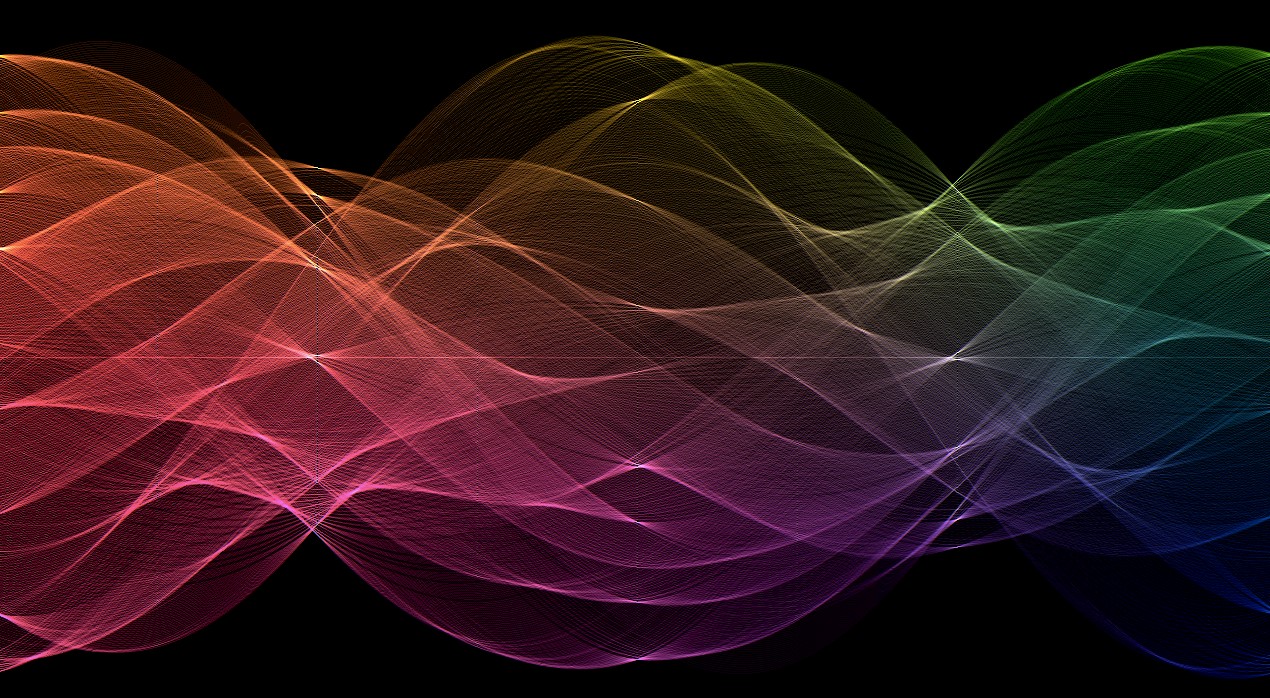
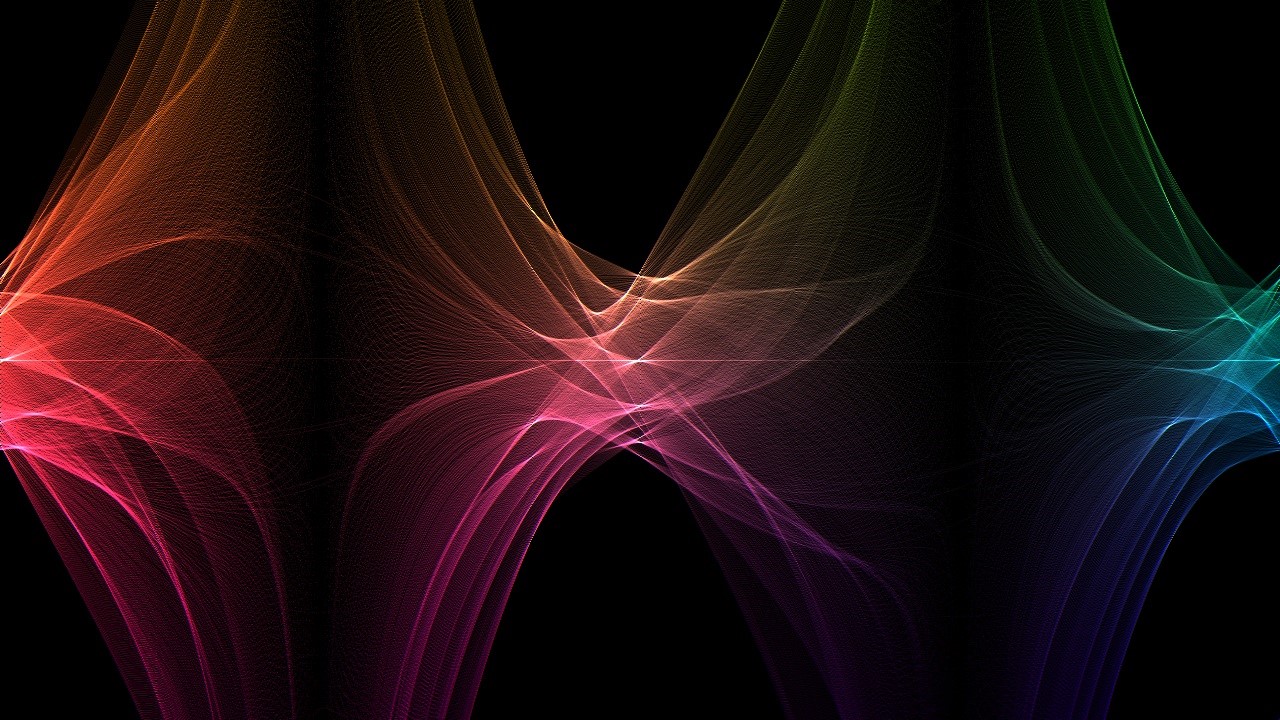
**CT Unit 8: Digital Artist Project Group: 10 Name: Ryan Zhang**

