

Painterly Rendering with a Painter's Perspective

by

Caroline Dahllöf

B. S. Brown University, 1997

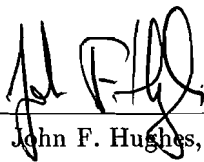
A thesis submitted in partial fulfillment of the
requirements for the Degree of Master of Science
in the Department of Computer Science at Brown University

Providence, Rhode Island

May 1999

This thesis by Caroline Dahllöf is accepted in its present form by
the Department of Computer Science as satisfying the thesis requirement
for the degree of Master of Science.

Date 4/13/99



John F. Hughes, Associate Professor

Approved by the Graduate Council

Date _____

Peder J. Estrup
Dean of the Graduate School and Research

Abstract

One objective of painterly rendering is to convert photographic images into renderings that look as if they have been painted by an artist. Previous research has drawn inspiration from a variety of artists, but we believe that further advances require an in depth exploration of a particular artist.

We describe a new painterly rendering algorithm modeled after Cézanne’s methodology. Our system takes an input image and user defined depth planes, and with these creates a “painting.” We use a combination of image processing techniques to mimic Cézanne’s process. The main steps are choosing a subject, defining depth planes, selecting a color palette, defining outlines, creating and applying strokes, and tracing the outlines.

Modeling our system after a painter’s methods provides a road map to take us from a subject to a painting. The painters we copy introduce us to new painting techniques, in our case the use of color planes. By combining images with depth planes, we are able to create paintings that have arbitrary complexity and depth. The concept of a painter’s palette is introduced, along with a technique for choosing the colors. A new brush model gives strokes the natural shape and texture variation typical of hand painted strokes.

Acknowledgments

I would like to thank my advisor John Hughes for his guidance and valuable feedback. I would also like to thank David Laidlaw, Barbara Meier, Anne Morgan Spalter, and Audrey Wong for looking at my renderings and giving feedback. Thanks to Greg D'Alesandre, Steven Dollins, Dan Gould, and Dave Powell for discussing different aspects of my thesis. Finally, I am greatly appreciative to Greg D'Alesandre, Steven Dollins, Dan Gould, Keith Hall, Loring Holden, Dave Powell, Tim Rowley, and Joe LaViola for helping me with my thesis writing.

Contents

List of Figures	vii
1 Introduction	1
2 Previous Work	4
3 A Painter's Method of Painting	6
3.1 Creating a Painting	6
3.2 Brush Stroke	7
4 Cézanne's Method of Painting	9
4.1 Subject	9
4.2 Palette	9
4.3 Brush Stroke	10
4.4 Painting the Canvas	10
5 Our Method of Painting	11
5.1 Choosing Subject and Determining Depth Planes	11
5.2 Selecting Color Palette	12
5.3 Determining Outlines	16
5.4 Determining Color Planes	16
5.5 Determining Orientation of Color Planes	17
5.6 The Painting Process	17
5.7 Determining Brush Stroke's Position	18
5.8 Determining Size of Brush Stroke	18
5.9 Creating and Applying Brush Strokes	19
5.10 Painting Outlines	20
6 Discussion and Results	21
7 Future Work	27

8 Conclusion	28
Bibliography	29
.1 APPENDIX	30

List of Figures

1.1	Cézanne’s still life paintings. (a) Still Life with Skull (Nature morte au crane). 1895-1900. Oil on canvas. The Barnes Foundation, Merion, PA USA. (b) Still Life with Compotier. 179-1882. Oil on canvas. Collection Mr. and Mrs. Rene Lecomet, Paris France.	2
1.2	Cézanne’s landscape paintings. (a) Le Mont Saint-Victoire vu de la carrière Bibemus. 1897. Oil on Canvas. Baltimore Museum of Art, Baltimore, MD USA. (b) Riverbanks (Bords d’une rivière). 1904-1905. Oil on canvas. Private Collection Switzerland . . .	3
5.1	(a) Input image (b) Depth planes (c) Outlines (d) Color planes (e) Painting	12
5.2	Working from Cézanne’s painting. (a) Cézanne’s painting (b) Treece’s painting (c) Our painting	14
5.3	(a) Input image (b) Initial palette (c) Final palette (d) Histogram of distances, the horizontal axis represents different colors and the vertical axis represent smallest distance to the palette	15
6.1	(a) Our painting (b) Input image	22
6.2	(a) Our painting (b) Input image	23
6.3	(a) Our painting (b) Input image	24
6.4	(a) Our painting (b) Input image	25
6.5	(a) Our painting (b) Input image	26

Chapter 1

Introduction

Art students are known to spend long hours copying paintings of famous artists to learn valuable techniques. They get ideas from these paintings and learn to create various styles and effects. The techniques learned can eventually be molded into their own style. The goal for painterly rendering is the same as for all painters: to create beautiful expressive paintings. In the same way that art students are inspired by other painters' techniques, we must also be inspired by them to develop new painterly algorithms. Painterly rendering tries to copy the end result of a human process (to paint) and should therefore take into account the steps a person takes in creating a painting. We copied Cézanne's method of painting by studying his work and works [10] [9] [17] [15] about his style.

We chose Cézanne largely because of his beautiful work but also because his paintings lacked minuscule detail. He used big brush strokes and planes of colors. He was more concerned with the impression of the whole painting, rather than with small details. In general, his small details were not smaller than a pixel in a large image.

Cézanne had two main periods. His first was heavily influenced by the Impressionistic movement and yet also very personal. He chose somber colors and motifs and used heavy brush work. His second period can be divided into three phases. In the first phase (early 1870's), his style was similar to impressionism in color, light, and brush work but with his own touch. He was able to create more solid objects by the use of color planes. In the late 1870's he entered the phase known as "constructive". His paintings had more brush directions and he used his color planes well to create space and volume. His last phase was more fluid, and he tended to introduce other mediums such as watercolor.

During his "constructive" phase Cézanne's paintings can best be characterized by his use of color planes and dark outlines. His brush work was very organized. He used more brush orientations than before but still a small fixed set. He utilized color planes that overlapped to create volume and space.

Looking through Cézanne's paintings it is apparent that his style evolved constantly. Even paintings within the same periods are often extremely different. Despite these changes in style we



Figure 1.1: Cézanne’s still life paintings. (a) Still Life with Skull (*Nature morte au crâne*). 1895-1900. Oil on canvas. The Barnes Foundation, Merion, PA USA. (b) Still Life with Compotier. 179-1882. Oil on canvas. Collection Mr. and Mrs. Rene Lecomte, Paris France.

are still able to recognize distinctive elements of his style in his paintings. We have based our algorithms on Cézanne’s “constructive” phase. The dominating style in these paintings is what we hope to recreate in our own work.

Cézanne never wrote down the exact steps he took to create a painting. Even if he had, it would only describe for a few paintings — all his works were created uniquely, which is part of what makes them so magnificent. To create our painting algorithm we had to determine general steps Cézanne took with most of his paintings. Some of the steps were altered and some were added simply because this is painterly rendering and not painting.

