

Animatrix

New Mexico

Supercomputing Challenge

Final Report

April 2, 2018

200

Saturday Science and Math Academy

Members

Sarai Rankin

Dominique James

Teacher/Mentor

Eric Rankin

Summary

Smooth animation is a feat accomplished by few and coveted by many aspiring digital artists worldwide. While frame-by-frame animation can result in near-flawless video, it is time consuming and difficult. Computer generated animation software is expensive. To combat this issue, shows that are on weekly deadlines or are of slightly lesser quality cut frames and replaced them with distorted filler images to create the idea of movement and let the human brain fill in the rest of the action. A genius idea, but not one that applies to all parts of the body. The face, more specifically the mouth, can not always have a filler frame to imply motion, it moves too much, too often. The goal of Animatrix is to create a code that knows how the mouth moves while speaking, and can relay an animated mouth fluidly synced.

Problems

Rome was not built in a day, and Animatrix could not be coded in these past few months. Our project was so ambitious, it had to be broken into parts, especially for our team's novice at best ability to code. Animating by hand is slow, and programming a computer to do the same thing takes a lot of time and work as well. However, when we complete the entire code, Animatrix will solve the most time-consuming part of drawn cinematic art.

Solutions

A significant amount of our time on this project was dedicated solely to researching animation, lip syncing, audio recognition, and image recognition. The best part about our project is that most of the code needed already exists in one form or another. It was up to us to find the pieces and glue our code together like a model airplane. We decided on taking a new approach; Rather than trying to write the entire code this year, we would break Animatrix into achievable stepping stones that all work as standalone codes, but when finished, will fit together. For our first step, all we

needed to do was detect a face, narrow down the selected area to the mouth, and grab the image.

Discussion

The first few times we attempted this code, we succeeded with nearly complete accuracy -around 90%- with our main struggles in crowds and blurry images. Eventually, we created a face detector that could grab the selected images. Our plan for next year is to feed the images into a program and using Tensorflow, teach the computer association between the images and their corresponding sound bytes.

Conclusion

Initially, our code was intended to take an audio segment, and then transform that into an animated mouth portraying the words. However, we did have to narrow it down a bit due to lack of time, so our current code detects faces and mouths. Using helpful python interpreters and tutorials, we were able to get a more accurate and consistent detection. We gathered information about web cams and how certain programs use facial detection, and we had to figure out how to get it down to just the mouth.

Software

Python (Base)

OpenCV

<https://realpython.com/face-recognition-with-python/>

Acknowledgments

We'd like to thank our mentor, Eric Rankin, for teaching us the basics of coding and helping us into our first year of competition and also Naomi Rankin, for introducing us to supercomputing.