*Mr. Lin*

# Introduction

The purpose of the *Digital Artist* project is for students to learn the foundation of digital images and the skills to create digital art using mathematical transformations. Students will have to implement their programs using a high-level programming language (such as *C*) and experiment their own mathematical functions to create different artistic expressions in digital forms. Through this exercise, students will be able to link mathematics, computer programming, and art creation all together.

# Tasks

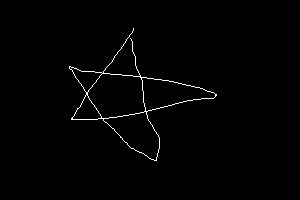
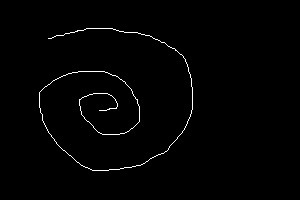
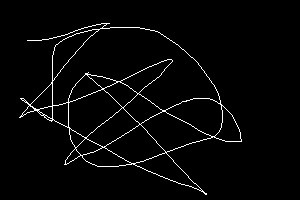
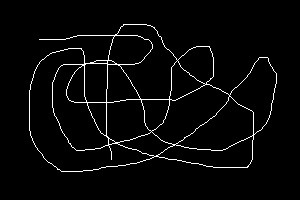
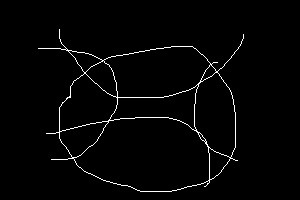
The *ELRO Digital Art Studio* (*ELRODAS*) is planning to create a mobile app (called *Digital Artist*) for art creation. The purpose of this mobile app is to allow general public with no formal art training to be able to create unique and beautiful artistic expressions on the go easily in digital forms. The core of this app is a set of algorithms that will convert a simple drawing from the user to a piece of digital art.

Recently, our *Computational Thinking* classes signed a

contract with *ELRODAS* to research the algorithms for the *Digital Artist* mobile app. Each project group is required to develop at least one algorithm to create the digital art. A panel from *ELRODAS* will visit us and judge on the algorithms. If any algorithm is picked by *ELRODAS,* it will be integrated into the *Digital Artist* app. At the same time, *ELRODAS* will donate *$1,000* to any charity organization specified by our school. In addition, the summer paid internship opportunity will be guaranteed for those project groups.

# Specification

***Input Files****:* Five black-and-white (white line/curve drawn on black background) input images will be provided. They represent the random drawings from users. The file format for these images is *PPM* file. The size of the files is *200 x 300 (rows x columns)*.The value of white color is *255* and the value of black color is *0*. The following are some examples:



