Determining the Center of Expansion for the optical flow occurring during video transects...

Daniel Davis

The goal is to find an effective algorithm that uses the movement of objects in video transects to find the point on the screen from which these objects seem to move. Also see the document "Getting Size Scale From a Single Moving Underwater Camera".

We model the geometry of imaging using a pin-hole model as shown in Figure 1.:

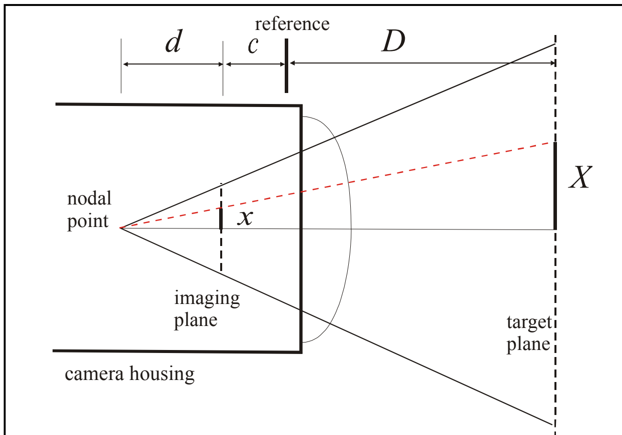

Figure 1. Model of camera geometry

In this model all points in a target are imaged in the image plane by projection to a single *nodal point* behind the image. The distance d is called the *model focal length.* It is the distance between the image plane and the nodal point. The constant c is called the *image plane offset*. We note that the imaging plane has a fixed width W and height H. The model focal length establishes the scale relationship between sizes in a target plane, the distance of the target plane from the camera, and size in the image plane, as shown in the relation:

$$\frac{X}{x} = \frac{D+d+c}{d} \qquad \text{(eq. 1)}$$

The model (horizontal) zoom angle is then determined by

$$\alpha = 2 \cdot atan\left(W / 2 \cdot d \right) . \qquad \text{(eq. 2)}$$

The values of the constants c and d can be determined by least squares from direct measurements of the ratio X/x for a selection of tow or more known distances D. For a variable zoom angle camera these constants must be determined for each zoom position.

During ROV transects the ROV moves in a relatively constant direction and video is recorded. It is generally NOT the case that the camera axis is mounted in the exact direction of ROV movement. In the resulting video the viewer then sees a stream of objects pass which appear to be moving away from a fixed point in the image. We develop an algorithm for finding this point on the image plane.

The first step is to model the movement of the camera through space as shown in Figure 2 below.

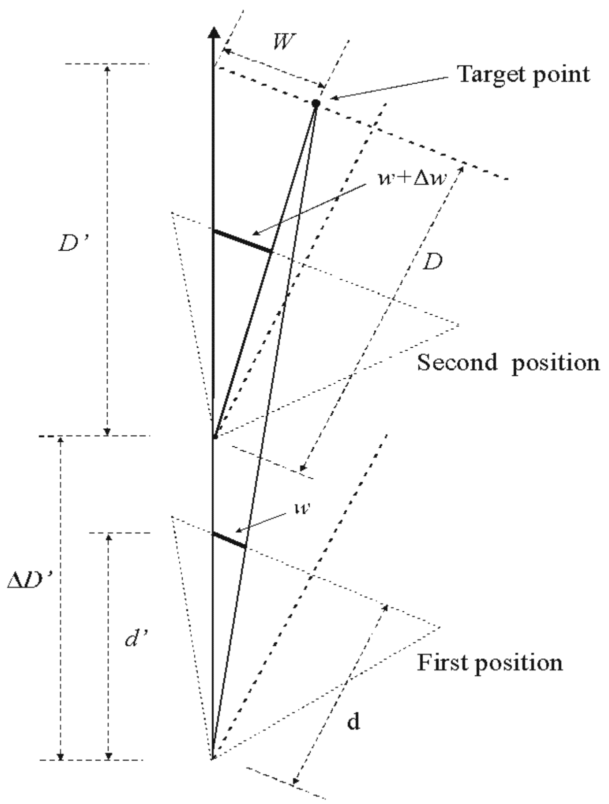

Figure 2. Geometry for single moving camera

This is a plan view looking down on the ROV/camera system with the direction of ROV movement in the plane as the main axis, and in this case with the camera turned slightly to the right of this axis. The only part of the camera shown is the image plane, with a focal length *d*. The camera is shown in two positions along this axis. In effect then, the point we are trying to determine is where the main axis of movement intersects the image plane of the camera. The image point of a target point is shown in each camera position., along with its distances w and $w + \Delta w$ illustrating the increase of distance of this point from the center of expansion between the two positions.

