PIVOT TRANSFORMS

ANNEX 2

BISECTION METHOD

In designing a bisection procedure we need to choose a suitable parameter and function which is to be solved. Referring to Figure 9 in the main text, for a given pixel on the screen we know the height h of the image plane, we know α_1 and α_2 and hence k, k_1 and k_2 from the bud transform **T** selected; and we know $C(x_1,y_1)$, $O(x_2,y_2)$ and W from the vortex selected. From h we calculate h'. We thus also know a,b,c and e. This leaves u and $F(x_3,y_3)$ to be found, and thence m, such that the planes pass through Q. There can be several solutions for different values of u, two for each value of u. The reason the solution is not straight forward is that the equation of the vortex is involved, which contains a power of u and hence we have a transcendental situation.

Now as u varies G moves along OC, and since Q is fixed for a given h, clearly F moves along a straight line which is the projection of OC from Q on the upper invariant plane. We find the equation of this line and from it find a function f(m)=0 which we use for the bisection i.e. we use m as our parameter. This was found to be better than using u.

Recalling that

$$x_3=\frac{a\left[ex_2+u(x_1-x_2)\right]}{e\left[a-e+u\right]}$$

we have

$$u=\frac{e\left[(a-e)x_3-ax_2\right]}{a(x_1-x_2)-ex_3} \qquad (1)$$

Similarly from y_3 we get

$$u=\frac{e\left[(a-e)y_3-ay_2\right]}{a(y_1-y_2)-ey_3}$$

Eliminating u and re-arranging gives the equation

$$\begin{aligned}&x_3\left[(a-e)(y_1-y_2)-ey_2\right]-y_3\left[(a-e)(x_1-x_2)-ex_2\right]+a\left[y_2x_1-y_1x_2\right]=0\\&i.e.\\&Ax_3-By_3+C=0\end{aligned}$$

where

$$\begin{aligned}A&=(a-e)(y_1-y_2)-ey_2\\B&=(a-e)(x_1-x_2)-ex_2\\C&=a(y_2x_1-y_1x_2)\end{aligned}$$

which gives us the line along which F is moving as u varies, since it is independent of u and A B C are constants.

Now from (13) in the main text

$$y_3 - mx_3 = \frac{y(1+m^2)}{kv}$$

$$\text{so } y_3 = \frac{y(1+m^2)}{kv} + mx_3 = \frac{Ax_3 + C}{B}$$

$$\text{giving } x_3 = \frac{\frac{x(1+m^2)}{mkv} + \frac{C}{B}}{m - \frac{A}{B}} \qquad (2)$$

Thus we can find x_3 and y_3 from m as x is determined by the selected pixel on the screen.

In Annex 1 we showed that

$$m^2\left[R^2 - (x_1 - x_3)^2\right] + 2\mathrm{m}(x_1 - x_3)(y_1 - y_3) + R^2 - (y_1 - y_3)^2 = 0$$

where

$$R = W\frac{e-u}{u}\left(\frac{u}{e}\right)^{\mu} \qquad (3)$$

This may be re-arranged as

$$(1+m^2)R^2 = \left[m(x_1 - x_3) - (y_1 - y_3)\right]^2$$

giving

$$y_3 = y_1 - m(x_1 - x_3) \pm R\sqrt{1+m^2} \qquad (4)$$

The sign ambiguity gives two solutions for y_3 and hence two tangent planes for a given m, so the bisection must be done twice.

Thus the procedure is:

