Paint By Numbers: Abstract Image Representations

Paul Haeberli

Silicon Graphics Computer Systems

ABSTRACT

Computer graphics research has concentrated on creating photo-realistic images of synthetic objects. These images communicate surface shading and curvature, as well as the depth relationships of objects in a scene. These renderings are traditionally represented by a rectangular array of pixels that tile the image plane.

As an alternative to photo-realism, it is possible to create abstract images using an ordered collection of brush strokes. These abstract images filter and refine visual information before it is presented to the viewer. By controlling the color, shape, size, and orientation of individual brush strokes, impressionistic paintings of computer generated or photographic images can easily be created.

CR Categories and Subject Descriptors: I.3.2 **[Computer Graphics]:** Picture/Image Generation - Display algorithms; I.3.6 **[Computer Graphics]:** Methodology and Techniques - Interaction techniques;

Additional Key Words and Phrases: Painting, image processing, abstract images.

Introduction

This paper is not about radiosity, anti-aliasing, or motion blur. Its not about making pictures more realistic. It is about creating interesting abstract representations of natural and synthetic scenes.

Graphic designers are experts at visual communication. In their work, graphic designers use photographic images when they are appropriate, but often chose to use more abstract images such as drawings or paintings. In many cases the designer must balance realism and effectiveness. Sometimes a realistic photographic image may be less effective than a stylized image.

In a panel discussion at Siggraph 1988 [Phillips 88] on the design of effective images Margret Hagen described the goal of the visual artist:

> "The goal of effective representational image making, whether you paint in oil or in numbers, is to select and manipulate visual information in order to direct the viewer's attention and determine the viewer's perception."

Impressionist painters use brush strokes to control light to simulate objects without modelling object detail explicitly. Only a few brush strokes are needed to represent a standing figure, a person's face at a distance, or a tree. By carefully selecting the location, color, size and direction of brush strokes, they control visual information to communicate abstract images to the viewer.

A Simple Painting Technique

Our goal is to take a synthetic or natural scene, and convert it into abstract impressionistic image. We want to make it easy for a user to interactively select and manipulate visual information to explore many different representations of a single source image. To do this we will point sample the source image at some set of brush stroke locations, and draw a synthetic brush stroke with the appropriate color.

A simple interactive program allows the user to operate on a source image. The basic interactive technique