We do not assume that the camera axis is in the same plane as the axis of movement, since it could be pointed up or down, as well as to the right or left of the axis of movement. We do assume however that the distances w and $w + \Delta w$ are measured truly, that is in the plane defined by the axis of movement and the target point and in the image plane in each of the two positions. This is correctly visualized in Figure 2 if the plane of the view is set to the one containing the axis of movement and the target point.

From Figure 2, it is straightforward to show that:

$$D' = \left(\frac{w}{\Delta w}\right)\Delta D' . \qquad \text{(eq. 3)}$$

Or,

$$D' + \Delta D' = \left(\frac{w + \Delta w}{\Delta w}\right)\Delta D' . \qquad \text{(eq. 4)}$$

The value $D' + \Delta D'$ is the distance from the first position of the camera to the point where the image plane meets the target point. If then we considered a third position of the camera beyond the first two we would obtain a similar relation to the above:

$$D'' + \Delta D'' = \left(\frac{w + \Delta w'}{\Delta w'}\right)\Delta D'' , \qquad \text{(eq.5)}$$

where of course,

$$D' + \Delta D' = D'' + \Delta D'', \qquad \text{(eq.6)}$$

and where

$$\Delta D' < \Delta D''.$$

Thus it follows that the quantity

$$\left(\frac{w + \Delta w}{\Delta w}\right)\Delta D' = \left(\frac{w + \Delta w'}{\Delta w'}\right)\Delta D'' \qquad \text{(eq.7)}$$

is an *invariant.* In fact this quantity is the distance to be traveled from the initial position where a point has been observed to the position where the point will disappear behind the image plane. As such it is a ranging invariant along the direction of movement. We will show that this invariant can be used to characterize the center of expansion using the movement of points on the image as well as the movement of the ROV through the water.

First we note some properties of this invariant. Since it involves a *ratio* $\left(\frac{w + \Delta w}{\Delta w}\right)$ of image plane measurements, it is *scale-invariant* relative to the image. In other words distances in the image can be measured in screen coordinates or any other image coordinates and the invariant will remain the same. Secondly, distance measurements of the movement of the ROV can also be in any scale, since this invariant will still be a measurement of the same *distance*, but in the scale used. This also means that video timecodes (converted to elapsed time) can be used to measure $\Delta D'$ as long as we can assume that the speed (and direction) of the ROV is held constant.

We now show that the above quantity is an invariant only if the distances w , $w + \Delta w$ and $w + \Delta w'$, etc. defined by the moving point are measured relative to the true center of expansion. Suppose that given an initial position (X_0, Y_0), we have *N* screen additional positions (X_i, Y_i) of a moving point and for each position a relative distance S_i of movement along the ROV direction of movement from the ROV position at the initial point. Assume that (X_C, Y_C) is the center of movement. Then in these terms, the *N* values

$$ImageDist(X_i, Y_i, S_i; X_C, Y_C) = \frac{\left|(X_i, Y_i) - (X_C, Y_C)\right|}{\left|(X_i, Y_i) - (X_0, Y_0)\right|}\,(S_i - S_0)\,, \qquad \text{(eq.8)}$$

are equal. In particular, the *variance function* of these *N* values, defined by

$$Var(X, Y, S, x, y) = stdev(ImageDist(X_i, Y_i, S_i, x, y))\,, \qquad \text{(eq.9)}$$

equals 0 when $(x, y) = (X_C, Y_C)$. Moreover this occurs only when

$$ImageDist(X_i, Y_i, S_i, x, y) = ImageDist(X_j, Y_j, S_j, x, y) \;\; for\ any\ i\ and\ j\,. \qquad \text{(eq. 10)}$$

Figure 3 is a graph of $Var(X, Y, S, x, y)$ as a function of x, y, for (x, y) near the center of expansion (X_C, Y_C). The graph also shows the points (X_i, Y_i).