1. Given m, calculate x_3 from the Equation (2)

2. Then calculate u from Equation (1)

3. Calculate R from (3)

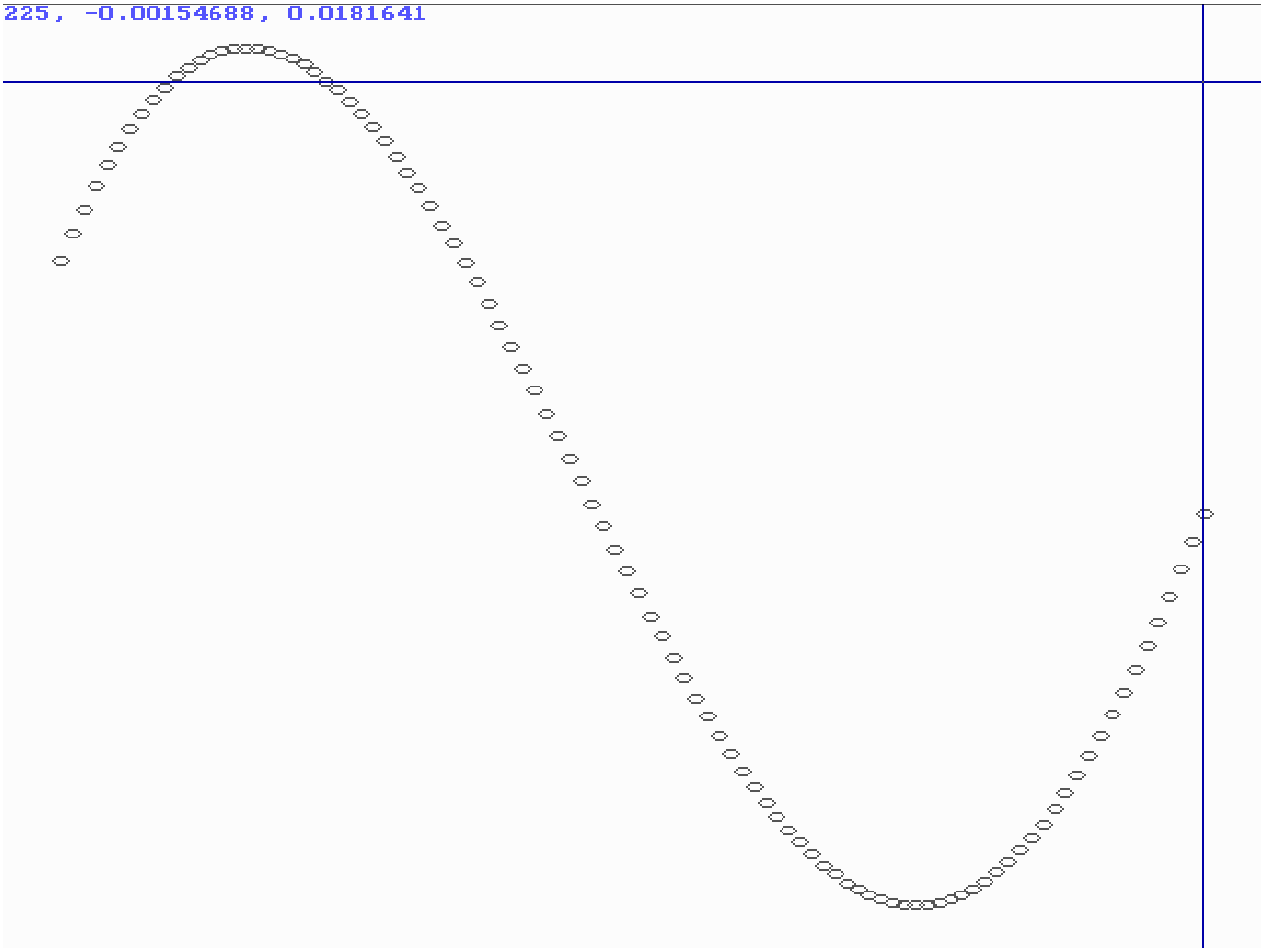
4. Find y_3 from (4)