This paper will show that the steps we take to mimic a painter bring our renderings closer to the look of a real painting and further from the feel of the photographic input image. By copying all the steps that a specific painter takes from selecting a subject to creating a painting, we obtain a technique that works with the subject. Using the combination of photographic images and depth planes, we provide our system with complex Cézanne-like subjects and their depth information. The use of a color palette insures color coherency. The brush stroke’s color represents the generic color of a region and not the color of an individual pixel. From the image and these user-defined depth planes our system creates a “painting” without any further user interaction.

Our system uses minimal user interaction (input image and depth planes). If our system were heavily interactive, the user would have to be intimately familiar with Cézanne’s work to use it effectively. We therefore opted to make our system as user independent as possible.

In Section 3 we give background about how a painter might go about creating a painting. We will go into some depth about Cézanne’s method of painting and his style in Section 4. Our method of painting will be described in detail in Section 5.

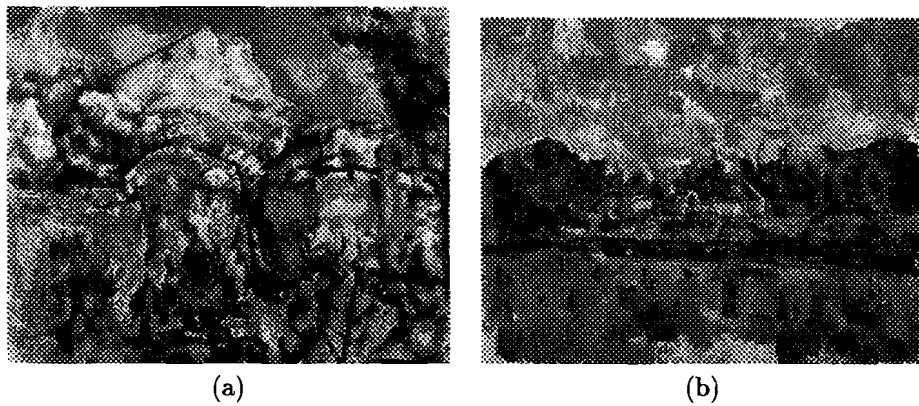


Figure 1.2: Cézanne's landscape paintings. (a) *Le Mont Saint-Victoire vu de la carrière Bibemus*. 1897. Oil on Canvas. Baltimore Museum of Art, Baltimore, MD USA. (b) *Riverbanks (Bords d'une rivière)*. 1904-1905. Oil on canvas. Private Collection Switzerland

Chapter 2

Previous Work

Researchers in computer graphics have taken three approaches to producing images that look as if they've been painted. These applications differ in what input they require. The first is a kind of image processing: a source image is used as a reference for producing an output image; the process is entirely 2 dimensional. The second is a 3D rendering scheme, in which a 3D model is the starting point. Intermediate 2D images are produced from the model, and these, sometimes together with the 3D model, are used to generate an output image. The third is simulation, in which the application of some sequence of brush strokes to the canvas are simulated at some level of detail. The boundaries between these approaches are blurry: simulation, at various levels of quality, can be applied either to the image-processing approach or to the 3D rendering approach. But there is a clear distinction between systems in which there is a 3D model and those in which there is not.

There has been a lot of research done in converting images into ones with a painterly feel. Haeberli [4] introduced an interactive painting system. The user specified where she would like to place a brush stroke. With this simple approach Haeberli was able to create very attractive and compelling results. Litwinowicz [8] created an automatic painting system in which he used a grid to determine the position of the strokes and applied this to video, developing a method to preserve inter-frame stroke coherence. Hertzman [5] introduced a new approach for an automatic painting system. He combined a lot of ideas from non-photo-realistic rendering (NPR) to create his system, using a modified grid method to place his strokes. All these systems chose their brush stroke color from the underlying image and take advantage of various of image processing techniques.

Painterly rendering that starts from 3D geometry can use more information about the scene than systems that work only from images. Such programs can use heuristics based on the 3D geometry which is not possible using just images. The advantage of working from images is that it is easier to get inputs for your system. Meier's [11] painting system worked from 3D geometry. The brush stroke position was specified by the position of particles on the surface of the geometry. Her paintings were artistically pleasing partly because she used many techniques that a real painter would when creating her paintings.

The brush strokes become very convincing when the reaction between the paint and canvas

is simulated. Strassmann [16] created beautiful Sumi-e illustrations by modeling the bristles of a brush. His brush model allowed him to take into account the pressure applied to the brush as well as the amount and distribution of ink on the brush. Curtis [1] created impressive brush strokes with watercolor. In his painting algorithm he took into account the wetness of the stroke, the amount of pigment, and the characteristics of the paper to which it was applied (a paint brush with the same characteristics applied to wet paper creates a different look than one applied to dry paper).

The work presented in this paper lies somewhere between the 2D and 3D approaches, in that the user starts with a 2D image, but must indicate various regions in the image that correspond to “depth planes” – portions of the image that lie at different depths from one another. Often this consists of identifying background, middle ground, and foreground, for example.

We have not only adopted ideas from painterly rendering techniques but also pen and ink techniques. Salisbury and Wong’s [13] created an algorithm that created beautiful pen and ink illustrations from images. The algorithm makes sure that more strokes are put down in the dark areas of the images than the light areas. [14] observed that in order to maintain the crispness of an image when scaled, the number of strokes should be the only parameter to increase. If the stroke size is increased the image may suffer some artifacts such as blockiness and grayness (in the case of a pen and ink illustrations).

Painted-image work also varies in one last way. Some systems are designed to allow a user to place strokes in any way s/he pleases and others are designed to generate a particular style. Our system is at one extreme; it’s designed to generate Cézanne-like images. But the hope is that the techniques described will be applicable to faithful reproduction of other styles.

Chapter 3

A Painter's Method of Painting

We believe that the best way to create a rendering with a hand-painted feel is to copy the method an artist uses to create her work. To do this successfully it is important to understand how a painter created her work and why she chose certain steps over others. Apart from understanding the painter's process it is also important to understand the medium and the tools. We will describe the painter in the next subsection and the medium and tools in the last subsection.

3.1 Creating a Painting

The steps a painter follows from a blank canvas to a work of art vary depending on the artist. Painters have their own style and therefore need their own steps. A painter will also vary and even drastically change her steps in order to achieve a different style for a specific painting. For some specific styles, though, there are some very general guidelines that one might use. The next few paragraphs will outline such a progression of steps that a realist painter might use.

The painter starts the painting process by selecting a subject, figuring out what this subject means to her, and what she wants the painting to say about the subject. She sets up the scene (or her view of it) in a way that enhances the objects and creates harmony in both space and color. She is also concerned about the lighting for the scene. The light gives color and creates shadows; the right combination is important.

After the subject is chosen, the painter starts selecting the colors to use that best represent what she wants to create. She carefully places them in a specific order onto her palette. She mixes these base colors to create other colors. This limits the artist to the base colors and combinations thereof to use as her palette. Keeping color limitations in mind, an artist will often take advantage of various color theories to change the impression of a color. For example putting a dark stroke next to a light stroke and therefore creating the illusion that the dark stroke darker and the light stroke lighter.

A painter could use a different colored canvas rather than the standard white. She can apply a color wash to her canvas. This will change the way she picks her colors and applies them. Many painters will try to maintain a color balance on their canvas. This balance will of course be influenced

by the initial canvas color.

