Fall 2015 Computational Art Zoë Wood

1 Lab 6 - Implicit Lines and Circles

Goals

The goals for this lab are:

1. Practice using a `loop` control structure to create an image made of `strokes` based on `implicit lines`.
2. Practice using `implicit lines` and `implicit circles` and the distances from these equations to create a scene or object.
3. Create new stroke styles using patterns of `points, lines, and ellipses` that model the textures seen in many impressionist paintings.

Modality

Individual

Overview

Impressionism: "Impressionist art is a style in which the artist captures the image of an object as someone would see it if they just caught a glimpse of it. They paint the pictures with a lot of color and most of their pictures are outdoor scenes. Their pictures are very bright and vibrant." [1]

In particular, in this lab, we will focus on creating a scene that has very obviously textured strokes that just gives the observer the impression of a scene. We will be combining our knowledge of implicit equations in order to 'order' space as either on one side or the other of lines and curves in order to model very simple scenes.

Sample Impressionist paintings:

[1] http://www.artmovements.co.uk/impressionism.htm

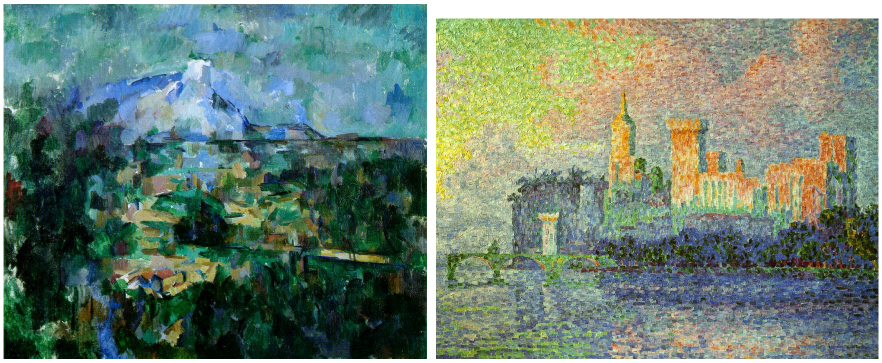

Figure 1: (left) Paul Czanne: Mount Sainte-Victoire view from Lauves” and (right) Paul Signac: ”Le Palais des Papes, Avignon (The Papal Palace, Avignon)

Figure 2: Vincent Van Gogh: ”Starry Night Over the Rhone”

Details

This lab requires you to create an image using an implicit representation of geometry that is drawn using shapes to emulate paint strokes. Implicit functions can represent common geometric shapes like circles, lines and curves in two dimensions (and similarly planes and spheres in 3D). For every point in 2D space, for example, the implicit function for a circle will return a signed value proportional to the Euclidean distance of that point to the circle. The fact that the distance value is *signed* allows you to distinguish which points are on the interior of the circle and which are on the exterior. Specifically, for every sampled x, y point in space, by defining the center of a circle, (cx, cy) and radius r, the implicit equation for a circle is:
$f(x, y) = (x - cx)^2 + (y - cy)^2 - r^2 = 0$

Using this equation and then looping over your entire screen (for a given sample rate), you can for example decide to draw points inside the circle one color and points outside the circle another color. For example, drawing an ellipse for every 10th pixel, blue on the outside, and black on the inside, you can create an sketch similar to this (see example code under resources):

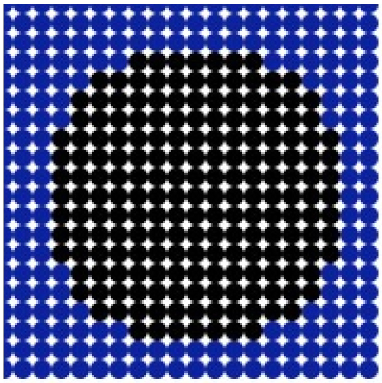

Figure 3: Implicit circle sketch

Similarly, you can create an image using an implicit equation for a line to create various images that color the image differently on two sides of a line, such as this:

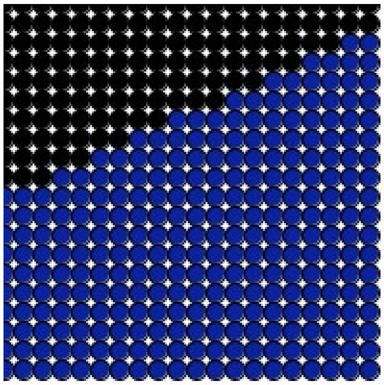

Figure 4: Implicit Line sketch

To make these sketches look more like a painting, you can vary the shape drawn and color slighting, using random. Such as this variation on the circle and the line:

Task:
You must create an image using Processing which emulates paint strokes for and uses a combination of both implicit circle(s) and an implicit lines. For this sketch you must create a scene that models a composite scene using lines and circles, for example trees and the sun *created using implicit lines and implicit circles* as bounding lines for objects. textbfWe suggest you start with one shape at a time, for example try to just replicate the 'messy sun' (shown in Figure 1) first.