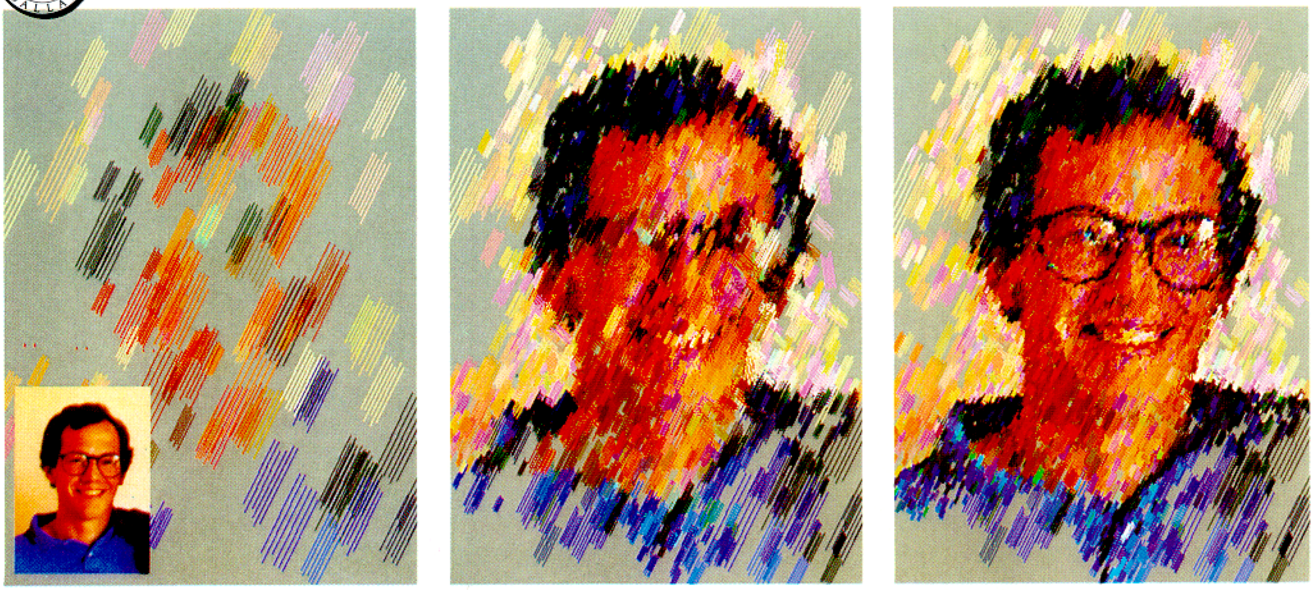

Figure 1. Creating a painting.

is to follow the cursor across the canvas, point sample the color of a stored image at the location of the cursor, and then paint a brush stroke of that color. Figure 1. shows a painting being created. The image that drives the painting is shown as an inset. The final painting is on the right. Using this process a user can easily explore different abstract representations of the source image.

In the work described here, a painting is an ordered list of brush strokes. Each stroke of the brush is described by a collection of attributes as shown in Figure 2.

Location - Position of the brush stroke.

Color - The RGB and Alpha color of the stroke.

Size - How big the stroke is.

Direction - Angle of the stroke in the painting.

Shape - The look of the brush stroke.

Figure 2. Brush stroke attributes

We will use the location, color, size, direction, and shape of brush strokes to communicate visual information to the viewer. The kinds of information we want to communicate are surface color, surface curvature, center of focus, and location of edges.

The program follows the position of the cursor across the canvas and point samples the source image to obtain a color. If a mouse button is down, an image of a brush stroke is drawn at a particular size and direction. In this way, visual information is selectively transferred from the source image to the canvas. By changing the size, direction and shape of brush strokes, many different representations of a single photographic image may easily be created.

One limitation in conventional paint programs is the time needed to pick a new color to paint with. [Lewis 84] described this as the "put-that-color-there" procedure that most paint programs use. Since the overhead in chosing a new color is high, it is very difficult to create a painting with a large number of different colors. The simple technique described escapes this problem by continuously sampling the color of the input image as painting proceeds.

Brush stroke locations are created in a stochastic distribution in the neighborhood of the cursor. This generates a scattering of brush stokes instead of a line of strokes when the cursor is moved in a straight line. This is an example of a simple interactive particle system [Reeves 85].

The size of brush strokes can be controlled in two ways. One technique uses the average speed of the cursor to control the brush size. If the user is painting quickly, the brush strokes will be made larger. When the user slows down, the brush size gets smaller. This makes it easy to first create a rough representation of the image with large bush strokes before adding finer detail with smaller brush strokes. Another technique allows the user to control the size of brush strokes using up and down arrow keys.

The orientation of brush strokes is controlled in several ways. One technique orients brush strokes with respect to the direction that the user moves the cursor. This is the tracking brush described in [Smith 79]. An alternative is to use a gesture of the mouse to set the current direction for brush strokes. To do this the user clicks down with the middle mouse button, moves in the direction brush strokes should flow, and releases the button. Other input devices could be used to interactively change the brush stoke size and orientation, as described in [Smith 79].

The shape of brush stokes can significantly influence the character of the final painting. The geometry of brush

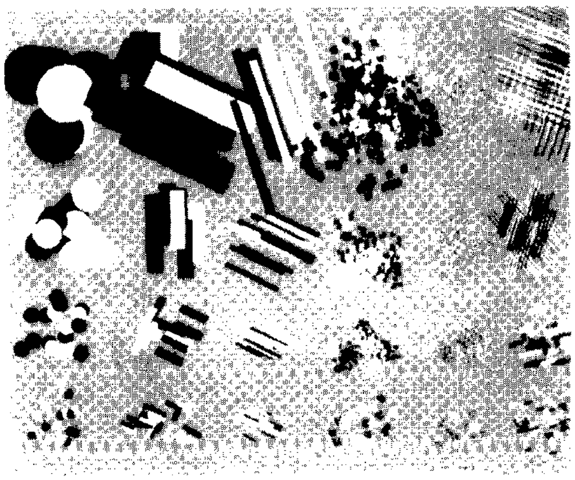

Figure 3. Several brush shapes.

strokes can be selected using a pop-up menu. The user can select among circles, rectangles, lines or scatterings of points and polygons for brush strokes, or they draw their own stroke if they want. In this implementation, all brush strokes are drawn using geometry instead of bitmaps - this makes scaling and rotating the brush very efficient on our hardware. Figure 3 shows several possible brush stroke shapes.

