grippers is equivalent to the above. We then find a motion along the axis of the grippers w.r.t. the most extended:

$$\ell \left(1 - \sqrt{1 - \left(\frac{d - d_0}{2\ell}\right)^2} \right)$$

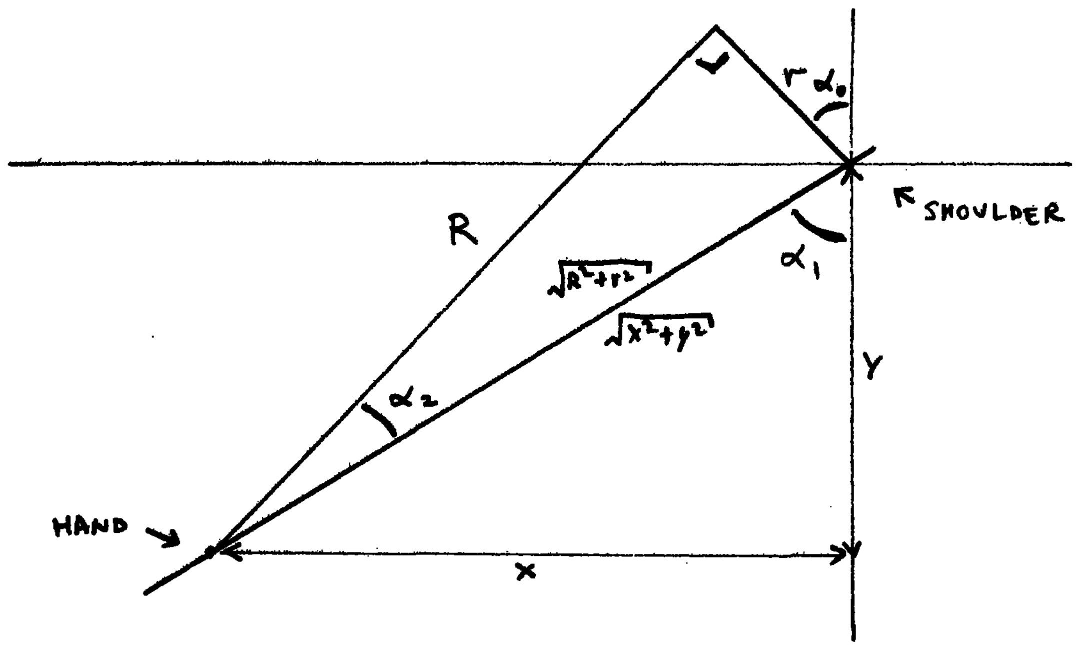
COMPENSATION FOR TILT MOTION:

When the tilt-axis is inclined θ w.r.t vertical one can adjust the horizontal extend by sin θ * hand-extend and the vertical motion by cos θ * hand-extend.

On the whole the arm geometry is very simple and allows direct determination of joint angles and extensions given a desired hand position and orientation, keeping in mind that one only has 5 degrees of freedom.

CONVERSION FROM RECTANGULAR COORDINATES TO PSEUDO-POLAR:

The AMF arm has an offset in its otherwise extremely simple geometry:



Given x, y we need to find R and α_0.

$$R = \sqrt{x^2 + y^2 - r^2} \qquad x^2 + y^2 > r^2$$

$$\tan \alpha_1 = x/y \qquad \tan \alpha_2 = r/R$$

$$\tan \alpha_0 = 1/\tan(\alpha_1 - \alpha_2) = \frac{yR + xr}{xR - yr} = \frac{xy + rR}{x^2 - r^2} = \frac{xy + rR}{R^2 - y^2}$$

