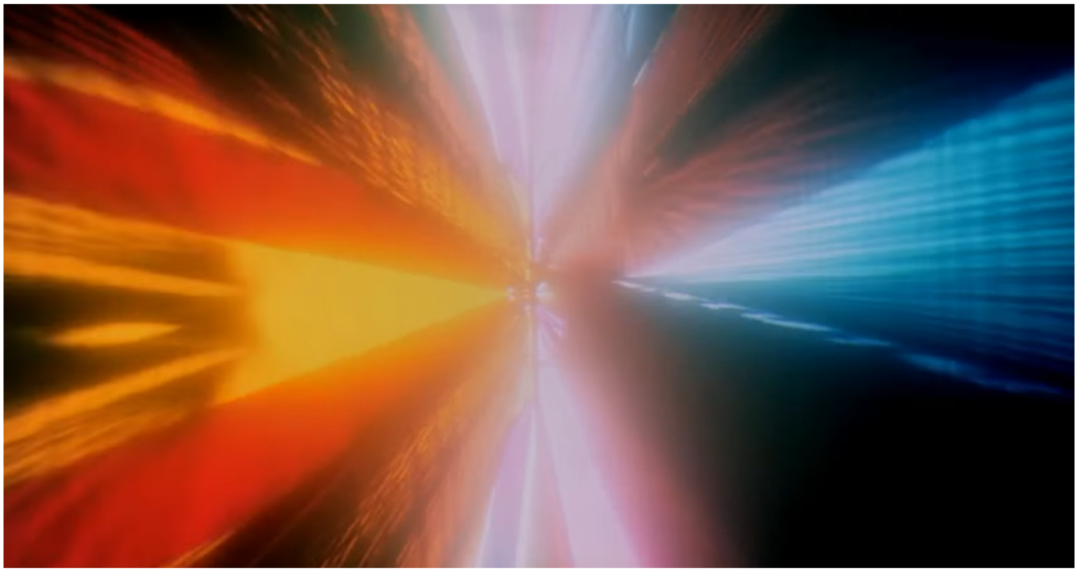
Slit Scan – Generate an effect similar to 2001's star gate sequences

Fred Mitchell

May 8, 2022

Contents

1 Fundamentals and Design



1.1 Introduction

Slit-Scan is the command-line tool that allow you to create slit-scans similar to the star-gate sequences in 2001: A Space Odyssey. The idea here is to add a lot of flexibility to allow you to acheive results even beyond what was acheived in what is arguably the greatest Science Fiction film of all time.

1.2 Mathematics

Here, we fully specify the mathematics involved with the slit-scan function.

1.2.1 Generating the slit

Firstly, we define the slit function as a function between 2 points.

t is time, in seconds, and

p_1, p_2 are points defining tbe beginning and ending of the slit, and

$$slit(p_1, p_2, \rho) = g(p_1, p_2, curve(\rho))\Big|_{\rho=0}^{\rho=1}$$

where

$$p_1 = (x_1, y_1)$$
$$p_2 = (x_2, y_2)$$

and $0 \geq x, y, \rho \geq 1$

where $curve(\rho)$ defines the shape of the slit. For the traditional case, $curve(\rho)$ will be 0, defining a straight line from p_1 to p_2. In other cases, $curve(\rho)$ will result in pertubations in the (p_1, p_2) line, perpendular to the line itself.

ρ is the parametic for the slit. x and y represnts the idealized coordinates of the image being scanned by the slit, which will be converted to the actual physical coordinates of the pixels in the image, along with pixel averaging in the 3x3 or 5x5 square with the physical coordinate being at the center.

Secondly, we define the movement of the slit across an image in terms of time t (in seconds):

$$p_{1t} = p_1(t)$$
$$p_{2t} = p_2(t)$$

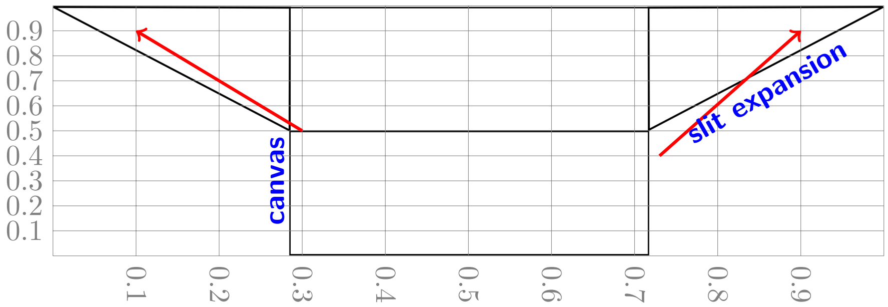
Where said movement of p_1 and p_2 may be independent with each other, solely with t as their common parametric basis, but in most cases the "trivial" or traditional approach will be taken as the movenent being equivalent, basically to "scan" the image from one side to the other.