Note that there is a distinct minimum trough in this surface. This trough goes through the center point (X_C, Y_C) of expansion. (Note that in this example $N = 5$.)

Figure 4 shows this trough in greater detail (the red point is the center of expansion (X_C, Y_C)). Although the surface value is exactly 0 at the point (X_C, Y_C), points in the trough can have values numerically indistinguishable from 0, and therefore the point (X_C, Y_C) cannot be reliably found using a search for the

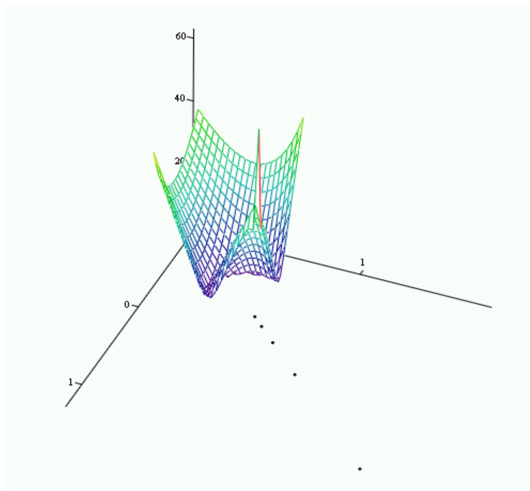

Figure 3. Graph of $Var(X, Y, S, x, y)$

minimum point. (In the example, $(X_C, Y_C) = (-0.5, 0.2)$).

However if we track multiple data sets (X, Y, S), (X', Y', S'),etc., of points moving from the center of expansion in different directions, it will follow that the sum of their corresponding functions $Var(X, Y, S, x, y)$, $Var(X', Y', S', x, y)$, etc. , will have a unique minimum at the center of expansion. Equivalently, we can use each track of points moving from the center of expansion to incrementally adjust the position of the center of expansion in a direction parallel to the movement of the points. In this way each track makes a robust contribution to our knowledge of the position of this center.

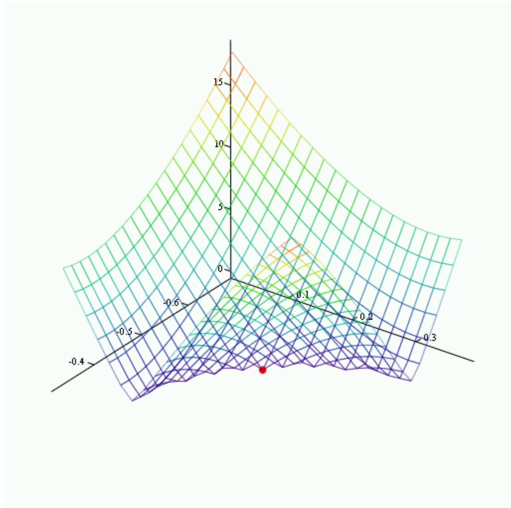

Figure 4. $Var(X,Y,S,x,y)$ near center of expansion.

Figure 5 shows the track of two points, with data sets (X,Y,S) and (X',Y',S'). The graph of the corresponding surface $Var(X,Y,S,x,y)+Var(X',Y',S',x,y)$ in the neighborhood of the point (X_C,Y_C) is shown in Figure 6. Clearly this surface has a unique minimum at this center of expansion.