Three paintings of the same image are shown in Figure 4. The original image is shown inset. Diagonal brush strokes were used to create the painting on the left. In these paintings the amount of detail across the painting was modulated to direct the viewer. Noise was added to the original image to create a distribution of paint colors.

One interesting brush stroke geometry is a cone. When a group of cones are z-buffered onto the canvas, they create domains of color that share the image plane. The result is a pattern of color that has a cellular characteristic. These domains of color are Dirichlet domains [Preparata 85] derived from the location of the brush strokes. A painting made with z-buffered cones is shown in the center of Figure 4. By lingering with the cursor more detail can be exposed on any part of the canvas. A pointillist representation is shown on the right of Figure 4.

A description of the painting can be saved by interacting with a menu. A complete painting is an ordered set of brush stokes. A binary file format is used to store a stroke by stoke description of the painting. Figure 5 shows what a textual representation of a painting looks like.

```
painting with 9298 strokes
position        RGBA color          siz dir brush
0.447 0.541  241 128 173 255  attr:  30  89   5
0.444 0.531  220  57  35 255  attr:  30  89   5
0.441 0.524  172  29   1 255  attr:  30  89   5
0.444 0.553  230 100  75 255  attr:  30  89   5
0.447 0.526  220 112 162 255  attr:  30  93   5
0.456 0.554  245 189 137 255  attr:  30  93   5
0.503 0.522  245 183 237 255  attr:  31 107   5
0.479 0.545  228 141  92 255  attr:  31 101   5
0.498 0.517  246 181 230 255  attr:  31 103   5
```

Figure 5. A textual representation of a painting

Operations on Paintings

After a painting is saved, we can transform it using several paint processing programs. A painting renderer transforms a description of a painting into an RGB image. This can be used to create an extremely high resolution image of a painting.

A set of unary operators modify a single painting. All the brush strokes in a stored painting can be scaled up or down. Paintings can be altered by adding noise to brush stroke locations, color, size or direction. If wanted, each

Figure 4. Three paintings of one image.

brush stroke can be rotated by adding a constant to its direction. The order of brush strokes in a Painting can be sorted by color, direction, position, or size.

Binary painting operators use two paintings as input. It is easy to interpolate between or extrapolate beyond two paintings that have the same number of brush strokes. Paintings may be animated in this way. The brush strokes in two paintings can also be concatenated to overlay one painting on top of another.

Another tool lets us use an RGB image to modify the color, size, or direction of brush strokes in a stored painting.

The tools described above can be used to automatically roto-scope live action. To do this, we generate a rectangular array of brush locations, then add noise to their positions. Finally, RGB colors are assigned to each brush stroke in the painting by sampling one frame of live action.

Advanced Painting Techniques

Interesting effects can be created by using arbitrary images to control the brush direction across the canvas. Figure 6. shows a painting that was created by using a second image to control the brush direction. This painting consists of 9298 brush strokes.

When sampling a natural image, we start with only RGB information. Sometimes when painting, we might want to make brush strokes become aligned automatically with edges in the image. This can be done by generating an image that contains information about direction of the luminance gradient in the image [Kass 87]. Figure 7. shows a painting that was created using this technique. First the original image was converted to black and white and low pass filtered. Then the direction of the gradient of this image was used to control the brush stroke direction while painting. Notice how the brush strokes outline the shape of the head and flow along the collar of the shirt.

Another interesting technique is to blend the brush strokes onto the canvas. In Figure 8, a scanned in image of a real brush stroke is used. This brush stroke image was texture mapped onto the canvas to create the final painting.