Next, and most important, the brush must be chosen. The different brushes a painter uses to apply the paint create different effects. A soft brush creates almost no texture and blends the paint well; a hard brush creates a lot of texture and does very little blending. Bigger brushes might be used for areas of less importance in the painting and smaller brush strokes might be used in areas of great importance and detail. The background of a scene is often painted with bigger brush strokes, causing the viewer to focus on it less strongly.

The color of a stroke is chosen because it is similar to the color of the object that the painter is trying to paint. A painter might favor certain colors and will tend to shift the painted object's color towards those. Blue colors tend to recede and red colors advance, meaning that if there are many objects at the same depth, the red ones will seem closer. Painters often take advantage of this by giving their backgrounds bluish hues and the center objects reddish ones. This is one way to give the viewer depth clues about the painting.

A painter will start in many different ways. Often she will start by sketching out the scene on the canvas. This can be done with a pen or a brush. She will slowly start giving color to the objects, at first only using three values of each color (dark, medium, light) to define the light, medium, and dark areas of each object and later making the transition from one area to the next smoother and adding more values to try to capture more of the color variation. The background is usually painted before the foreground, insuring that the background strokes will not overlap the foreground ones. To do so creates flatness while the reverse can help give depth to the painting.

3.2 Brush Stroke

A painting is made up of brush strokes and although they are relatively small in comparison to the size of the canvas they play a large role in the whole impression of the painting. A real brush stroke's characteristics are determined by an enormous numbers of variables (ie. amount of paint, type of brush, application point, etc.). Looking at any group of brush strokes in a painting, it is obvious that each of them is unique, even in cases where there has been a conscious effort to make them as similar as possible.

Generically, the silhouette of a stroke can be described as a rectangle with round ends. The sides tend to be pretty straight with very small variations. The ends tend to be arcs. They can be very smooth or very jagged. There are large number of variables that determine the stroke's silhouette (ie. the speed, the pressure, the length, the type of brush used to apply the stroke).

The different brushes available to a painter create different effects. The bristles of a brush can give the stroke a 3D structure, which creates texture in the color of the stroke because of the shadows cast by parts of the structure. The amount and the viscosity of the paint determines the stroke's transparency.

Brush strokes are also influenced by other underlying brush strokes. The end result of a brush stroke applied on top of a dry stroke is very different than the result of applying it to a wet one.

In a wet application, the underlying stroke's color can become mixed into the new stroke or it can simply shine through the stroke.

Chapter 4

Cézanne's Method of Painting

Cézanne saw the world in “sensations of color” [15] and the role color plays in his work is obvious. Cézanne represented a “sensation” by a group of strokes with the same color. A group of strokes formed a color plane - these color planes were the main building blocks for Cézanne when creating a painting.

By studying Cézanne we get a deeper understanding of how to create a visually and artistically pleasing “painting”. He taught us a technique for going from a subject to a painting. He also showed the importance of choosing the right palette, the right subject and then emphasizing the right parts of the subject in the painting. In the next subsections we will describe the steps he took to create a painting.

4.1 Subject

Cézanne painted mostly landscape and still life. He was very concerned with light and space in his paintings. The harmony of the whole composition was more important to him than the correctness of the perspective. Cézanne was known to change the size of objects in order to achieve the right balance in his painting.

He would spend hours setting up his still life. He would flip and turn an object until he found the interesting parts to view. He would tip his plates to allow the viewer to see more of the inside of the plate. This gave the illusion that he had deformed the plate when it was painted.

For landscape he would actually deform the objects in order to give more importance to the more interesting parts of the scene. All this was done to achieve a global harmony in color and space between the objects on the canvas.

4.2 Palette

When examining Cézanne's paintings it can be observed that he used a similar palette for most of them. It is apparent in his landscapes that he was fond of saturated and vibrant colors. In

particular, his colors tended to shift towards yellow. This gave his paintings a feeling of warmth.

4.3 Brush Stroke

Cézanne was able to make each brush stroke stand out from one another and yet at the same time all the strokes in the same color plane blend together. His brush strokes were very uniform in size and shape and they had a lot of texture. It seems that Cézanne used the same sized brush for whole paintings, as there is no noticeable difference between the brush size for the background and the foreground. The shape of his strokes were very rectangular with slightly rounded ends.

4.4 Painting the Canvas

Cézanne started by outlining the scene on the canvas. He first used a pencil and retraced this with a dark bluish brush stroke. With these lines he made sure that the spatial relationships on the canvas were correct. The lines also served as a guideline for the painting process.

After the outlines were defined to his liking, he started putting down small planes of color all over the canvas. He started with neutral (unsaturated/pale) colors working from the darker toward the lighter side of an object. Adding color in some area meant adding more color in another area to insure harmonious colors in the painting. He would leave part of the canvas unpainted if this meant he would achieve the perfect harmony in color. Cézanne worked on the whole canvas at the same time never stopping to finish any one part.

Cézanne created volume through his use of color planes. The color planes were put down from cool dark to warm light colors. He allowed the warm planes to overlap the cool color planes, this gives solidity (firm shape) and volume to the objects in his scene.

Cézanne used a set of orientations in his paintings. These orientations were used to differentiate the depth planes and objects in his paintings and thereby create space. Cézanne did not always orient his strokes after the form of the object, they were mostly oriented according to his color planes. Brush strokes in the same color plane had the same orientation.

The outlines were used to give form and volume to his objects. This is necessary when dark and light colors fill the object without intent to define the form of the object. He almost never used continuous lines, most of the time he used “lost and found” lines, lines that would cease to exist when the background and the foreground colors merged together and reappear when they diverged.

As Cézanne constantly re-applied paint to his canvas, it was necessary to retrace the outlines constantly or else they would have been lost between the planes. To complete a painting, Cézanne would retrace all hidden outlines.

Chapter 5

Our Method of Painting

We have modeled our steps after the steps Cézanne took to create a painting. Some steps have been added, deleted, or modified because we are dealing with pixels and not paint.

Like most painters, we start by selecting a subject. We give the system an image of the subject and define the depth planes in the image (Section 5.1). From the image we select a color palette (Section 5.2) and determine the outlines of the objects (Section 5.3) in the scene. We then determine the color planes (Section 5.4) and their stroke orientations (Section 5.5). Starting from the background moving towards the foreground we paint the color planes from dark to light (Section 5.6), determining the location (Section 5.7), size (Section 5.8) and appearance (Section 5.9) of our brush strokes. When all the color planes have been applied, we paint the outlines of the objects with a dark colored brush stroke (Section 5.10).

5.1 Choosing Subject and Determining Depth Planes

The input image does not make the painting a success but it will determine where color goes in the “painting”. It is therefore important that the image has a good composition — there should be color as well as spatial harmony in the image.

We use subjects similar to Cézanne’s because we are trying to model his methods which work best with certain subjects. Not every subject lends itself well to a painting and some subjects will only lend themselves well to certain styles. A lot of photographs are quite beautiful but will not work with Cézanne’s style. For instance, the crisp details in an Ansel Adams photograph will be lost if painted in a style such as Cézanne’s.

It was important to us that we take the photograph of the subject from Cézanne’s perspective. The objects need to have the right colors and spatial composition. Our system cannot make an object have a more intense color (as a real artist might) or make it bigger to give it more importance. A photograph (and therefore our input image) must capture everything perfectly. We use one main light source for our scenes to distinguish between the dark, medium, and light parts of an object, which helps to define the shape of the object.