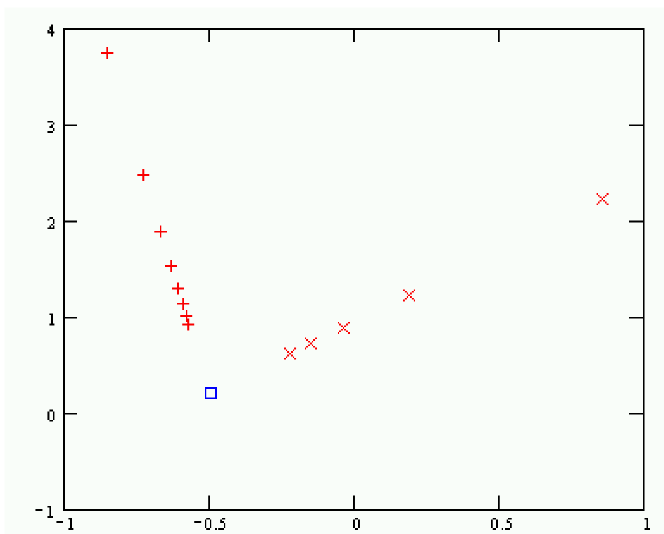

Figure 5. Tracks of moving points in image plane

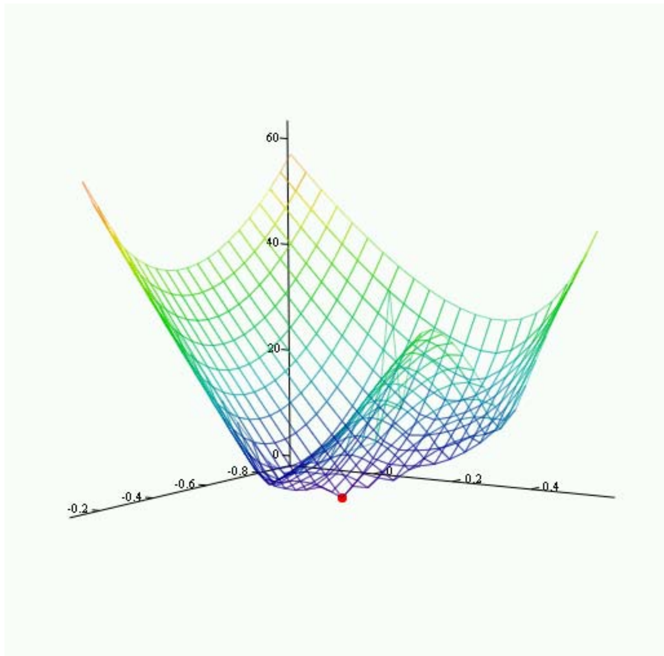

Figure 6. Surface
$Var(X,Y,S,x,y) + Var(X',Y',S',x,y)$

This gives two methods to find and maintain the center of expansion as tracks of points are identified. One is to modify an objective function by updating it by adding new tracks, while removing tracks that are 'outdated', (several seconds old). The objective function is minimized using the search as above. If this is done regularly (for example after several tracks in different directions are acquired), the center point should remain relatively stable. Alternatively, the search could be modified to look for the minimum point only along the track of the points (X,Y). The algorithm would then be used to adjust the point only in this direction. This modification would also speed up the search significantly.

Also to each track we associate the invariant calculated for it, along with its data. That is, we save the values (S,X,Y) and $ImageDist(S,X,Y,X_C,Y_C)$.

Note that by assumption

$$ImageDist(S,X,Y,X_C,Y_C) = D' + \Delta D' = D'' + \Delta D'' = ... \quad \text{(eq. 11)}$$

is the nodal distance, along the direction of ROV movement, from the *initial camera position* to the image plane containing the point. It follows from Figure 2, and eqs 5 and 6, that for any *k*,

$$D_k' = ImageDist(S_k,X_k,Y_k,X_C,Y_C) - (S_k - S_0). \quad \text{(eq. 12)}$$

is the slant distance of the nodal point at the k^th^ position of the camera to the point imaged. If we let,

$$a = \left|(X_C,Y_C\right| = \sqrt{X_C{}^2 + Y_C{}^2}, \quad \text{(eq. 13)}$$

then from Figure 2, given the model focal length *d*, the factor,

$$\cos(\arctan(a/d)) = \frac{d}{\sqrt{a^2+d^2}} \quad \text{(eq. 14)}$$

equals the ratio of nodal distances along the camera axis to nodal distances along the direction of motion.

Therefore, by eq. 1,

$$D_k = D_k' \frac{d}{\sqrt{a^2 + d^2}} \qquad \text{(eq. 15)}$$

is the distance to the target point *along the camera axis* when the camera is at the kth position. Thus if we know the model focal length d, it follows from eq. 1 that the scale ratio of the size X of any object at the target distance to its image size x at the kth position satisfies:

$$\frac{X}{x} = \frac{D_k}{d} = \frac{D_k'}{\sqrt{a^2 + d^2}} \qquad \text{(eq. 16)}$$

Thus once we have the image plane coordinates of the center of expansion, given the model focal length, we also have the size scale for the target image at each position of the points in a track.