Spice for images: Video-Sodium Glutamate

Before painting a synthetic or natural image it may be good to use a little spice (VSG) to accentuate the features we consider most important. These enhancements can be applied globally or locally to draw attention to particular parts of the painting.

When traditional artists make illustrations or paintings, many techniques are used. Often important edges in the scene are exaggerated. Where a dark area meets a light area the dark area is drawn slightly darker, and the light area is drawn slightly lighter. Some artists call this tech-

Figure 6. Using a second image to control brush stroke direction.

Figure 7. Using the gradient direction to control brush strokes.

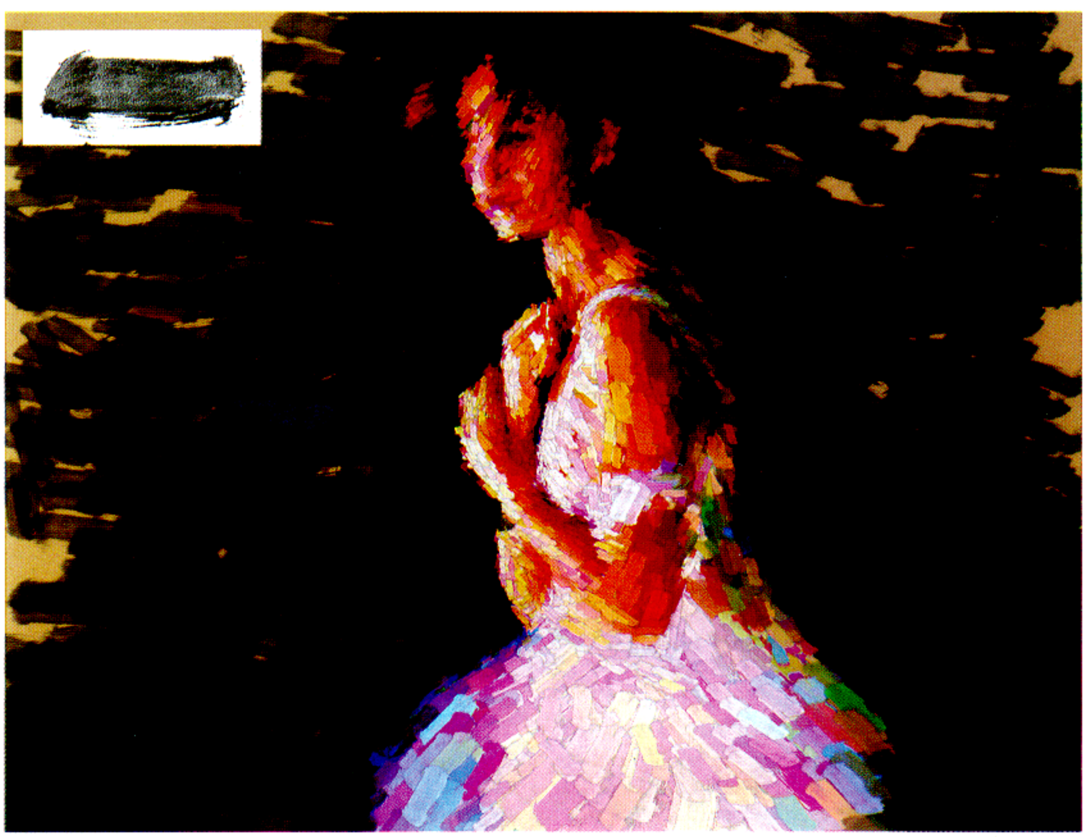

Figure 8. A texture mapped brush stroke.

nique "pushing" an edge. This technique can be used to make depth relationships between objects in the scene more explicit where they overlap [Porter 88].

To simulate "pushing edges" we can high pass filter the original image. High frequencies are enhanced by using unsharp masking. This is done as a two step process. First the original image is blurred by convolution. Next we create an enhanced image by extrapolating from the blurry image out beyond the original image.

To do this we define a linear interpolation operation on these two images such that a parameter of 1.0 gives us the blurred image and 0.0 gives the original image. To create an image with pushed edges, we make the parameter negative. As a result of this flat fields in the original image will remain unchanged, but edges are accentuated.