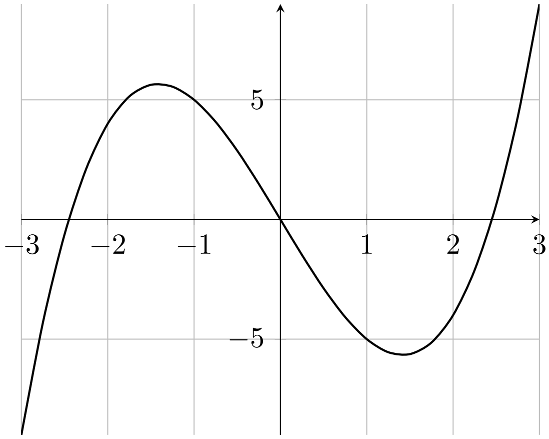
Thirdly, we define the compositor function. The compositor is responsible for compositing the time-varying contents of the slit unto the final canvas. Here, the canvas is bisected either horizontally or vertically by a straight-line slit, and with each time instant, the slit contents is forwarded away from the slit.

- The Hip library provides a way to navagate the image via map functions. The imap provides (x,y) coordinates of the pixels given. We can and will problably have to do do some sort of lookup "table" on the coordinates to perform the expected operation, but I need to give this more thought.

1.2.2 The Compositor

The compositor may "stretch" the contents of the slit as a function of the distance from the slit until it reaches the end of the canvass.





$$\frac{f(x)-f(a)x-a}{g}$$

1.2.3 Actual implementation of the mathematics

We shall do our scans from the canvas, and map that to the two source images, because it simplifies things with regards to smoothing between pixels, etc.

1. Matrix transforms and the math coding conventions

    - In the code, A_x, A_y, ..., D_x, D_y shall be represented by ax, ay..., and likewise, M_{11}, M_{12}, ..., shall be represented by $m11$, $m12$...
    - We shall designate the former as "square arrows", and the latter, of course, are the parameters in the transformation matrix.
    - Note that our origin is in the upper-left corner, and positive moves to the right and down, for the x any y axis, respectively. This reflects the normal way computer graphics are always implemented.

    The following was taken from: https://blog.plantingcode.net/2d-only-perspective-transfor

    To make things easier, the solution maps a unit square to the square defined by the pair of vectors. In other words:

$$
\begin{aligned}
(0,0,1) &\mapsto A \\
(0,1,1) &\mapsto B \\
(1,0,1) &\mapsto C \\
(1,1,1) &\mapsto D
\end{aligned}
$$

Using the above notation, it is possible to start calculating the matrix which will take any 2D point in unit coordinates and convert it to projected coordinates.

$$
\begin{bmatrix}
M_{11} & M_{12} & M_{13} \\
M_{21} & M_{22} & M_{23} \\
M_{31} & M_{32} & 1
\end{bmatrix} \tag{1}
$$

With the mapping to point A being straightforward, two entries can already be solved for (subscripts x and y are for the 2D components):

$$M_{13} = A_x \tag{2}$$

$$M_{23} = A_y \tag{3}$$

Using the unit mappings with the matrix and factoring in the perspective divide required for points gets the following system of equations for matrix elements to known points.