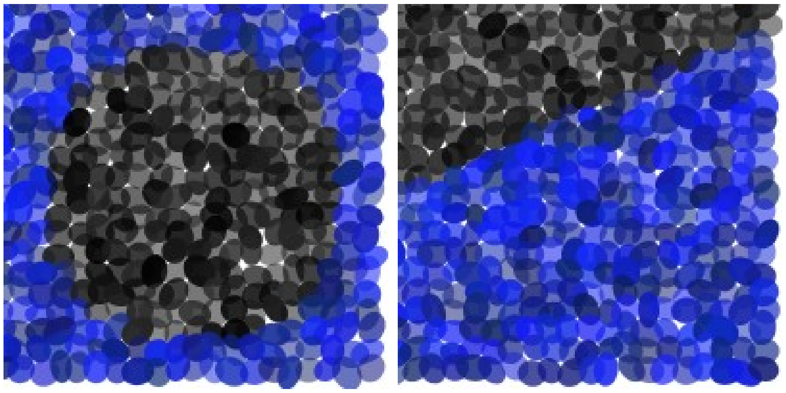

Figure 5: More painterly renderings of an implicit circle and implicit line

For the sketch you must use the magnitude of the implicit function (which is proportional to distance) to determine how far the stroke is from the original line and which side of the line/circle the equation falls. Use this distance to determine whether this stroke should be colored and, if so, what color. For example, for a line or circle, using the distance to weight the color might look like one of the images shown in Figures 8. Be creative about alternating your stroke style and a random radius for each point.

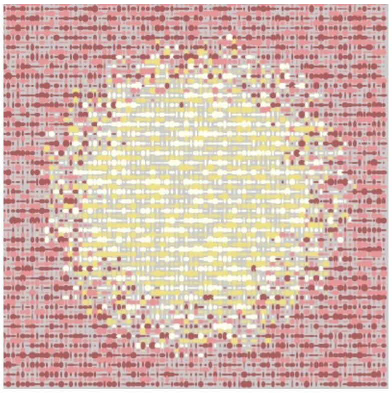

Figure 6: Painterly rendering of a sun using Processing

Your lab must:

- be at least 400 x 400
- be in color
- create a scene using both `implicit lines` and `implicit circles` (you must use at least 3)
- use the sign and the distance (magnitude) from the implicit line/circle to determine the color

- use `points, lines, or ellipses` to create the same scene with two different stroke types
- strokes should vary in color across the scene
- strokes should model the style of the textures found in impressionist art

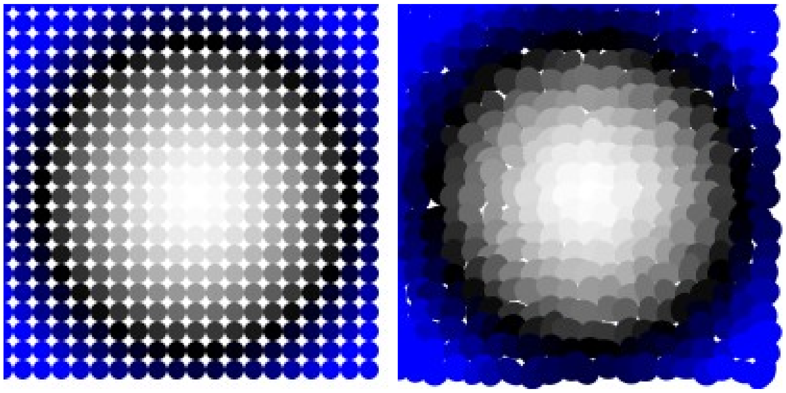

Figure 7: Using the distance to weight the color of 'strokes' for an implicit circle.

The scene should be created using strokes that are similar to the textures found in impressionist paintings. The final scene should then be created with at least two different stroke types (i.e.: by varying the sizes of the strokes or how close together the strokes appear to each other).

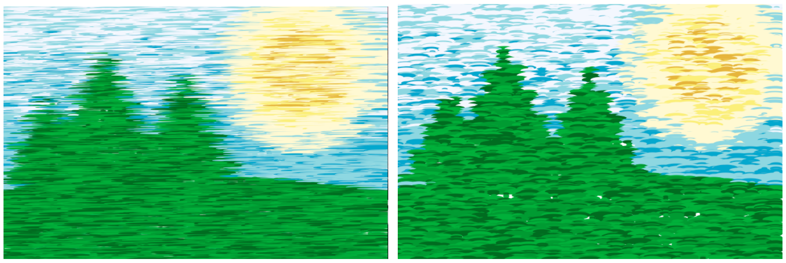

Figure 8: An example of using implicit circles for the ground plane and the sun and using implicit lines to find the intersection of three lines for each tree. The distance from the ground is used to determine the color of the sky and the distance from the center of the circle determines the color of the sun. Both of these distances determine the color intensity. Varying sizes of ellipses were used to create horizontal strokes and create different stroke types on the images.

Demo:

In order to receive credit for this lab, you must demo your sketch to your instructor or TA. For every lab, your score will be broken down 75% for meeting the technical requirements and 25% for aesthetics.

Submitting your sketch

You must submit your pde file on polylearn and you must post an image of your sketch to your pinterest Computational Art board. Please also pin any reference art.

Resources:

- Implicit Line - given two end points x_0, y_0 and x_1, y_1 the implicit line equation is:
  $f(x, y) = (y_0 - y_1) * x + (x_1 - x_0 *)y + x_0 * y_1 - x_1 * y_0 = 0$
- Implicit circle - given a center (cx, cy) and radius r:
  $f(x, y) = (x - cx)^2 + (y - cy)^2 - r^2 = 0$

```
/* Fall 2015 - example implicit circle sketch - ZJ Wood */

void setup() {
  size(400, 400);
}

/* method to evaluate the implicit equation for a circle */
float impl_circ(float cx, float cy, float radi, float in_x, float in_y) {
  return ( (in_x-cx)*(in_x-cx) + (in_y-cy)*(in_y-cy) - radi*radi);
}

void draw() {

  background(255);
  noStroke();
  for (int y=0; y < height; y+= 5) {
    for (int x=0; x < width; x+=5) {
      if (impl_circ(width/2.0, height/2.0, 100, x, y) < 0.0) {
        fill(128, 128);
      } else {
        fill(12, 34, 56, 128);
      }
      ellipse(x, y, 5, 5);
    }
  }
}
```