This process has two important variables; how blurry the image is made in the first step, and how much we extrapolate in the second step. Convolving with a 3 by 3 kernel, and making the parameter -0.5 will accentuate very high frequency detail and change the shading only very near edges. To make the shading change within 10 pixels of edges, a kernel with a width of more than 20 should be used. Some interesting effects can be created when the kernel is made very large - approaching 20% of the image diameter.

An artist may chose to enhance the richness of some colors in a scene. Sometimes the color is uniformly saturated throughout the scene, or only particular parts of the image are enhanced in this way.

If we want to increase the saturation of the image first we create a luminance image using a formula like this:

```
lum = 0.3*r + 0.59*g + 0.11*b.
```

Next, we extrapolate from the luminance image out beyond the original image. Achromatic parts of the image will remain unchanged, but all the colored parts of the image will be even more colorful, while preserving the same luminance. It is important that the original image be properly color balanced before this is done, otherwise improper colors for skin tones will become obvious.

When painting solid colored areas, artists may use a wide range of colors to communicate the color of a surface. This helps the viewer see a range of component colors in a surface that may be a single, flat color.

To add detail to regions of flat color, noise may be added to the image. When this is done, the final painting will have brush stokes with an interesting distribution of colors. This can make the final painting much more lively and interesting.

Most books on painting recommend sticking to a fairly limited palette of colors so as to achieve an overall harmony of color across the painting. The palettes used by the impressionists usually contained fewer than 12 colors [Callen 82]. These raw colors were mixed to create additional intermediate colors. Many beautiful paintings use remarkably few colors. With a restricted set of colors some brush strokes in the sky will closely match the color of brush strokes used to represent water. Restricting the number of different colors in a painting has the effect of unifying the painted image as a whole.

By quantifying colors in the source image, we can reduce the number of different colors in a scene without restricting the color gamut. If a sufficient amount of noise was added as discussed above, then no contouring will be visible.

Artists often cover the entire canvas with a wash of color before painting the image. The color of this background image can affect how the colors are perceived by the viewer if it is left exposed in some areas of the final painting. It has been noted by Michel-Eugene Chevreul [Smith 87] that having some proximity to gray makes all primary colors gain in brilliance and purity. Allowing the background wash color to be exposed throughout an image gives it a kind of unity and integrity.

Colors are sometimes used to provide depth cues. Colors in the range green (grass), cyan, and blue (sky) recede, while yellow, orange, red and magenta move to the foreground.

We can use many of the techniques above to enhance digital images before painting begins.

Sampling Geometry using Ray-Painting

These painting techniques can be used to create painted representations of synthetic 3D scenes as well. When sampling geometry, we have direct access to the color of each surface, its normal, and its depth. We can use the surface normal to control the direction of the brush strokes. This provides the viewer with valuable information about the orientation of surfaces. Figure 9. shows a raytraced scene, and a painting that uses surface normals to control the direction of brush strokes. To make these illustrations, the user interface of the paint program was attached to a raytracer, letting the user reveal the geometry by sampling it in real-time. Notice how the brush strokes appear to wrap around the sphere, the cone and the cylinder.

Approximating Images by Relaxation

An iterative relaxation technique may be used to create interesting paintings with remarkably brush strokes. The left side of Figure 10 shows a painting of 100 rectangular brush strokes that approximates an image of a seated man. The right side of Figure 10 shows a painting