$$\frac{M_{12} + A_x}{M_{32} + 1} = B_x \tag{4}$$

$$\tag{5}$$

$$\frac{M_{22} + A_y}{M_{32} + 1} = B_y \tag{6}$$

$$\tag{7}$$

$$\frac{M_{11} + A_x}{M_{31} + 1} = C_x \tag{8}$$

$$\tag{9}$$

$$\frac{M_{22} + A_x}{M_{31} + 1} = C_y \tag{10}$$

$$\tag{11}$$

$$\frac{M_{11} + M_{12} + A_x}{M_{31} + M_{32} + 1} = D_x \tag{12}$$

$$\tag{13}$$

$$\frac{M_{21} + M_{22} + A_x}{M_{31} + M_{32} + 1} = D_y \tag{14}$$

Multiplying both sides by the left-hand denominator and arranging all matrix terms to the left and all constant terms to the right changes the equations to:

$$\begin{aligned}
M_{12} - B_x M_{32} &= B_x - A_x \tag{15}\\
M_{22} - B_y M_{32} &= B_y - A_y \tag{16}\\
M_{11} - C_x M_{31} &= C_x - A_x \tag{17}\\
M_{21} - C_y M_{31} &= C_y - A_y \tag{18}\\
M_{11} + M_{12} - D_x M_{31} - D_x M_{32} &= D_x - A_x \tag{19}\\
M_{21} + M_{22} - D_y M_{31} - D_y M_{32} &= D_y - A_y \tag{20}
\end{aligned}$$

Setting up the equations to perform Gaussian Elimination on them yields:

$$\begin{array}{cccccccc}
M_{11} & M_{12} & M_{21} & M_{22} & M_{31} & M_{32} & & \\
1 & 0 & 0 & 0 & -C_x & 0 & = & C_x - A_x \\
0 & 1 & 0 & 0 & 0 & -B_x & = & B_x - A_x \\
0 & 0 & 1 & 0 & -C_y & 0 & = & C_y - A_y \\
0 & 0 & 0 & 1 & 0 & -B_y & = & B_y - A_y \\
1 & 1 & 0 & 0 & -D_x & -D_x & = & D_x - A_x \\
0 & 0 & 1 & 1 & -D_y & -D_y & = & D_y - A_y
\end{array}$$

The top four rows are already in a nice enough form that their equations can be extracted directly:

$$\begin{aligned}
M_{11} &= (1 + M_{31})C_x - A_x \tag{21}\\
M_{12} &= (1 + M_{32})B_x - A_x \tag{22}\\
M_{21} &= (1 + M_{31})C_y - A_y \tag{23}\\
M_{22} &= (1 + M_{32})B_y - A_y \tag{24}
\end{aligned}$$

Subtracting out the first two rows from the fifth row and the middle two rows from the last row results in:

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & C_x - D_x & B_x - D_x & = & D_x + A_x - B_x - C_x \\
0 & 0 & 0 & 0 & C_y - D_y & B_y - D_y & = & D_y + A_y - B_y - C_y
\end{array}$$

These two equations serve as the starting point for determining the M_{31} and M_{32} terms. Focusing on simplifying the M_{31} term first through division yields:

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & \frac{B_x - D_x}{C_x - D_x} & = & \frac{D_x + A_x - B_x - C_x}{C_x - D_x} \\
0 & 0 & 0 & 0 & 1 & \frac{B_y - D_y}{C_y - D_y} & = & \frac{D_y + A_y - B_y - C_y}{C_y - D_y}
\end{array}$$

Subtraction of the top equation from the bottom and subsequent division results in the very large (but ultimately very simple) equation for M_{32}:

$$
\begin{align}
\left(\frac{B_y - D_y}{C_y - D_y} - \frac{B_x - D_x}{C_x - D_x}\right) M_{32} &= \frac{D_y + A_y - B_y - C_y}{C_y - D_y} - \frac{D_x + A_x - B_x - C_x}{C_x - D_x} \tag{25}\\
M_{32} &= \frac{\frac{D_y+A_y-B_y-C_y}{C_y-D_y} - \frac{D_x+A_x-B_x-C_x}{C_x-D_x}}{\frac{B_y-D_y}{C_y-D_y} - \frac{B_x-D_x}{C_x-D_x}} \tag{26}
\end{align}
$$

Dividing out the M_{32} term instead before subtracting the bottom equation from the top and dividing solves for M_{31}.

$$
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & \frac{C_x-D_x}{B_x-D_x} & 1 & = & \frac{D_x+A_x-B_x-C_x}{B_x-D_x}\\
0 & 0 & 0 & 0 & \frac{C_y-D_y}{B_y-D_y} & 1 & = & \frac{D_y+A_y-B_y-C_y}{B_y-D_y}
\end{array}
$$

$$
M_{31} = \frac{\frac{D_x+A_x-B_x-C_x}{B_x-D_x} - \frac{D_y+A_y-B_y-C_y}{B_y-D_y}}{\frac{C_x-D_x}{B_x-D_x} - \frac{C_y-D_y}{B_y-D_y}}
$$

