Mapping arbitrary graphs to equations using Discrete Fourier Transform. Using Turtle as a visualizer and animator.

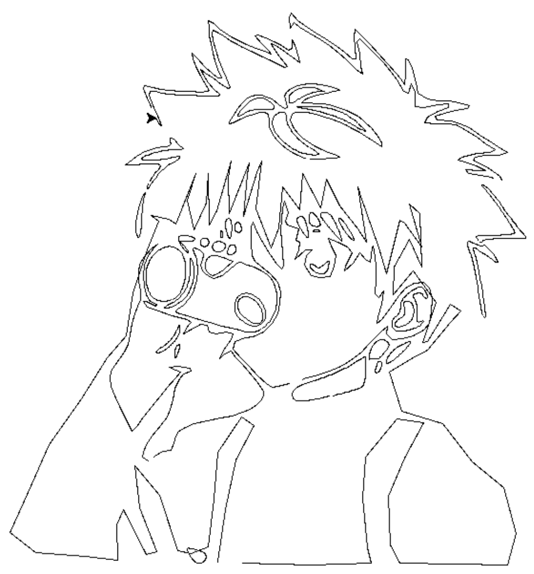
Run

`python3 main.py {hunter, scene, fairy}`
See `python3 main.py -h` for more details.

See SVG files in resources for sample of the original image.

Turtle and animation.

We use turtle as a visualizer for visualizing the entire curve. Every closed loop is a single curve. We can hence get a single equation for the entire curve. The curve is animated as if someone is drawing the curve by hand. This is generic and works with most SVG files with no additional overhead.



Representation of 2D coordinates.

We use a complex number (inbuilt support from python) for this.

Parsing subset of SVG

First, we create a partial SVG (see references) command parser able to parse absolute commands (L, C, M, V, H, Z).

The most interesting of them being the cubic bezier curve, which is a parametric curve taking values from 0-1.

$$\mathbf{B}(t)=(1-t)^3\mathbf{P}_0+3(1-t)^2t\mathbf{P}_1+3(1-t)t^2\mathbf{P}_2+t^3\mathbf{P}_3,\ 0\leq t\leq 1.$$

We then calculate values of different points in the curve, as well as the length of the curve.

This requires us to integrate as well as differentiate the bezier curve over the interval [0,1]

The derivative being the simple equation.

$$\mathbf{B}'(t)=3(1-t)^2(\mathbf{P}_1-\mathbf{P}_0)+6(1-t)t(\mathbf{P}_2-\mathbf{P}_1)+3t^2(\mathbf{P}_3-\mathbf{P}_2)\,.$$

Since calculating the integral cannot be done easily, we use a numerical approximation to calculate the integral.

Discrete Fourier Series

From the samplers created for the above curves, we generate a single equation for the entire curve.

We calculate the coefficients for the fourier series using the following formula. Integrating the equation is not simple (We could not find a simple integral. Therefore, we use a numerical approximation.

$$\begin{aligned}X_k&=\sum_{n=0}^{N-1}x_n\cdot e^{-\frac{i2\pi}{N}kn}\\&=\sum_{n=0}^{N-1}x_n\cdot\left[\cos\left(\frac{2\pi}{N}kn\right)-i\cdot\sin\left(\frac{2\pi}{N}kn\right)\right],\end{aligned}$$

Applications

Known Applications:

- Lossy data compression (JPEG for example uses the discrete cosine transform function to efficiently compress lossily)
- Fast multiplication algorithms using FFT.
- Spectral analysis
- Signal processing.
- See (Applications for more details)

References

Discrete Fourier Transform: In mathematics, Fourier analysis is the study of the way general functions may be represented or approximated by sums of simpler trigonometric functions.

Scalable Vector Graphics (SVG) is an Extensible Markup Language (XML)-based vector image format for two-dimensional graphics with support for interactivity and animation.

Bezier curves A Bézier curve is a parametric curve used in computer graphics and related fields.