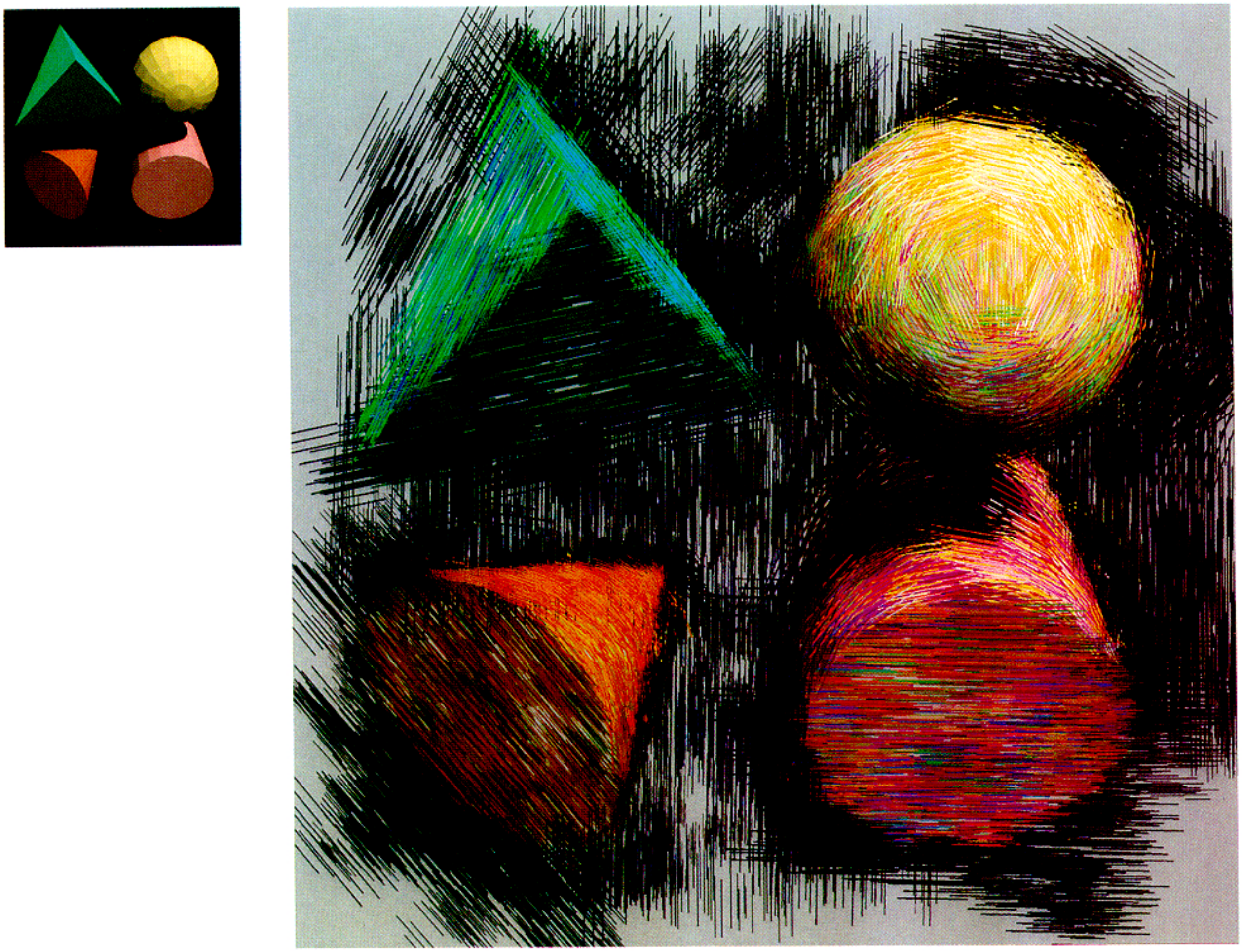

Figure 9. Sampling geometry using raytracing.