Continuing forward with M_{31} and using algebraic techniques to consolidate and simplify terms brings about:

$$
\begin{align}
M_{31} =& \frac{(D_x + A_x - B_x - C_x)(B_y - D_y) - (D_y + A_y - B_y - C_y)(B_x - D_x)}{(B_x - D_x)(B_y - D_y)} \tag{27}\\
& \cdot \frac{(B_x - D_x)(B_y - D_y)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{28}\\
& \tag{29}\\
=& \frac{(D_x + A_x - B_x - C_x)(B_y - D_y) - (D_y + A_y - B_y - C_y)(B_x - D_x)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{30}\\
& \tag{31}\\
=& \frac{(A_x - C_x)(B_y - D_y) - (B_x - D_x)(B_y - D_y) - (A_y - C_y)(B_x - D_x) + (B_x - D_x)(B_y - D_y)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{32}\\
& \tag{33}\\
=& \frac{(A_x - C_x)(B_y - D_y) - (A_y - C_y)(B_x - D_x)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{34}
\end{align}
$$

Doing the same for M_{32}:

$$M_{32} = \frac{(D_y + A_y - B_y - C_y)(C_x - D_x) - (D_x + A_x - B_x - C_x)(C_y - D_y)}{(C_y - D_y)(C_x - D_x)} \tag{35}$$

$$\cdot \frac{(C_y - D_y)(C_x - D_x)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{36}$$

$$\tag{37}$$

$$= \frac{(D_y + A_y - B_y - C_y)(C_x - D_x) - (D_x + A_x - B_x - C_x)(C_y - D_y)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{38}$$

$$\tag{39}$$

$$= \frac{(A_y - B_y)(C_x - D_x) - (C_x - D_x)(C_y - D_y) - (A_x - B_x)(C_y - D_y) + (C_x - D_x)(C_y - D_y)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)}$$

$$\tag{40}$$

$$\tag{41}$$

$$= \frac{(A_y - B_y)(C_x - D_x) - (A_x - B_x)(C_y - D_y)}{(B_y - D_y)(C_x - D_x) - (B_x - D_x)(C_y - D_y)} \tag{42}$$

With the equations simplified it becomes a little easier to see that it is a combination of several vectors based on the provided A, B, C, D points. Also notable is that M_{31} and M_{32} share a denominator. Using $^{\curvearrowleft}$ to be mean rotating a vector counter-clockwise by 90°, the equations for each matrix component become:

$$u = \overrightarrow{DC} \cdot \overrightarrow{DB}^{\curvearrowleft} \tag{43}$$

$$M_{31} = \frac{\overrightarrow{CA} \cdot \overrightarrow{DB}^{\curvearrowleft}}{u} \tag{44}$$

$$M_{32} = \frac{\overrightarrow{DC} \cdot \overrightarrow{BA}^{\curvearrowleft}}{u} \tag{45}$$

$$\tag{46}$$

$$M_{11} = (1 + M_{31})C_x - A_x \tag{47}$$

$$M_{12} = (1 + M_{32})B_x - A_x \tag{48}$$

$$M_{21} = (1 + M_{31})C_y - A_y \tag{49}$$

$$M_{22} = (1 + M_{32})B_y - A_y \tag{50}$$

Placing these calculated values in the matrix allows for transforming arbitrary points from normalized space into the projected space defined by $\overrightarrow{AB}$ and $\overrightarrow{CD}$. It is important to divide by the resulting z term to normalize the resulting 2D coordinate just as you would divide by the resulting w term for a 3D perspective transformation.

2 General Documentation and Use of Slit Scan

This covers the usage from the command-line. Rather than go into extensive documentation on the command-line commands and options (the program itself provides this), we shall give some helpul examples.

2.1 Simple case – one image, simple vertical slit

```
slit-scan –i1 flower.jpg –format png –out film/
```

2.2 Typical – 2 images simple vertical and horizontal slits

2.3 Advanced 1 – 2 images, staight line rotating slit

2.4 Advanced 2 – 2 images, sine slit

2.5 Advanced 3 – 2 images, rotating sine slit

2.6 Advanced 4 – 2 images, arbitrary function for the slit and the 2 endpoints