APPENDIX A – MathCad routine used to create figures and calculate center of expansion. The routine returns the coordinates of the center of expansion, the value if the *ImageDistance* function, and the number of steps it took for the algorithm to find the minimum point. The algorithm must be initiated with the coordinates of the best estimate of the center of expansion, a search step size *delta*, and a maximum number n of steps.

$$\mathrm{Objective}(x, y) := \mathrm{Variance}(S1, X1, Y1, x, y) + \mathrm{Variance}(S2, X2, Y2, x, y)$$

MathCad Algorithm to search for a point that minimizes the the Objective function...

```
Search_MinObjective(x, y, delta, n) := | deltax ← [0, delta, -delta]^T
                                       | deltay ← [0, delta, -delta]^T
                                       | for i ∈ 0.. n
                                       |   | min ← Objective(x, y)
                                       |   | minx ← x
                                       |   | miny ← y
                                       |   | for j ∈ 0.. last(deltax)
                                       |   |   for k ∈ 0.. last(deltay)
                                       |   |     | testx ← x + deltax_j
                                       |   |     | testy ← y + deltay_k
                                       |   |     | testmin ← Objective(testx, testy)
                                       |   |     | if testmin < min
                                       |   |     |   | minx ← testx
                                       |   |     |   | miny ← testy
                                       |   |     |   | min ← testmin
                                       |   | if (minx ≠ x) + (miny ≠ y)
                                       |   |   | x ← minx
                                       |   |   | y ← miny
                                       |   | break otherwise
                                       | [x, y, min, i]^T
```

Algorithm to do a directional search from the track points toward the center of expansion..

```
Search_Directional(S, X, Y, x, y, delta, steps) :=
    for i ∈ 0.. last(X) − 1
        DX_i ← X_0 − X_(i+1)
        DY_i ← Y_0 − Y_(i+1)
    unit_0 ← mean(DX)
    unit_1 ← mean(DY)
    unit ← unit / |unit|
    dx ← delta·unit_0
    dy ← delta·unit_1
    minx ← x
    miny ← y
    min ← Variance(S, X, Y, x, y)
    for i ∈ 0.. steps − 1
        testx ← minx + dx
        testy ← miny + dy
        testmin ← Variance(S, X, Y, testx, testy)
        if testmin < min
            minx ← testx
            miny ← testy
            min ← testmin
        break otherwise
    [minx, miny, min, i, unit]^T
```

Tests of each algorithm…

Use a weighted combination of the first two points to create a seed point toward the center of
from the first point and away from the second...

$$DELTA := 0.001 \quad STEPS := 500 \quad DELTASTEPS := 0.5$$

$$TEST1 := \begin{bmatrix} X1_0 \\ Y1_0 \end{bmatrix} \cdot 3.0 + \begin{bmatrix} X1_1 \\ Y1_1 \end{bmatrix} \cdot -2.0$$

$$Search\_MinObjective\left(TEST1_0, TEST1_1, DELTASTEPS\right) = \begin{bmatrix} -0.50124 \\ 0.20064 \\ 0.03764 \\ 215 \end{bmatrix} \qquad C = \begin{bmatrix} -0.5 \\ 0.2 \end{bmatrix}$$

Test the directional search algorithm using each set of track points....

$$SOLN1 := Search\_Directional\left(S1, X1, Y1, TEST1_0, TEST1_1, DELTA, STEPS\right)$$

$$SOLN1 = \begin{bmatrix} -0.50003 \\ 0.19996 \\ 0.00355 \\ 252 \\ \{2,1\} \end{bmatrix} \qquad C = \begin{bmatrix} -0.5 \\ 0.2 \end{bmatrix}$$

Test using the second set of track points...

$$TEST2 := \begin{bmatrix} X2_0 \\ Y2_0 \end{bmatrix} \cdot 3.0 + \begin{bmatrix} X2_1 \\ Y2_1 \end{bmatrix} \cdot -2.0$$

$$SOLN2 := Search\_Directional\left(S2, X2, Y2, TEST2_0, TEST2_1, DELTA, STEPS\right)$$

$$SOLN2 = \begin{bmatrix} -0.50406 \\ 0.24059 \\ 1.40224 \\ 499 \\ \{2,1\} \end{bmatrix} \qquad C = \begin{bmatrix} -0.5 \\ 0.2 \end{bmatrix}$$