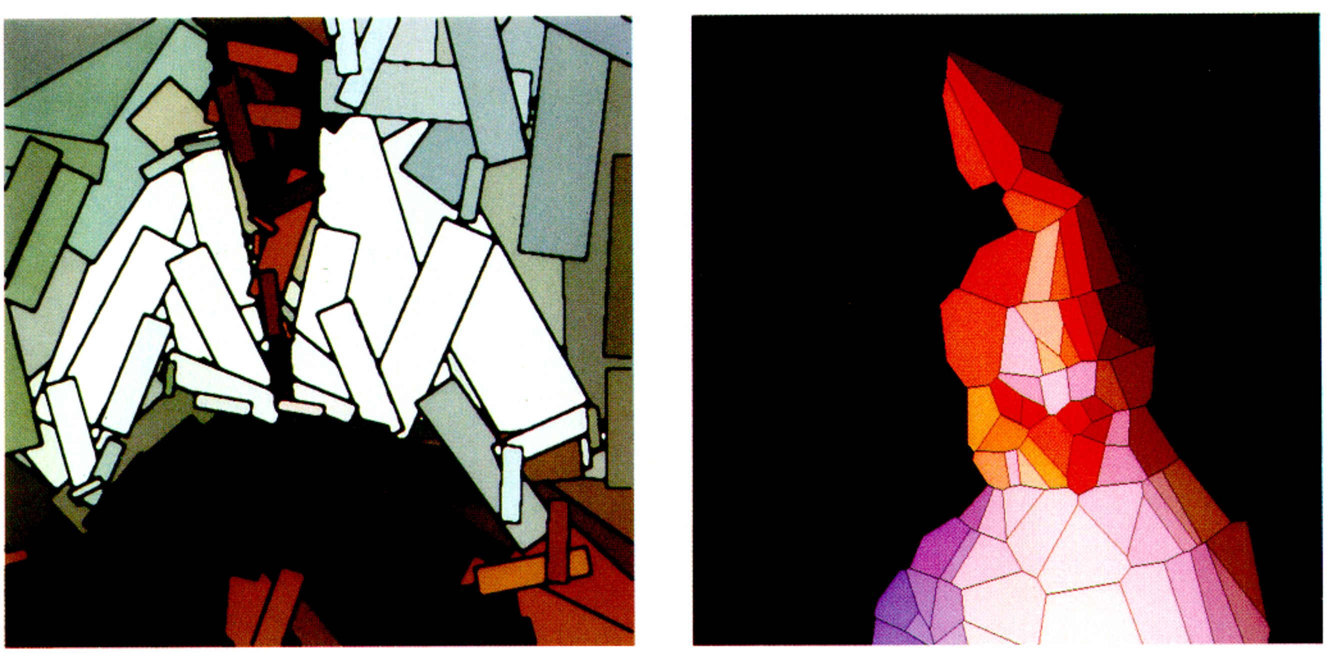

Figure 10. Using relaxation to create paintings automatically.

of 100 Dirichlet domains. These two paintings were created by stochastically perturbing the attributes of the brush strokes, while minimizing the root mean squared difference between the original image and the painted representation. This process ran for several hours before these images were saved.

Conclusions

We present several techniques for creating static and animated abstract images of photographed and synthetic scenes.

In this work, the goal is not to make photo-realistic images, but rather effective, interesting images that communicate. By interactively processing an image we can select and manipulate visual information to eliminate distracting detail, provide cues about surface orientation, and influence the viewer's perception of the subject.

It is natural that we want to continue to explore new painting techniques. A logical extension of this work would incorporate the texture synthesis work of [Lewis 84] and the brush modelling work of [Strassman 86].

Acknowledgements

I would like to thank Seth Teller for creating software to support rendering Dirichlet domains. I also appreciate the many helpful comments provided by the reviewers.

Some aspects of this work are patent pending.

References

[Callen 82] Anthea Callen, "Techniques of the Impressionists", 1982.

[Heckbert 87] Paul S. Heckbert, "Ray Tracing Brand Gelatin", Computer Graphics, 1987.

[Kass 87] Michael Kass and Andrew Witkin, "Analyzing Oriented Patterns", Computer Vision, Graphics, and Image Processing, 37, 1987.

[Lewis 84] John-Peter Lewis, "Texture Synthesis for Digital Painting", Computer Graphics, 1984.

[Phillips 88] Dick Phillips, "Siggraph '88 Panels Proceedings", 1988.

[Porter 88] Tom Porter and Sue Goodman, "Designer Primer", Charlse Scribner's Sons, 1988.

[Preparata 85] Franco P. Preparata and Michael Ian Shamos, "Computational Geometry", Springer-Verlag, 1985.

[Reeves 85] William T. Reeves and Ricki Blau, "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems", Computer Graphics, 1985.

[Smith 79] Alvy Ray Smith, "Table Paint", SIGGRAPH tutorial notes for "Two Dimensional Computer Animation, 1981.

[Smith 87] Ray Smith, "The Artists Handbook", Alfred A. Knopf, 1987.

[Strassman 86] Steve Strassman, "Hairy Brushes", Computer Graphics, 1986.