Artistic Vision: Painterly Rendering Using Computer Vision Techniques

Bruce Gooch  Greg Coombe  Peter Shirley
University of Utah  University of North Carolina at Chapel Hill  University of Utah

Abstract

We present a method that takes a raster image as input and produces a painting-like image composed of strokes rather than pixels. Our method works by first segmenting the image into features, finding the approximate medial axes of these features, and using the medial axes to guide brush stroke creation. System parameters may be interactively manipulated by a user to effect image segmentation, brush stroke characteristics, stroke size, and stroke frequency. This process creates images reminiscent of those contemporary representational painters whose work has an abstract or sketchy quality. Our software is available at http://www.cs.utah.edu/npr/ArtisticVision.


Keywords:  image moments, image processing, medial axis, non-photorealistic rendering, painting

1 Introduction

The art of painting relies on representation and abstraction. In representational painting, the abstraction occurs when the detail of real images is approximated with limited spatial resolution (brush strokes) and limited chromatic resolution (palette). Economy is a quality of many paintings, and refers to the use of only those brush strokes and colors needed to convey the essence of a scene. This notion of economy has been elusive for computer-painting algorithms. We explore an automated painting algorithm that attempts to achieve economy, particularly in its use of brush strokes.

There are two tasks involved in the creation of a digital painting. First is the creation of brush stroke positions, the second is the rendering of brush strokes. If the brush stroke positions are manually created by a user, then this is a classic paint program. If the brush stroke positions are computed algorithmically, then this is an automatic painting system. In either case, once the brush stroke geometry is known, the brush strokes must then be rendered, usually simulating the physical nature of paint and canvas [4, 17, 24].

We review digital painting strategies in Section 2 and give an overview of our algorithm in Section 3. The conversion from a single segment of an image to a set of planned brush strokes, which is the core of our contribution, is covered in Sections 4–8. We discuss possible extensions to our method in Section 9.

2 Background

Two basic approaches to digital painting are used in computer graphics. The first simulates the characteristics of an artistic medium such as paint on canvas. The second attempts to automatically create paintings by simulating the artistic process. These approaches can be combined because they deal with different aspects, one low-level and one high-level, of painting.

Work intended to simulate artistic media can be further divided into those which simulate the physics a particular medium, and those which just simulate the look. Strassmann [24] simulated the look of traditional sumi-e painting with polylines and a raster algorithm. Pham [17] augmented this algorithm using B-splines and offset curves instead of polylines to achieve smoother brush paths. Williams [27] provides a method of merging painting and sculpting by using the raster image as a height field.

Smith [22] points out that by using a scale-invariant primitive for a brush stroke, multi-resolution paintings can be made. Berman et al. [1] showed that multi-resolution painting methods are efficient in both speed and storage. Perlin and Velho [16] used multi-resolution procedural textures to create realistic detail at any scale.


While the works discussed above are concerned with the low-level interaction of pigment with paper or canvas, other authors aid a user in the creation of an art work, or automate the artistic process. Haeberli [8] built a paint system that re-samples a digital image based on a brush. Wong [28] built a system for charcoal drawing that prompts the user for input at critical stages of the artistic process. Gooch et al. [7] automatically generated technical illustrations from polygonal models of CAD parts. Meier [15] produced painterly animations using a particle system. Litwinowicz [14] produced impressionist-style video by re-sampling a digital image. Pham [17] augmented this algorithm using B-splines and offset curves instead of polylines to achieve smoother brush paths. Williams [27] provides a method of merging painting and sculpting by using the raster image as a height field.

Smith [22] points out that by using a scale-invariant primitive for a brush stroke, multi-resolution paintings can be made. Berman et al. [1] showed that multi-resolution painting methods are efficient in both speed and storage. Perlin and Velho [16] used multi-resolution procedural textures to create realistic detail at any scale.


While the works discussed above are concerned with the low-level interaction of pigment with paper or canvas, other authors aid a user in the creation of an art work, or automate the artistic process. Haeberli [8] built a paint system that re-samples a digital image based on a brush. Wong [28] built a system for charcoal drawing that prompts the user for input at critical stages of the artistic process. Gooch et al. [7] automatically generated technical illustrations from polygonal models of CAD parts. Meier [15] produced painterly animations using a particle system. Litwinowicz [14] produced impressionist-style video by re-sampling a digital image.