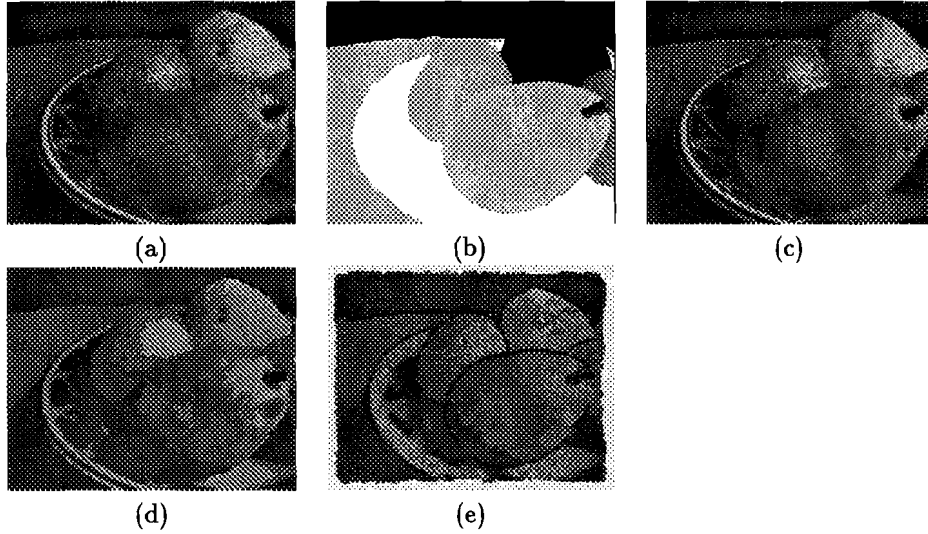


Figure 5.1: (a) Input image (b) Depth planes (c) Outlines (d) Color planes (e) Painting

An image only has 2D information about the scene; there is no information about the depth-ordering of objects between the background and the foreground. Many painters want to convey this depth information by allowing the foreground brush strokes to overlap the background strokes and by the use of color.

Depth was important to Cézanne and is therefore important to us. We have the user define depth planes in the image by identifying the regions of approxinably constant depth. To do so, the user fills regions in the same color. They define areas in the image which are at a similar depth. We are not interested in how far an object is from the viewer but just whether it is in front of or behind the other objects. With this information we are able to create “paintings” with depth. Figure 3b shows the depth planes defined for the input image (Figure 3a).

5.2 Selecting Color Palette

We are trying to create a “painting” using the steps of a painter. If a palette is not used (e.g. if the brush color is sampled from the input image) we are not mimicking the steps of a painter. A painter does not work with every color in a scene; she works with a few representative colors to create the painting. A side effect of not having a palette is that the “painting” looks like a photograph to which a filter has been applied because the colors sampled represent a very small area (pixel) rather than representing a larger area (brush strokes).

In order to recreate the feeling that Cézanne created in his paintings it is very important to select the right palette. This is shown in Figure 4. Both Dan Treece’s rendering (Figure 4b) and our rendering (Figure 4c) used Cezanne’s painting (Figure 4a) as the input image and therefore used his colors.

Cézanne has three distinct color regions in his paintings (red, blue, green). His outlines tend to



(a)



(b)



(c)

Figure 5.2: Working from Cézanne’s painting. (a) Cézanne’s painting (b) Treece’s painting (c) Our painting

be a dark color and he usually uses some light strokes. With this in mind, we created our palette to include groups of red, blue, green, black and white. It is clear from his work that he used a dark, medium and light color within each of the color groups, we concluded that using the few colors that span from dark to light of each group resulted in a palette that fit the color range of the input images well.

To select the average hue color value from the input image, we used the k -means algorithm [7], which finds a pre-defined number of clusters in a space. From the cluster a center point can be determined which represents the average value of that cluster. In broad terms, the k -means algorithm works as follows: it takes the number of cluster, the starting center point of each clusters, and the values in the space. It finds the center point closest to each value in the cluster. After all values are assigned a center, it calculates the new center point of the clusters by taking the average value of each cluster. This process is repeated with the new centers until the center points are fairly stable.

In our case the space was a color space with one entry for each input image pixel. The clusters represented groups of similar colors. The center point of a cluster is the average color of the cluster. Specifically we have mapped all the colors from the input image into CIE LUV [2] color space. LUV is a perceptually uniform color space, meaning that moving of a color the same distance anywhere in the space will produce the same color change. This is important because the k -means algorithm uses distance in determining clusters.

The k -means algorithm is very sensitive to the starting points of the clusters. We divided the colors of the image into five bins (red, blue, green, white, black) using HSV space (see Appendix).

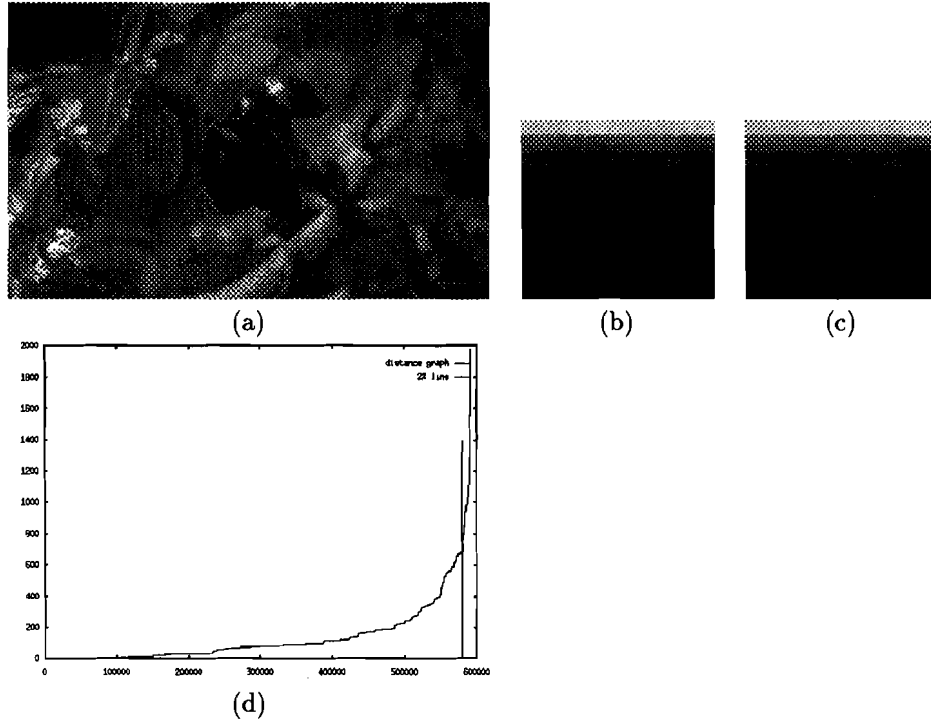


Figure 5.3: (a) Input image (b) Initial palette (c) Final palette (d) Histogram of distances, the horizontal axis represents different colors and the vertical axis represent smallest distance to the palette

We take the average color of each bin in LUV. This gives us good starting point that closely reflects the colors in the image.