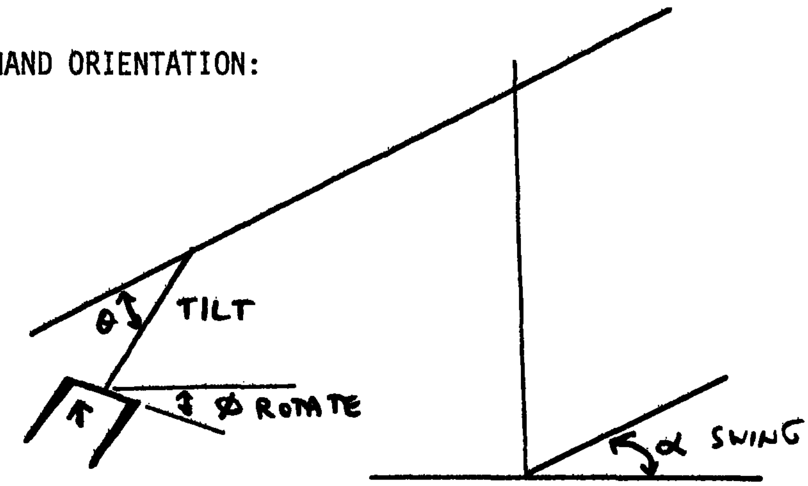
Here we cannot avoid the use of arctan because we actually need the angle. Note that for the AMF arm r = 2.75" .

We used the fact that $x^2 - r^2 = R^2 - y^2$

And that tan(a-b) = (tan a - tan b) / (1 + tan a tan b)

R is the "horizontal" extend and α_0 is the "swing".

MAINTAINING A CONSTANT HAND ORIENTATION:



Let the normal unit vector to the plane containing the two fingers be (a,b,c)

$$(a\ b\ c)=\begin{pmatrix}\cos\alpha & \sin\alpha & 0\\ -\sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\phi & \sin\phi\\ 0 & -\sin\phi & \cos\phi\end{pmatrix}\begin{vmatrix}0\\0\\1\end{vmatrix}$$

$$=\begin{vmatrix}\cos\alpha\sin\theta\cos\phi+\sin\alpha\sin\phi\\ -\sin\alpha\sin\theta\cos\phi+\cos\alpha\sin\phi\\ \cos\theta\cos\phi\end{vmatrix}$$

So given one constraint we can solve for α,θ,ϕ keeping in mind that $a^2+b^2+c^2=1$

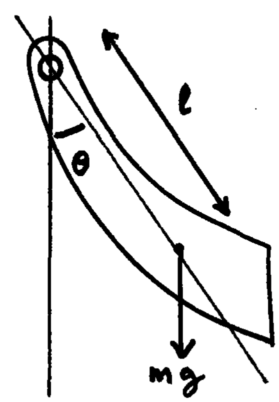
Most commonly we would be given the swing, α, then:

$$\sin\phi=(a\sin\alpha+b\cos\alpha)$$

$$\tan\theta=(a\cos\alpha-b\sin\alpha)/c$$

MEASURING THE INERTIA OF A LINK IN AN ARM:

For fast arm motions one requires a good dynamic model of the arm, including the geometry of joints and motor torques. Also required is the approximate moment of inertia of the links in the arm. It is usually not feasable to calculate these because of the complex shape and number of parts a link is made of. A simple empirical method requires one to measure the total mass and distance of the centre of gravity from the connection to the preceeding link as well as the period of oscillation when the link is suspended from this connection.



Let m = mass of link, l = distance of c.g. from preceeding connection
g = acceleration due to gravity (9.8 meter/second^2)
T = period of oscillation, I = moment of inertia

$$I\ddot{\theta} = -mgl \sin\theta \qquad \ddot{\theta} = -(mgl/I)\,\theta$$

$$\theta = A \cos(\sqrt{mgl/I}\,)t = A \cos(2\pi t/T)$$

$$T = 2\pi\sqrt{I/(mgl)}$$

$$I = mgl\,(T/2\pi)^2$$

Example: Pendulum made of string and heavy weight: $T = 2\pi\sqrt{(l/g)}$

$$I = mgl\,(l/g) = ml^2$$