5. Calculate x for the assumed value of m, from x'=-mkvy" (c.f. (13) in the main text)

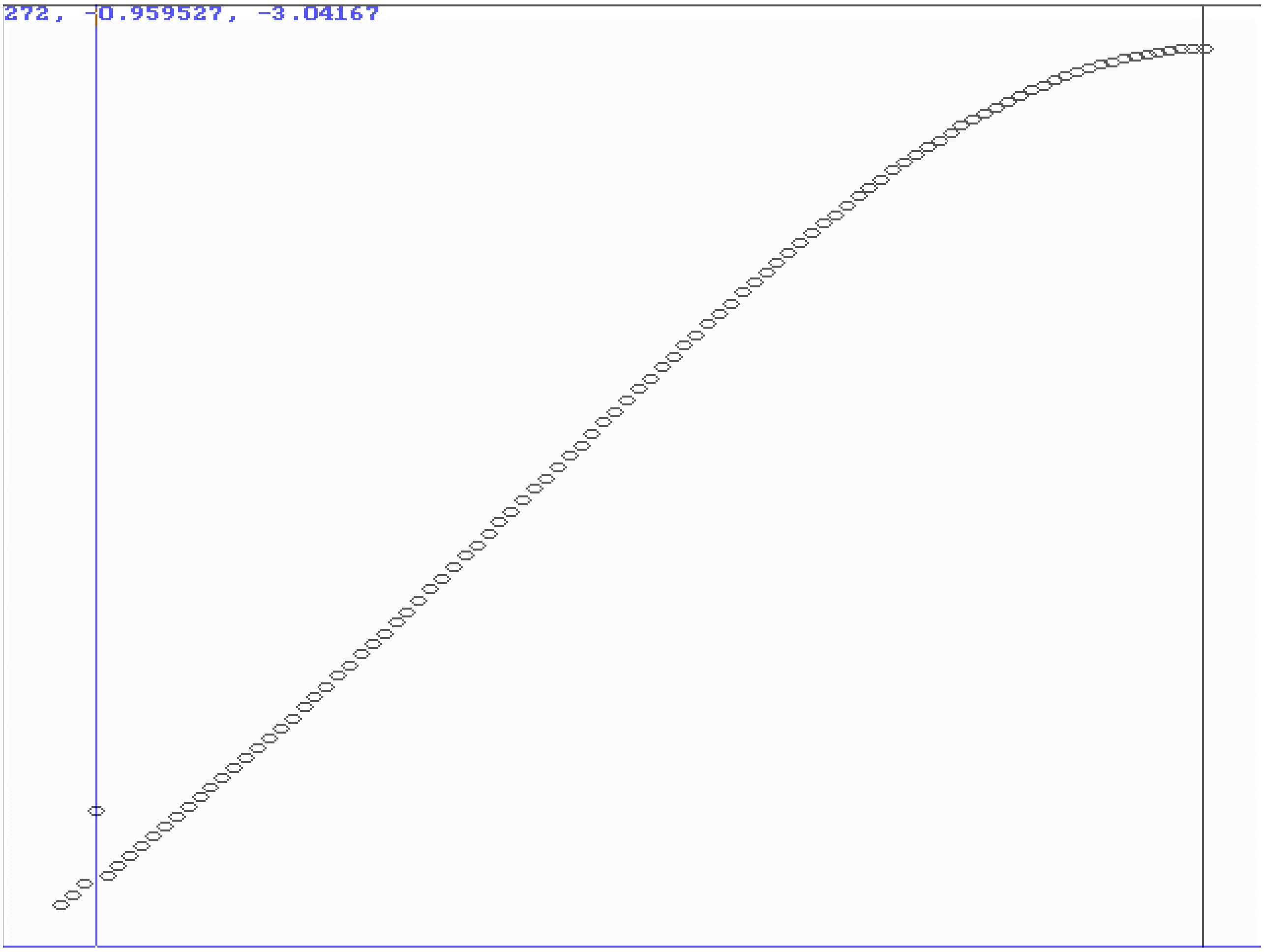
6. Return as the value of the bisection function the difference between x' and the known value of x (from the pixel position).

Thus varying m until the difference is (in principle) zero yields a solution for m and hence for the other parameters. In particular it enables the coordinate y orthogonal to the screen to be calculated. This is incorporated in a global search function which seeks all such solutions for the intersections with the horizontal profile, and hence finds the value of y nearest to the viewer.

It is here that real problems emerge as there is no basis for a normal bisection so the search procedure must start from the axis and step outwards. If the steps are too small production of a picture is excessively slow (all this is done for each pixel), but on the other hand if the stepping factor is too large roots that are close together may be missed. Plotting m against the bisection error as we move out from the axis gives a typical plot shown below.



Here one or both of the two roots could be missed by the search. Incorporation of an analysis of changes in the gradient solved cases such as this. But the following graph shows a much more severe case:



A very sharp “spike” occurs which will almost certainly be missed by a search algorithm, and indeed hardly shows on the graph! Sometimes these spikes can be predicted theoretically and a logical test for such cases is included in the algorithm.

The point is that some of the pictures have “blemishes”, and the explanation lies in difficulties such as this. If the wrong root is found (i.e. not nearest to the viewer) then the angle of the tangent plane calculated applies to a hidden part of the surface and hence presents a shading discontinuity. Although they could be manually “edited out” quite easily – and this has been done for some purposes - they are left in here as this is more a mathematical investigation than a “lovely picture” exercise. Also the location and type of blemish does convey information.