Using k -means we first find the center point for five clusters (red, green, blue, black, and white). For each iteration we recalculate a new center point for the red, blue, and green cluster but not for the white and black cluster. This helps insure that we get dark and light colors.

For each of the five clusters, we find four center points of sub clusters — ranging from dark to light color. Again we use HSV space to divide each cluster into four bins (see Appendix) and make the average color of the bin be a starting point for the cluster, then apply k -means.

We have now selected twenty colors for our palette. To insure that the whole color range of the input image is represented correctly we add three more colors to our palette. These colors are the colors that are worst represented by the current palette. We calculate the smallest distance between the palette colors and the input image's colors. We sort the colors by this distance and take the 2% of the colors that are furthest away from the palette. We got this value from graphing the distances and observing that around 2% there is a big increase in the distance (Figure 5d). This value can be better chosen by letting the user determine the knee for each image. We take these 2% and divide them equally in three bins. We apply the k -means to the bins and use the average color of the bins for the initial starting point.

While we could have picked any number of colors for our palette, we found that twenty-three

colors gave us enough color variation while still producing some blockiness, which is important for creating Cézanne’s color planes.

After we have created the initial palette (Figure 5b), we shift towards yellow to obtain the final palette (Figure 5c); to mimic Cézanne’s palette. To do this, we convert the palette into HSV [2] color space. HSV space allows us to modify the hue while keeping the saturation and value of the color fixed. Colors closer to yellow are changed more than ones farther away (see Appendix).

5.3 Determining Outlines

Cézanne began by defining the objects’ outlines and main groups of color on the canvas and we start the same way. We are interested in finding areas with the same color. We convert the input image into an image with five colors (black, white, red, green, blue) by using our HSV bin technique(see Appendix).

A *blockfilter* is used to remove non-frequently found values by replacing them with the most frequently found value in the area. The filter counts how many times each of the different values appears in the area. The center point is replaced with the most frequent found value. This filter lends itself well if an array contains just a few values — say red, white, and black pixels — a block filter will tend to remove isolated pixels and convert them to the color of their surrounding.

We pass a 5x5 block filter over the image to solidify the areas. The image is then scanned for small (3x3) islands of color, which are replaced with the color of the surrounding area (see Appendix). A Sobel filter [6] is then used to find the edges of the color regions. The depth plane information is used to find outlines for objects that were overlapping but have the same color. Each outline is associated to a depth plane; if there is more than one possibility it is of determining color planes the closest depth plane. The outlines are used to insure that the objects keep their shape as more paint is applied (Figure 3c). At the end, these outlines are retraced with a dark brush to visually strengthen the object boundaries (Section 5.10).

5.4 Determining Color Planes

Cézanne’s color planes usually represented areas of similar color but could also be totally abstract. Our color planes represent an area of similar color (Figure 3d). To determine color planes we try to find areas that are similar to the colors in our palette. These areas are allowed to overlap each other to smooth the transition from one color plane to the next (see Appendix).

We start the process by determining the shortest distance *sip* between the input image’s color and the palette’s color. We also find smallest distance for each palette color between itself and another color in the palette *spp*. We order the palette colors from dark to light using LUV space ¹.

¹In the beginning we used the "value" variable of HSV, but this produced some artifacts. We switched to LUV space to get a perceptually correct space. We tried to order them by their distance to white (100, 0, 0). But this happens to be very close to black (0, 0, 0), and also produced artifacts.

We order them by the L value, 100 being light and 0 being dark. We create a color plane map *cpm* for each of the palette's color (23 maps) which represent where the color plane located.

```
for p=darkest to p=lightest
  for i=0 to i=width
    for j=0 to j=height
      if(dist(p, I(i,j))<(sip+spp*tolerance))
        cpm(i, j)=TRUE
```

$I(i, j)$ is the input color at pixel (i, j) . p is a specific palette's color. The tolerance value depends on the darkness of the palette's color — the darker the color the bigger the tolerance. All the calculations were done in LUV space.

Finally, to solidify the areas marked on the map we pass a 11x11 block filter over it. The twenty-three resulting color plane maps are stored and used in the painting process (Section 5.6).

5.5 Determining Orientation of Color Planes

We wanted each color plane to have a uniform brush stroke orientation as Cézanne's do. The actual orientation of the color plane is not nearly as important as the strokes having the same orientation.

We calculate the intensity map from the input image [2]. The intensity map is blurred using a filter to reduce the noise (see Appendix). For each entry of the intensity map we sum the difference of the value between the current entry and five entries above and below it, taking into account the distance in entries between the entries. We do the same in the horizontal direction. This gives us two vectors, one in the horizontal and another in the vertical direction. A greater magnitude means a greater rate of change in that direction. After these two vectors are calculated, the average change in the vertical and horizontal direction are calculated by averaging the five vectors around each entry. Doing all these calculations insures that the two vectors do not represent noise.

We add these two average vectors together and calculate the orientation and magnitude of a final vector that indicates the direction of greatest change — in a sense, a filtered gradient of the intensity. These orientation vectors are then “snapped” to a one of a few fixed orientations (see Appendix) to make the brush work more organized in the manner of Cézanne's.

For each of the twenty-three color plane maps, we assign the corresponding orientation to each marked map entry (color plane). Now we take each color plane map with its orientation and pass a block filter over its orientation. The block filter will only consider the orientations of the current map (see Appendix). These orientations are stored in the color plane map and are used in (Section 5.6).

5.6 The Painting Process

Our painting method is a bit different from Cézanne's. Instead of “bringing the whole painting forward” at the same time (i.e. ,instead of working each piece simultaneously, we paint our canvas

by starting from the background depth plane and moving forward. Each plane is finished before moving to the next. We need not follow Cézanne’s method of painting in this regard because we are not concerned about the color balance during the painting process. This is definitely something that we hope to do in our future work.

We start painting the dark color planes and move towards the light color planes. To paint a color plane we first determine the position of the stroke (Section 5.7). Knowing the position we can determine the orientation of the stroke (Section 5.5) and then the size of the stroke (Section 5.8). We then create the brush stroke and apply it to the canvas (Section 5.9). We continue this process until the color plane is painted and then move on to the next color plane.

5.7 Determining Brush Stroke’s Position

To insure that the canvas is completely covered with brush strokes yet not overcrowded we use a similar method to Salisbury et al. [13]. In the Salisbury technique, placing a stroke marks an area around the center of the stroke as “done.” In this marked area no other strokes can have their centers. By varying the size of the area one can change how close one allows strokes to be.

For each color plane we take all the pixels in the input image associated with it. We order them from dark to light, using the *value* variable in HSV space. We start placing strokes in the dark areas of each color plane and move toward the light part. Each time a stroke is painted, an area around its center is marked. The size (area) of the marked area is $3/5$ of the size of the stroke. This size gave us complete coverage of the canvas. The size of the marked area depends on how rectangular the shape of the brush stroke is, and its opacity. The more opaque and rectangular shaped the stroke is the less area is needed to be marked. There is no real formula for this ratio it depends on the look that is desired.

