Semi-Structured Image Vectorization Through Neural Networks Based on Perception

S. F. Ballais[*]
Division of Natural Sciences and Mathematics
University of the Philippines Visayas
TACLOBAN COLLEGE

06 November 2018

Contents

[*] snballais@up.edu.ph

Abstract

We describe the CSG system, a secure online portal for the Computerized Registration System of the University of the Philippines Diliman. The portal incorporates a modified one-time password scheme that allows users to utilize their university DILNET credentials to login to any of the university computing resources. We designed the system to run on top of existing authentication mechanisms and is backward-compatible. Our preliminary tests indicate that the CSG system adds a layer of security to the current authentication process without any significant computational or storage overhead. (This paper describes a fictitious Computer Science Undergraduate Research Project at the University of the Philippines Diliman. This is only intended to show as a guide on how to write the first semester report.)

1 Introduction

Many images in typical use, from photographs to pixel art, are composed of pixels. Each pixel in an image contains a value that represents the colour it displays. These images are called, *raster images.* This type of image are ideal when creating images that are rich and detailed. However, raster images suffer from their inability to scale well. Scaling raster images will output grainy and distorted results, sometimes even blurry ones[2]. *Vector graphics* are an alternative to raster graphics, being resolution-independent and based on objects (often called *primitives*) and mathematical statements[3][1]. As a result, vector graphics generally tend to be easier to manipulate[3].

Semi-structured images can benefit from vectorization. As defined by Hoshyari, S., et.al., these images ”consists of distinctly coloured regions with piecewise continuous boundaries and visually pronounced corners”. Frequently, many digital artworks, specifically computer icons, comic book imagery, and simple graphic illustrations, are considered to be semi-structured images. Many legacy semi-structured images, typically those considered to be icons and clip arts, are stored in raster formats[4]. Additionally, in manual vectorization of images, artists often have to hand-trace each stroke and region using specialized software, such as Inkscape or Adobe Illustrator, which is a labourious task[6]. Automatic vectorization tools, such as Adobe Live Trace and Potrace, often demand more time rectifying the numerous tracing errors the tool produces by hand[6], and does not always produce results aligned with human perception[4]. Most vectorization algorithms are geared towards natural images (i.e. photographs) which frequently produce results not aligned with human expectations on artist-drawn imagery[4]. These cases call for a robust vectorization algorithm for semi-structured images that produces outputs that primarily align well with human-perceived results As noted by Hoshyari, S., et. al., human observers have a clearer mental image of the expected vector result from a raster semi-structured data. As such, vectorization results of semi-structured images require to be as close as possible to the mental image produced by humans[4].

Due to the fact that semi-structured images consists of distinct regions based on colour, the general flow of vectorizing semi-structured images can simply be boiled down into extracting those distinct regions and their boundaries, performing vectorization on those boundaries and appropriately colouring them, and combining the resulting vectors into a single vector object. The major technical challenge for this is vectorizing the region boundaries via piecewise free-form vector curves[4]. The most recent work on this topic is that of Hoshyari, S., et. al. Their work revolves primarily on accuracy and two key principles of Gestalt psychology: *simplicity*, and *continuity*. Simplicity simply states that human observers prefer simpler interpretations of geometric interpretations of raster images. Continuity states that human observers have a tendency to group stimuli into one continuous curves and patterns. Jagged raster boundaries are perceived to be piecewise smooth curves in their vector forms. In addition, observers would only mentally segment boundaries at a small set of discontinuous corners. Their work, however, becomes computationally expensive on larger inputs. Despite utilizing a corner classifier developed via machine learning as part of their vectorization, their classifier still produces results that require them to have an additional processing step, called *corner removal*, to produce better results.

I propose a method that is an improvement of the work done by Hoshyari, S. et. al. by improving the corner classifier in an attempt to render the corner removal step unnecessary, and utilizing machine learning to automatically draw the appropriate vector curves and lines, without relying on greedily computing and fitting the appropriate curves on the region boundaries. The latter, of which, is the method utilized in the framework devised by Hoshyari, S., et. al., which, as stated earlier, is computationally expensive at higher image resolutions.

2 Image Vectorization

The ability to provide high quality and clear resolution-independent images is the strong point of vector graphics. From any display, and any resolution, vector images still look crisp and no loss of details. Vectorization consists of various algorithms, heuristics, data structures, and geometric primitives that work together to generate a vector output from a raster input. Previous works on image vectorizations deal with various input images from pixel art to natural images. All of which have varying degrees of success in vectorization depending on the input. In the pursuit of supporting higher resolutions, previous works have also delved with upscaling raster images rather than converting to their vector versions.

