Week 11 Report (8/18/16)

|  |
| --- |
| * Vector values are centers are not preserved. Blending of vectors are harsh and need to be smoothed. The elements have too much influence over each other. |

Weights

User have ability with manipulation of weight values. I have added the interface for manipulating weight values. I will continue its implementation by finishing the calculation required and redraw the vector values. Weight values will be changed by being able to change each element through selection. Additional work is also needed to fix element interaction and blending between vector values.

Barycentric Interpolation

To obtain the vector values at any given point, I applied barycentric interpolation. Vector values are stored upon the vertices of the triangle mesh. Given a coordinate, I will find the triangle that is the container for the x,y coordinate. This will allow me to calculate the vector value any an given coordinate. Being able to calculate the barycentric coordinates allows us to also perform the calculations to find the vector values at the desired coordinate by using its relativeness from each vertex. Sample calculations are below.

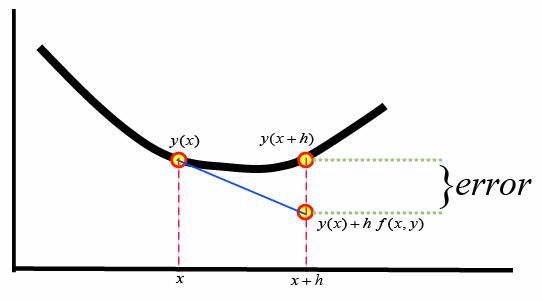
Barycentric coordinates: v+u+w=1

vx = u\*vx1+v\*vx2+w\*vx3

vy = u\*vy1+v\*vy2+w\*vy3

Streamlines

For calculating and drawing base streamlines, I started with Euler integration. Euler uses less calculations and was an easy starting point in working with the vector field in find the differential function for the field. Euler integration draws the streamline based on the current vector value at a coordinate. At the next coordinate it does the same until the streamline is finished. The problem with Euler is that the approximation is not exact enough therefore the drawn streamline deviates from actual streamline. Applying Runge–Kutta integration was simple since I had a reliable implementation of integrating the field. RK calculates the weighted average between the slope increments of beginning, midpoint and end of the interval. RK4 uses more estimates of the slope which provides less error in the values which makes the streamline a lot more precise.



|  |  |
| --- | --- |
| C:\Users\ootut\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture2.png  Euler | Runge–Kutta |

Centers should present their perfect circles as shown. However, my code encounters problems that mess with the streamline values that I need to fix. I have started with streamline implementation. A few problems do occur when multiple elements are presented such as the streamline not following the vector flow. I believe the problem with the miscalculation of the streamline would be not updating the vector values before recalculating them.

LIC Implementation

Once I fix my streamline to work properly, I will begin to create the LIC image. With a completed streamline, I will save its pixel coordinates and render it with the white noise image. The noise texture and streamline at the pixel are averaged out to get the desired pattern. I have also found a code example online by Bill Martin in case I need additional reference. An example convolution algorithm is below.