5.8 Determining Size of Brush Stroke

To mirror Cézanne’s brush sizes we tried to vary our stroke size as little as possible. There were some exceptions to this. The brush strokes in the background are slightly larger than the ones in the foreground. This gives the background areas less focus. Also, we made darker strokes bigger than lighter ones. Lighter strokes often represent highlights (people tend to notice lighter strokes more readily). We made these strokes smaller so the viewer would not be overwhelmed by large light strokes.

The color and depth planes determine the maximum size of the brush stroke. After the position of the brush stroke is determined we calculate the maximum size the stroke can have at that specific position: brush strokes are not allowed to cross outlines, and this becomes the limiting factor for the size of the stroke (see Appendix). If we allowed the strokes to cross outlines, the shapes of objects would be changed and the outlines would no longer represent the silhouettes.

5.9 Creating and Applying Brush Strokes

Our brush model aims to create brush strokes that are as varied as possible without actually simulating paint directly. We build our model on top of [4] and [11] brush models. Our brush model has the following variables; color, orientation, size, shape, position, texture and transparency. Each stroke gets its color and orientation from the color planes and its size from the color planes and the depth planes together with the outlines.

To create the shape of a stroke we used a summation of a few sine functions — a “poor-man’s Fourier series” [3]. We created four functions, one for each side and one for each end of the stroke. We have one sine function that gives the overall shape of function and then many small high frequency sine function that add noise and gives the shape irregularity which is found in real brush strokes. For the side of the stroke the main function is pretty straight, but may lean towards one side or the other. The noise functions will vary the main function by one or two pixels in larger bunches. For the ends we use an arc for the main function which varies with the width of the stroke. We add small very high frequency functions to the arc. They will move the main function by four or five pixels in small bunches, but sometimes move it only one pixel. We have different functions for each of the sides and ends of the stroke that we choose at random. To model Cézanne’s brush strokes we used a rounded rectangular shape for the main function to which we add noise functions.

The transparency and the texture in our brush model are obtained by scanning a canvas. We create this by first covering the canvas with a dark hue of a color. We let this layer dry and then apply another layer of the same color but a lighter hue. The last layer is supposed to convey texture, so we mix our paint with thickener to insure texture. The darker hue is obtained by mixing the lighter hue with black. When we paint our canvas we make sure that all brush strokes have the same orientation and that they blend together (we cannot see the shape of each brush stroke, only the texture). By painting the canvas in two layers first with a dark hue and then with a light hue, we make it easier to obtain and model the transparency and texture.

Looking at the scanned canvas and knowing that we painted light on dark we can draw the conclusion that the dark regions represent high transparency and light regions represent low transparency. We convert the scanned canvas into HSV space, and using the value axis of the space we determine the transparency. We store these values in a transparency mask. Cézanne’s strokes have some transparency but are mostly opaque. To make sure that the transparency values are close to opaque (1) we process the values obtained from the scanned canvas. We find the largest value and map that to 1 and then for all the other values we do: $\text{new value} = \text{old value} / \text{largest value}$.

We simulate texture in our brush model by changing the stroke’s color with the color variation of the scanned canvas. When we do so, we bias the stroke’s color towards the scanned canvas color. To solve this problem we use four canvases: red, blue, green, and gray. Reddish strokes use the red canvas, bluish strokes use the blue canvas, greenish stroke use the green canvas, grayish, white, and black strokes uses the gray canvas.

We supply our brush model with the color c , width w , height h , position p , and orientation o of the brush stroke. We start by determining which canvas to use with our HSV bin technique (see

Appendix). Then we select a random area A from the transparency mask with the same size as the stroke, and the corresponding area T from the scanned canvas. The texture of the scanned canvas is along the vertical direction. We calculate the average color avg of the selected area of the scanned canvas in LUV space. Using sine functions we create the shape of the stroke and a stroke map S , that is 1 inside the stroke and 0 outside. We add the color to the stroke map in the following way:

```
for i=0 to i=width do
  for j=0 to j=height do
    tmp=T(i,j)-avg /*calc color variation*/
    c=c+tmp /*add variation to color*/
    S(i, j)=S(i, j)*c
```

We rotate T and S by ϕ using three shears [12]. We apply our stroke at p on the canvas using T .

5.10 Painting Outlines

We want to draw smooth, slowly changing outlines. Using the outlines from our edge detector, we calculate the orientation at each point using a simple algorithm. We place a 21x21 square around a point of the outline and find the points where the outline intersects the perimeter of the square; we assign to the center point the orientation of a line between these points. After all the points of all outlines have been assigned an orientation, a block filter is passed over it to smooth out the curves. To paint the outline we use a brush of a random small width with a random dark color from our palette. We pick a point in the outline and make it the center point of the stroke that we paint. We mark an area around the selected point making sure that another center point cannot be created in that area. This method of center point marking is based on [13] and insures that the strokes are not spaced too closely together.

Chapter 6

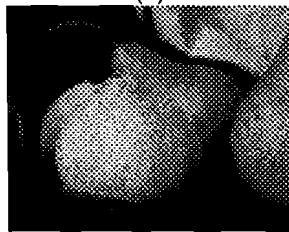
Discussion and Results

Our techniques have improved on painterly rendering with its new approach for creating the algorithm based on a particular painter, its brush model, and its concern about accurate modeling the whole painting process of a real artist. But it has only scratched the surface of the problem of producing natural looking paintings. There are still many limitations, mainly due to the lack of complete information about the scene.

- The system does not know what it is painting and therefore paints all objects in the scene in the same manner. A painter will typically paint water and a rock in different ways.
- We have created depth in our painting by user defined depth planes. Depth planes work well with scenes where there is a clear distinction between the foreground and the background objects. They tend to fail with scenes where this distinction is unclear.
- When selecting a palette our system gives all the colors in the image equal importance. Often a painter will pick more colors for the objects in focus than the background. This could be addressed by letting the user interact with the palette-selection part of the system.
- Our edge detection uses a generic method to determine the outlines. It does not take into account the balance of the images when selecting the edges. In some scenes it is desirable to highlight vague lines and in others the desire is to reemphasize sharp lines.
- Our brush model gives a more natural feeling with its emphasis on texture and generated silhouettes, but falls short when compared to a real brush stroke whose characteristics are determined by many more variables.
- Our brush size is determined by the depth plane and its color, and brush strokes are limited by the outlines. Another limiting factor should be the color planes: a brush stroke should stay inside of its color plane. We partially address this by making our block filter the same size as the brush stroke.