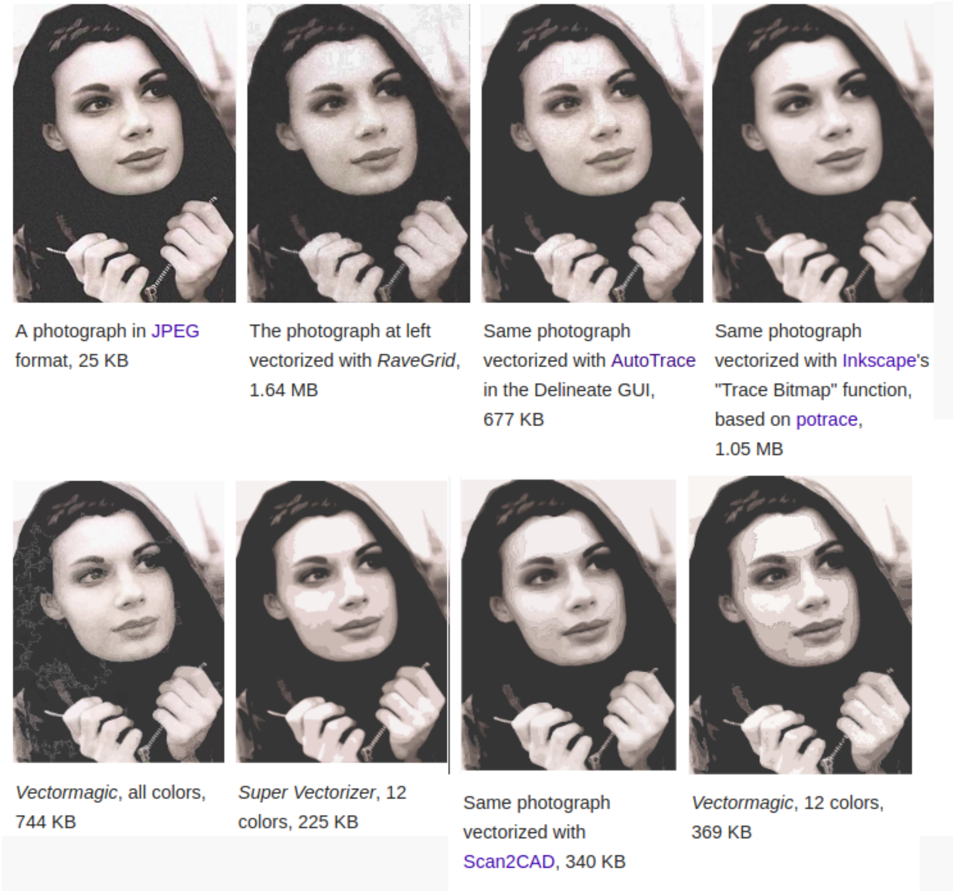
2.1 Previous Works

2.1.1 Image Upscaling

Image upscaling is the process of increasing the resolution of low-resolution raster images. The process utilizes single-image super resolution (SISR) algorithms to perform the transformation. There are many SISR algorithms that have been developed that focus on specific image classes such as faces, and graphics artwork. However, generic SISR algorithms exist as well to work for all kinds of images [8]. Examples of these generic SISR algorithms include Nearest-Neihbour, Lancosz, and Bicubic Interpolation [5]. As a side note, many raster editing software such as Adobe Photoshop and GIMP use bicubic interpolation or provided as an option (default in the aforementioned software packages) when transforming, i.e. resizing (in common terms), images. These generic SISR algorithms, being generic, assume nothing about the underlying data (aside from being band-limited), which results in blurring of edges and ring artifacts [5].

2.1.2 Natural Image Vectorization

One of the most common applications of vectorization is the vectorization of *natural images*. Natural images are essentially photographs. They contain continuous colour tones and fine details at the pixel level. Commercial packages employ natural image vectorization methods. Hoshyari, S. [4] notes that natural image vectorization methods addresses two separate challenging problems: image segmentation and segment boundary fitting. Most methods focus on the former and perform operations such as lossy moothing that is undesirable for inputs with perceptually well-defined boundaries designed by artists such as those in semi-structured images. Depending on the algorithm used, the vectorization results of natural images may or may not be satisfactory, as shown in Figure 1. Basing from the figure, the resulting outputs either may lose quality or details, or result in a larger file size.