***Output Files****:* Five output color images corresponding to the input images need to be created. The file format for these images is *PPM* file. The size of the files is *720 x 1280 (rows x columns)*.There will be 256 gray levels for each *red* (*R*), *green* (*G*),

|  |  |
| --- | --- |
| and *blue* (*B*) colors.  **Resource** |  |
| ***Paintbrush****:* | A paint program for Mac OS X. (http://sourceforge.net/projects/paintbrush/) |
| ***Image Converter****:* | An image to image converter. (http://www.sciweavers.org/free-online-image-converter) |
| ***PPM Format****:* | A ppm file format specification. (http://netpbm.sourceforge.net/doc/ppm.html) |
| ***Hough Transform****:* | A ppm file format specification. (http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm) |
| ***drawing1.ppm****:* | (https://drive.google.com/file/d/0B4er4OWY8aPMWVczcklXQ0t3aGc/view?usp=sharing&resourcekey=0-bMvDUSGvnU4tf1y38geOdQ) |
| ***drawing2.ppm****:* | (https://drive.google.com/file/d/0B4er4OWY8aPMVVlaTlVDYlpoOWM/view?usp=sharing&resourcekey=0-uSNguOM5D8wTX0aYtKqGDA) |
| ***drawing3.ppm****:* | (https://drive.google.com/file/d/0B4er4OWY8aPMVHUwZWRvWlR3Nzg/view?usp=sharing&resourcekey=0-jKbaJk-De8YFphPnwJRgOQ) |
| ***drawing4.ppm****:* | (https://drive.google.com/file/d/0B4er4OWY8aPMOXZYTi1kcmYyTHc/view?usp=sharing&resourcekey=0-idi3br4\_Hvm\_hp9h-GXCZg) |
| ***drawing5.ppm****:* | (https://drive.google.com/file/d/0B4er4OWY8aPMcnhOTlFydVFuLUU/view?usp=sharing&resourcekey=0-9QLiZGuyLf7F8522b7aOdQ) |

# Deliverable

1. A technical report in *Microsoft Word* format including (1) problem description, (2) explanation of mathematical functions and algorithms used in the program, (3) implementation *(attach computer programs)* of your project, and (4) the output images of your program.
2. Compressed project folder in ZIP file format. Five output images in JPG file format.
3. Project groups should be ready to demonstrate and explain their functions/algorithms/programs to the class.

**Problem Description:**

The goal of this project was to show how programming and mathematics can merge and create beautiful art that is both visually stunning and pleasing to the eye. To do this Mr. Lin used Hough Transform which is an technique used in computer vision to identify straight lines in a photo. What students tried to do was to be inspired by the newly learned Hough transform to enable themselves to create mathematical art. What this means is that the art created does not need to accurately represent a Hough transformation. We simply used the idea and procedures of a Hough transformation to generate new and unique art. To put it simply, we can do anything we want including changing all the parameters and all the functions being used. Thus, I wanted to create a website that allowed a great user experience for tuning these parameters while also enabling more people to explore programming, math, and art.

**Explanation Of Mathematical Functions:**

The mathematical ideas used in this project are the: Hough Transform, Polar coordinates, and Transformations. I will attempt to explain these ideas but since I do not know what I am doing, I would recommend lifting your eyes away from what I am about to write and read qualified articles about the listed topics. I have never officially learned these topics in depth, plus I have no idea what I am doing half the time. Please, proceed with caution.

Diagram

Description automatically generated

Figure - Credit: Mr. Lin

The Hough Transformation was/still is a computer vision technique used to find straight lines in an image. For example, in the image shown above (*Figure 1*) the original gray scale image was converted to white and black pixels, then through the Hough Transform (shown in the middle of the image), the computer was able to calculate the straight lines in the image through the pixel or points of the white and black image. The reason why it is wavy is because the Hough Transform was done in polar coordinates as computer cannot handle the normal *y = mx + b*  representation of a line. That is because computer can no store a potentially HUGE *b* value when it has a limited amount of storage. Long story short you basically turn points into lines. Then the lines all intersect, or at least closely intersect, because they share the same *m* and *b.* In this process, while in the Hough Space, it turns out these lines look very beautiful. Then we just used transformation to wrap the graph around the origin to create a wonderfully, beautiful, magnificent, awesome, flowery, interesting, *you get the point*, graph.

**Implementation:**

So how does it work? As previously explained, we took inspiration from the Hough Transform thus, it means if doesn’t really have to be accurate… it’s inspiration not real implementation. Here is what’s going on in the background.

There will be a random input image to use the Hough Transformation on. Alternatively, you can also submit you own image. Then a function will create a Hough Space of that image and out put it as a new image. The new image will also be colored to make it look even more pretty. In the Hough function you can tune all the parameters for fun. For example, instead of using *Cos* and *Sin* to calculate our radius how about we use *Tanh* with hmm… haha *Cosh*! Keep clicking random to generate random parameters and your basically done.

Please use my website. The code is also available on GitHub publicly and if you want to contribute, please just submit a pull request.

**My favorite Output Images**

