Evaluating Feasibility of Machine Learning Techniques on the Vectorization Problem Domain

GROVE CITY COLLEGE
Because faith and freedom matter

Dr. Cory Boatright, Ryan Brothers, Jonathan Worobey

Abstract

Image vectorization is the process by which rasterized images are converted from row/column pixel format into a collection of mathematical curves and regions. This allows for the image to be resized without losing any sharpness or detail.

Other Approaches

Approaches to image vectorization range from simple to complex. The simples method is to recreate the image by hand using vector graphics editing software. This process is obviously time consuming, and requires existing production software.

Many commercial software applications, including Adobe's Photoshop, include support for vectorizing an image. These programs produce good results, but the algorithms often take more time to execute. In addition, the software packages are often expensive.

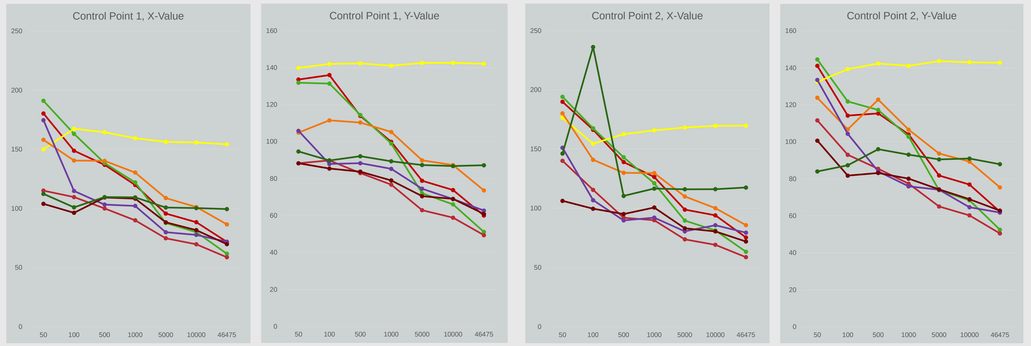
Machine Learning

A machine-learning based solution to this problem could provide a significant speed-up to the process. While the models themselves can take several hours to build, they can quickly produce results once constructed.

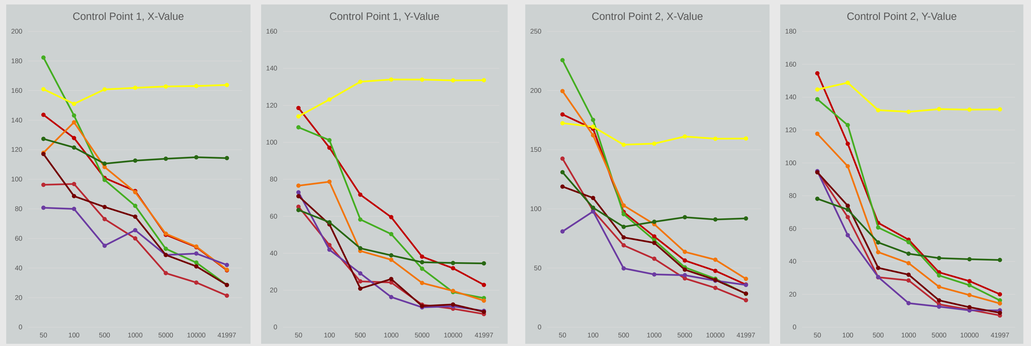
Our approach uses randomly generated Bezier Curves within certain parameters. These curves are used to build four different models; one for each X and Y value in the two middle control points. The start and end point of each curve are held fix to minimize the dimensionality of our problem.

Graphs

One Extrema



Two Extrema



Vertical axes: mean square error. Horizontal axes: size of training set. All training used 10-fold cross validation.

Conclusions / Future Work

- Ultimately, we found that some of the models worked much better than others. In particular, we found that, contrary to initial expectations, curves that have two extrema are much easier to fit than curves with only one extreme. We suspect that this is because small changes in the coordinates of curves with two extrema cause a much more predictable change in the curve itself. To this end, curves with only one extreme will require a different feature set.

Generating Data

- Our data was created in several parts. First, a curve interpreter program extracted the pixel data for each point in the rasterized input curve. This program also checked for curves that we would consider invalid. These include curves with loops, or which drew overlapping pixels during the rasterization process. We eliminated these as options because they would never show up in our data.
- Another program generated fixed-length feature vectors by selecting the extrema of each curve (based on Y values) and subdividing the curve until a set number of points was reached. This ensured that we get all of the important parts of the curve into our feature set.

Feature Sets

- Our feature set was obtained by selectively sampling values along the Bezier curve. Our sampling algorithm is designed to capture the most important aspects of the line.

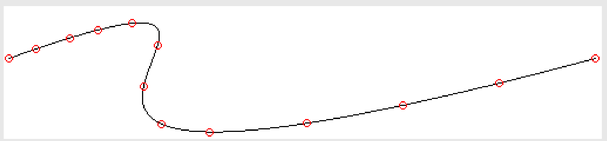


Figure 1: Sample points for a two extrema curve

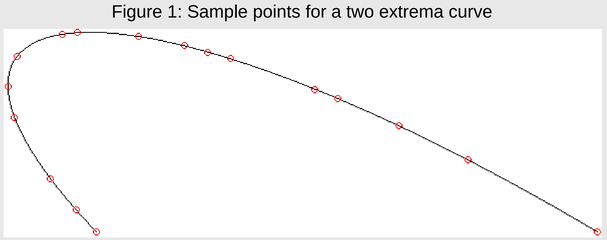


Figure 2: Sample points for a single extreme curve

Works Cited

- Freeman, Herbert. "On the encoding of arbitrary geometric configurations." *IRE Transactions on Electronic Computers* 2 (1961): 260-268.
- Pal, Sarbajit, Pankaj Ganguly, and P. K. Biswas. "Cubic Bézier approximation of a digitized curve." *Pattern recognition* 40.10 (2007): 2730-2741.