Abstract

Pixel artists rasterize vector shapes by hand to minimize artifacts at low resolutions and emphasize the aesthetics of visible pixels. We describe Superpixelator, an algorithm that automates this process by rasterizing vector line art at a low resolution pixel art style. Our technique successfully eliminates most rasterization artifacts and draws smoother curves. To draw shapes more effectively, we use optimization techniques to preserve shape properties such as symmetry, aspect ratio, and sharp angles. Our algorithm also supports “manual antialiasing,” the style of antialiasing used in pixel art. Professional pixel artists report that Superpixelator’s results are as good, or better, than hand-rasterized drawings by artists.

Our technique successfully eliminates most rasterization artifacts and draws smoother curves. To draw shapes more effectively, we use optimization techniques to preserve shape properties such as symmetry, aspect ratio, and sharp angles. Our algorithm also supports “manual antialiasing,” the style of antialiasing used in pixel art. Professional pixel artists report that Superpixelator’s results are as good, or better, than hand-rasterized drawings by artists.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Bitmap and framebuffer operations;

Keywords: rasterization, pixel art, antialiasing

1 Introduction

The conventional purpose of pixels is to provide small discrete samples of a continuous signal, ideally small enough for seamless reconstruction by the human visual system. In contrast, pixel art is a style of digital art that celebrates the aesthetics of visible pixels. The style developed out of necessity when early 8-bit graphics hardware had limited resolutions and colour palettes. It remains popular today in games, mobile applications, and graphic design – even though these limitations are imposed artificially.

Nearly all pixel art is constructed painstakingly pixel-by-pixel, with very little automation beyond flood fills. Automation is difficult because pixel art focuses on low resolution details which are not handled well by automatic drawing tools. By using pixel-based editors, artists are forced to work with individual pixels instead of addressing higher-level problems of shape, outlines, and composition.

A natural alternative is to create pixel art using vector-based illustration software. This approach enables artists to operate at a higher level of abstraction, making tasks like animating sprites for games much easier. But it fails to address the problem of representing vector art on a coarse pixel grid. Many standard rasterization algorithms are available, but these are optimized for infinitesimal pixels and produce unacceptable artifacts at low resolutions (see Section 3).

Our research focuses on pixelation [Inglis and Kaplan 2012], a special class of rasterization algorithms designed to respect the aesthetic conventions of pixel art at low resolutions. Every pixel counts in this context, and a pixelation algorithm must therefore consider the colour of every pixel carefully, taking into account its effect on its neighbours. Ultimately, pixelation should mimic the pixels that would be chosen by a human artist.

Pixel art is supported by a large online community where individuals regularly share their work, critique the work of others, and create tutorials. For example, a popular tutorial by Yu [2013] (reproduced in Figure 2) covers the cleaning, colouring, and shading steps involved in developing a finished piece of pixel artwork. By studying the work of pixel artists, and interacting with them directly, we can articulate the conventions they follow and ideally devise algorithms that embody those conventions.

In this paper we present Superpixelator, an algorithm for converting vector line art to pixel art. Our algorithm offers substantial improvements over the previous Pixelator algorithm [Inglis and Kaplan 2012] As a superset of Pixelator, Superpixelator generates smoother pixelated curves that preserve symmetry, yielding more faithful representations of geometric primitives. It also supports manual antialiasing, a form of antialiasing unique to pixel art that uses limited colours to draw clean, smooth outlines. We compare our results to hand-drawn pixel art and line art rasterized using existing software, and report on feedback from professional pixel artists. They report that shapes rasterized by Superpixelator are as good, or better, than those done hand rasterized by artists.