(a)



(b)

Figure 6.1: (a) Our painting (b) Input image



(a)

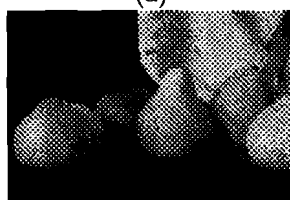


(b)

Figure 6.2: (a) Our painting (b) Input image

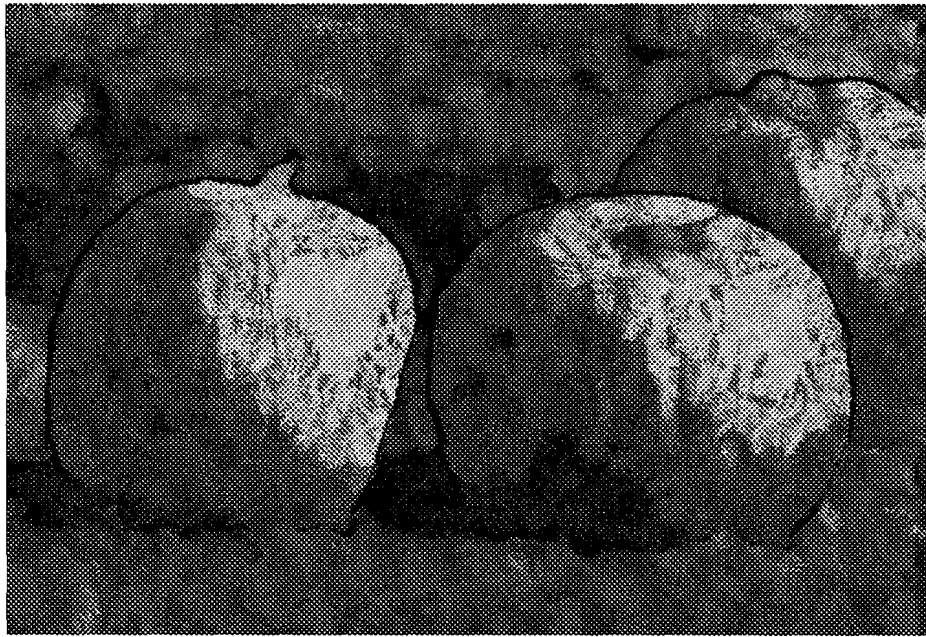


(a)

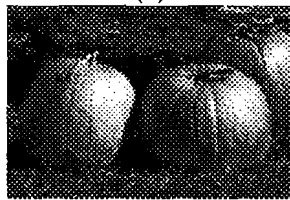


(b)

Figure 6.3: (a) Our painting (b) Input image

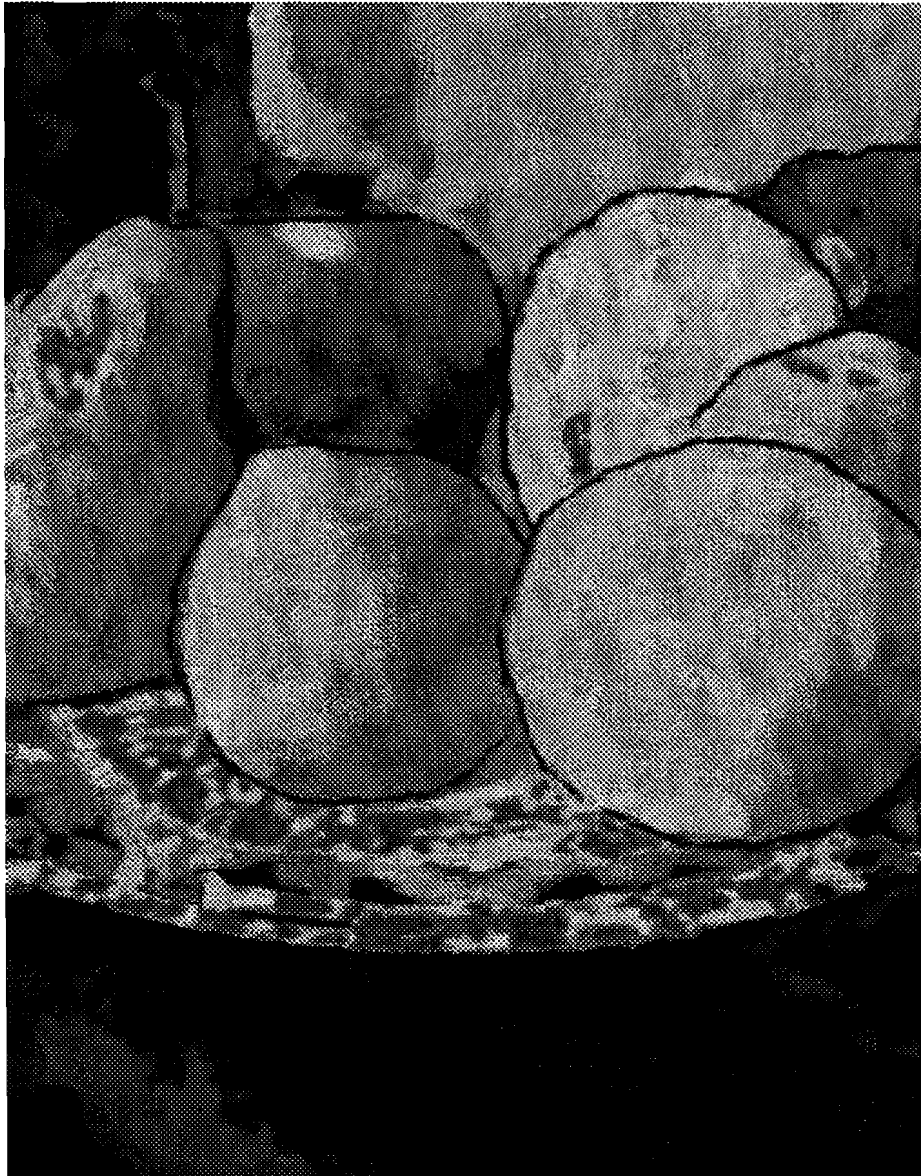


(a)

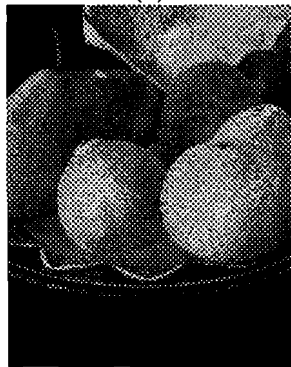


(b)

Figure 6.4: (a) Our painting (b) Input image



(a)



(b)

Figure 6.5: (a) Our painting (b) Input image

Chapter 7

Future Work

A natural extension to this work is a set of algorithms that would allow the system to make artistic judgments. One of these would be a color balance algorithm, able to correct the input image color, determine the right edges to highlight, and help determining color planes and brush size.

We would also like to make our brush model more complex and therefore more realistic. We believe it is necessary to have a better method for mixing the different paints than alpha-blending.

Chapter 8

Conclusion

We have shown that basing an algorithm on a painter's method can improve painterly rendered images. By choosing the subject (input image) carefully and allowing the user to define depth planes we can create paintings that convey depth well and have both spatial and color balance. By copying a specific painter's method we insure that all the pieces of the algorithm work well together to create a unified painting. This approach insures that the right combination of color palette, brush work, brush type and subject will be used.