A photograph in JPEG format, 25 KB

The photograph at left vectorized with *RaveGrid*, 1.64 MB

Same photograph vectorized with AutoTrace in the Delineate GUI, 677 KB

Same photograph vectorized with Inkscape's "Trace Bitmap" function, based on potrace, 1.05 MB

Vectormagic, all colors, 744 KB

Super Vectorizer, 12 colors, 225 KB

Same photograph vectorized with Scan2CAD, 340 KB

Vectormagic, 12 colors, 369 KB

Figure 1: Outputs of various vectorization algorithms in vectorizing a natural image. Acquired from Wikipedia *https //en.wikipedia.org/wiki/Image_tracing*.

Other image vectorization techniques that can handle natural images uses a geometric primitive called a *gradient mesh*. As defined by Sun, J., et. al. [7], Gradient meshes are geometric primitives that consist of topologically planar Ferguson patches with mesh-lines and has four boundaries, each of which consists of one or more cubic Bezier splines. Gradient meshes allow for multiple colours to be rendered with smooth transitions in a single primitive. This leads to image vectorizations that capture details and are close to their raster counterparts. Some image information such as fine details and highly textured regions may fail to be captured by gradient meshes [7]. Recent researches have improved gradient meshes, such as extending them with mathematically exact local refinement, branching capabilities, and sharp colour transitions, as was done in the work of Barendrecht, P., et. al. [3]. However, since we are dealing with vectorizing semi-structured images, which do not contain gradients, we will not be utilizing gradient meshes in this paper.

2.1.3 Vectorization of Artist-Generated Images

2.2 Accompanying Technologies

3 Problem Statement

State here, in detail what you will be working on.

You can follow the format below:

We are developing a *describe your project here*
in order to *describe the problem you are addressing*
because *describe the gap(s) that you have identified*.

You can revise the wording, you do not need to use the wording I have given. It is just a suggestion on the structure.

Here is a concrete example.

We are designing a disaster management system for flooding that allows for clustering, aggregation and filtering of data entries in order to improve the visualization and report generation of current flood hazard mapping systems. Some current solutions employ proprietary algorithms resulting in exorbitant system costs. Some of the more inexpensive options do not provide data checking and filtering, requiring manual intervention to check each and every data entry. Our solution provides for an accessible system that automatically checks each data entry and flags those that are possibly erroneous for manual verification.

You can also discuss here your constraints, i.e., what your thesis will not tackle.

This is how you should cite a work [?]. The entries will be in a separate file (we have named it mybib.tex in this example).

Here is another citation [?].

4 Proposed Methodology

Discuss here how you plan to develop the system. Discuss your design here.

What components are you using. What algorithms. What API libraries.

This is your WHOLE PLAN - not just what you have finished so far.

This is how you include a graphics file

This is how to reference the figure. See Figure ?? for the Xen logo image.

5 Preliminary Results

Discuss what you have done so far. This should be a subset of your Proposed Methodology.

References

[1] Raster vs vector. `https://vector-conversions.com/vectorizing/raster_vs_vector.html`, n.d.

[2] Raster vs. vector graphics: What's the difference? `https://pixellogo.com/blogs/pixellogo-blog/raster-vs-vector-graphics`, n.d.

[3] P. J. Barendrecht, M. Luinstra, J. Hogervorst, and J. Kosinka. Locally refinable gradient meshes supporting branching and sharp colour transitions. *The Visual Computer*, pages 949–960, 2018.

[4] S. Hoshyari, E. A. Dominici, A. Sheffer, N. Carr, Z. Wang, D. Ceylan, and I. Shen. Perception-driven semi-structured boundary vectorization. *ACM Transactions on Graphics*, 37(4):1–14, 2018.

[5] Johannes Kopf and Dani Lischinski. Depixelizing pixel art. *ACM Transactions on Graphics*, 30(4), 2011.

[6] R. Matheson and MIT News Office. Smoothing out sketches rough edges. http://news.mit.edu/2018/automated-vectorization-algorithm-animation-0911, September 11, 2018.

[7] Jian Sun, Lin Liang, Fang Wen, and Heung-Yeun Shum. Image vectorization using optimized gradient meshes. *ACM Transactions on Graphics*, 26(3).

[8] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 372–386, Cham, 2014. Springer International Publishing.