Bibliography

- [1] Cassidy Curtis, Sean Anderson, Josh Seims, Kurt Fleischer, and David H. Salesin. Computer-generated watercolor. In *Proceedings of SIGGRAPH '97*, pages 421–430, August 1997.
- [2] J. Foley, A. van Dam, S. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1992.
- [3] Geoffrey Gardner. Visual simulation of clouds. In *Proceedings of SIGGRAPH '85*, pages 297–303, August 1985.
- [4] P. Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of SIGGRAPH '90*, pages 207–214, August 1990.
- [5] Aaron Hertzman. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of SIGGRAPH '98*, pages 453–460, August 1998.
- [6] Jain, Ramesh, Rangachar Kasturi, and Brian Shunck. *Machine Vision*. McGraw-Hill, Inc, 1995.
- [7] Y. Linde, A. Bruzo, and R. m. Gray. An algorithm for vector quantizer design. In *IEEE Transactions on Neural Networks*, 1990.
- [8] Peter Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of SIGGRAPH '97*, pages 407–414, August 1997.
- [9] Erle Loran. *Cézanne's Composition*. University of California Press, 1963.
- [10] Pavel Machotka. *Cézanne: Landscape into Art*. Yale University Press, 1996.
- [11] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of SIGGRAPH '96*, pages 477–484, August 1996.
- [12] A. Paeth. *A Fast Algorithm for General Raster Rotation, Graphics Gems*. Academic Press, 1990.
- [13] Michael P. Salisbury, Michael T. Wong, and John F. Hughes David H. Salesin. Orientable textures for images-based pen-and-ink illustration. In *Proceedings of SIGGRAPH '97*, pages 401–406, August 1997.
- [14] Mike Salisbury, Corey Anderson, Dani Lischinski, and David H. Salesin. Scale-dependent reproduction of pen-and-ink illustration. In *Proceedings of SIGGRAPH '96*, pages 461–468, August 1996.
- [15] Paul Smith. *Interpreting Cézanne*. Tate Publishing, 1996.
- [16] Steve Strassmann. Hairy brushes. In *Proceedings of SIGGRAPH '86*, pages 225–232, August 1986.
- [17] Paris WebMuseum. <http://www.oir.ucf.edu/wm/paint/auth/cezanne/>.

.1 APPENDIX

Distance calculation. To determine the distance between two colors, we first plot the colors into a color space and then measure the distance between the two colors (a and b) in that color space:

$$\text{distance} = (a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2$$

Initial HSV binning technique. A color is mapped to one of five bins in the initial clustering. We use the HSV color space to define the bins. Hue values between 60-180 are mapped to green, 180-300 are mapped to blue, and 0-60 and 300-360 are mapped to red. If the saturation is smaller than .35 and the value is bigger than 0.7, the color is mapped to white instead. A color whose value is smaller than .15 is mapped to black.

Dividing the colors using HSV. The colors are first sorted using the value and saturation variables of HSV. Given two colors (a, b) in HSV space our sort-rule is:

```

if(a.V>0.87 && b.V>0.87){
    if(a.S<b.S)
        return a is before b
    if(a.S>b.S)
        return b is before a
    return b and a are the same
}
if(a.V>0.87 && b.V<0.87)
    return a is before b
if(a.V<0.87 && b.V>0.87)
    return b is before a

if(a.V>0.6 && b.V>0.6){
    if(a.S<b.S)
        return a is before b
    if(a.S>b.S)
        return b is before a
    return b and a are the same
}
if(a.V>0.6 && b.V<0.6)
    return a is before b
if(a.V<0.6 && b.V>0.6)
    return b is before a

if(a.V>0.5 && b.V>0.5){

```

```

    if(a.S<b.S)
        return a is before b
    if(a.S>b.S)
        return b is before a
    return b and a are the same
}
if(a.V>0.5 && b.V<0.5)
    return a is before b
if(a.V<0.5 && b.V>0.5)
    return b is before a

```

```

if(a.S<b.S)
    return a is before b
if(a.S>b.S)
    return b is before a
return b and a are the same

```

We then divide the colors into four bins keeping the sorted order.

Shifting towards yellow. We take the initial palette and convert it into HSV space. We transform the hue variables so that yellow (60 H) becomes 0 H.

```

new_hue = old_hue - 60
if(new_hue < 0)
    new_hue = 360 + new_hue

```

We move the hues towards yellow using a sine function — the hues closer to yellow move more than hues further away. We map the hue value to the range [0,1] (0 is mapped to yellow and 1 is mapped to 180). For color with initial hues from 60 to 240 we subtract the sine function ($\sin(\text{current hue} * \text{PI}) / 2$) and for hues from 240-360 we subtract the sine function ($\sin((\text{current hue} - (1/3)) * \text{PI} * (3/2)) / 2$). Colors with initial hue values between zero and sixty are not moved because we can to preserve the reddish colors.

Orientation. We use a Gaussian filter to reduce noise in our intensity image. The size of the filter is 14x14 in the examples of this paper. The size of the filter in general should vary with the size of the detail in the image.

Fixed orientations. We map out orientations to eight fixed ones. We had a N/S, NNW/SSE, NW/SE, WNW/ESE, W/E, WSW/ENE, SW/NE, and SSW/NNE orientations. The range in degree of the orientations are:

N/S: 80–90 and 270–281

NNW/SSE: 61-79
 NW/SE: 36-60
 WNW/ESE: 11-35
 W/E: 349-10
 SWS/NEN: 327-348
 SW/NE: 216-327
 SSW/NNE: 215-280

Block filtering orientation. We modified our block filter to take into account both the orientation and the magnitude of the normal to the gradient. If we did not take into an account the magnitude, we would simply find the orientation that is most frequently used in an area. The problem with this approach is that you could encounter an area with a lot of near to zero vectors and then one or two very large vectors in another direction. If magnitude was not used we would have picked the direction of the near to zero vectors instead of the orientation of the larger vectors. Instead of counting the number of vectors of a specific orientation, we add the magnitude of all vector of a specific orientation. We pick the orientation that has the largest sum.

Blending strokes. To remove the hard edges between the color planes, we made sure that the color planes overlapped. Every time a stroke is applied we first check which if the stroke will cover an underlying color plane; if it will, we set all the values smaller than .87 in the transparency mask to zero. This allows the under-paint to shine through. We also used a random value to determine whether one should use the color of the underlying color plane or the color from the current plane. In our implementation there is a 40% chance that we use the underlying color.

Brush Size. Given the orientation and the maximum length and width of the stroke we calculate the actual size of the stroke. We create a stroke coordinate system. Its center is the center of the stroke and its x -axis is along the orientation and y -axis is perpendicular to the orientation. We calculate the bounding box of the stroke in canvas coordinates. We loop through the bounding box checking whether a pixel is marked as an edge and on the same or further-away depth plane as the center point. If so, then we transform the entry into stroke coordinates and check whether it is inside the stroke or not. If the entry is inside the stroke, the stroke's size is reduced until the entry is no longer inside the stroke. If a stroke becomes smaller than the minimal size it is not drawn.

In our case the minimum size of a stroke is nine pixels across and twenty-two pixels along the stroke. This minimal size depends on the sine functions that create the shape. The bigger the amplitude is the bigger the minimum size must be, this insures that the sides don't overlap each other.

Paint orientation. The size of the filter to determine the orientation is 21x21. This size worked well for our input images and its detail. Had the detail been smaller, a smaller size filter would have been applied, and likewise if the detail were bigger, a larger filter would be applied. Our edges are

associated with depth planes; If there is more than one possibility, the closest depth plane is used. This helps the filter to disregard outlines not associated with itself.

The block filter used to smooth out the direction of the outline was 10 by 10 and also makes sure that outline of a different depth plane has no influence. Again, this filter should be changed according to the detail of the image.

The color of the outline is chosen randomly from the 4 darkest colors of the palette. The length of the stroke is twenty two pixels and the width varies randomly between eight and ten pixels.

The size of the area marked around the center point is eight by eight. This size should vary with the length of the stroke.