ABSTRACT

TATEOSIAN, LAURA GRAY. Nonphotorealistic Visualization of Multidimensional Datasets. (Under the direction of Christopher G. Healey)

The huge quantities of data that are being recorded annually need to be organized and analyzed. The datasets often consist of a large number of elements, each associated with multiple attributes. Our objective is to create *effective, aesthetically appealing* multidimensional visualizations. By mapping element attributes to carefully chosen visual features, such visualizations support exploration, encourage prolonged inspection, and facilitate discovery of unexpected data characteristics and relationships.

We present a new visualization technique that uses "painted" brush strokes to represent data elements of large multidimensional datasets. Each element's attributes controls the visual features of one or more brushstrokes. To pursue aesthetic appeal, we draw inspiration from the Impressionist style of painting and apply rendering techniques from nonphotorealistic graphics. We construct our mappings to harness the strengths of the human visual system. The resulting displays are *nonphotorealistic visualizations* of the information in the datasets.

Studies confirm that existing guidelines based on human visual perception apply to our painterly styles. Additional studies investigate the artistic appeal of our visualizations, along with the emotional and visual features that influence aesthetic judgments. Finally, we use the results of these studies to combine painterly styles to build a tool which creates visualizations that are both effective and aesthetic and we apply our method to a real-world dataset.

NONPHOTOREALISTIC VISUALIZATION OF MULTIDIMENSIONAL DATASETS

by

LAURA G. TATEOSIAN

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh, North Carolina

2002

APPROVED BY:

_____ _____

_____
Chair of Advisory Committee

BIOGRAPHY

Laura Gray Tateosian was born to Louis Hagop Tateosian and Sarah Gray Tateosian in York, Pennsylvania. She received a Bachelor of Arts degree in Mathematics from Towson University, Baltimore, MD and a Master of Mathematics from the University of Oklahoma, Norman, Oklahoma, after which she taught mathematics at Shippensburg University in Shippensburg, PA. Laura is currently enrolled in the computer science masters program at North Carolina State University. She plans to continue her studies at North Carolina State University in the computer science doctoral program.

AKNOWLEDGEMENTS

My advisor, Dr. Chris Healey, offered guidance, assistance, wisdom, and understanding, and generously shared his knowledge, time, and energy. I want to thank him for being determinedly dedicated to my success in this endeavor. Most importantly, I'm lucky he has a good sense of humor.

Thanks Dad and Mom for sharing your appreciation of science and art. You are alway supportive, even at the beach! Thanks for reading my thesis there. I'm glad to provide nap-inducing material.

My sister and her family also encouraged me along the way. As I wondered if I would ever finish this report, I recalled Sarah's little voice proudly saying, "bigger and bigger." She was talking about a hole we were digging in the sand, but it reminded me of my growing thesis.

My lab group, Amit, Brent, Jason, Jiae, Mike, Reshma, Sarat, Vivek, helped my through the process with assistance and encouragement. Amit, thank you for screening my presentation, proofreading my thesis, and bringing me up to date on practical issues when I surfaced. Brent, thanks for your help with anything and everything throughout my entire time here. I was glad to supply distractions from your own drudgery. Also, thanks for the company through the final stretch (misery loves it) and the doughnuts one morning after a particularly rough night in the lab. Jason, thank you for introducing me to the basics in the graphics. Jiae, thanks for being a good friend and blazing the trail. Mike, thanks for providing your hacker knowhow. Reshma thanks for the good company. It's nice to have another female in the lab! Sarat, thanks for being a very good friend and a great listener. Vivek, thanks for making me laugh.

Additionally, I want to thank my friends Hema, Mark, Lory, and Pankaj. Hema, thank you

for arriving at midnight with essential reinforcement: the best chocolate bar I've ever tasted. Mark, thank you for being a kind friend. Lory, my long-time dear friend from college at Towson, thank you being there through the stormy parts of my life. This was monsoon season. The computer was a great idea. Remind me to take your advice more often. Finally, Pankaj, thank you for giving so much of yourself to help me succeed in this. You probably read my thesis more times than I did. You listened to my presentation so many times, you could have given it yourself. You heard my doubts and worries and did all the right things. Merci bien pour tous, mon cheri.

I was lucky enough to find a roommate, Sarah, who is also a friend and mentor. Though she doesn't understand the value of home-cooked dolmas, she is quite brilliant. She has unselfishly offered support, guidance, and encouragement. Aurora, our cat, made her contributions by getting up of my thesis long enough for me to finish it.

Professoressa O.Nagel provided a useful motto "Forza e coraggio–Coraggio e forza!" - Thanks for the fortification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Extraordinary amounts of digital data, over 1 quintillion [1] bytes, are being recorded annually [Lym00]. For raw data to be useful, it must be organized and presented in a format that facilitates analysis and interpretation. Numerical format of massive datasets is clearly difficult to interpret. Statistical analysis alone may not yield as many insights as visualization. Visual exploration is more effective than computational exploration in certain situations. Visualizations allow for open-ended exploration of the data, so that unexpected discoveries can be made. McCormick et al. cite an example of an astrophysicist who found an erroneous boundary condition in his code after examining an image of a jet stream with an obvious irregularity not apparent in the numbers [Mcc87]. A combination of statistical analysis and visualization may be appropriate for some data.

Effective scientific visualizations convert datasets into images which allow viewers to analyze and explore their data, and to discover and/or validate patterns, anomalies, or other im-

[1] 1 quintillion = 10^{18}.

portant characteristics [Hea01]. Discussions of visualization amongst experts in Earth science, aerospace research, molecular biology, defense, and medicine emphasize its vital importance in these fields [Ger94, Smi98]. Visualization has been described as "a major component of computational science and engineering, and critical to understanding the results of large-scale scientific simulation." [Smi98]. A 1987 NSF panel on scientific visualization noted that an estimated 50 percent of the brain's neurons are associated with vision and called visualization essential for providing and communicating insights to others [Mcc87]. The panel strongly recommended launching a national program in scientific visualization, declaring that visualization would "change the way science is done."

More than a decade later, information sources and categories had expanded enormously and challenges in representing abstract and highly-dimensional data were still being discussed [Smi98]. Large multidimensional datasets, i.e., sets in which each element has multiple attributes, still present challenges for visualization. To create a visualization of an m-dimensional dataset of n elements, we must choose a set of visual features, $V = \{V_1, ..., V_m\}$ to represent the data's set of attributes, $A = \{A_1, ..., A_m\}$. In figure 1.1, for example, which shows a map of Napoleon's 1812 march on Russia, $A = \{size\ of\ the\ army,\ 2D - location,\ direction\ of\ the\ army's\ movement,\ temperature\ during\ the\ retreat\}$ and $V = \{$ line thickness, xy-coordinates of top bands, color, y-coordinates of red line $\}$.

Once V is established, we define a mapping, M, from values of each attribute to values of a visual feature. In figure 1.1, the mapping is as follows: $army\ size \rightarrow$ line width, $location \rightarrow$ xy-coordinates of top bands, $direction \rightarrow$ color, and $temperature \rightarrow y$-coordinate of red line. The viewer can rapidly observe the army's numbers dwindling as troops advance, and quickly

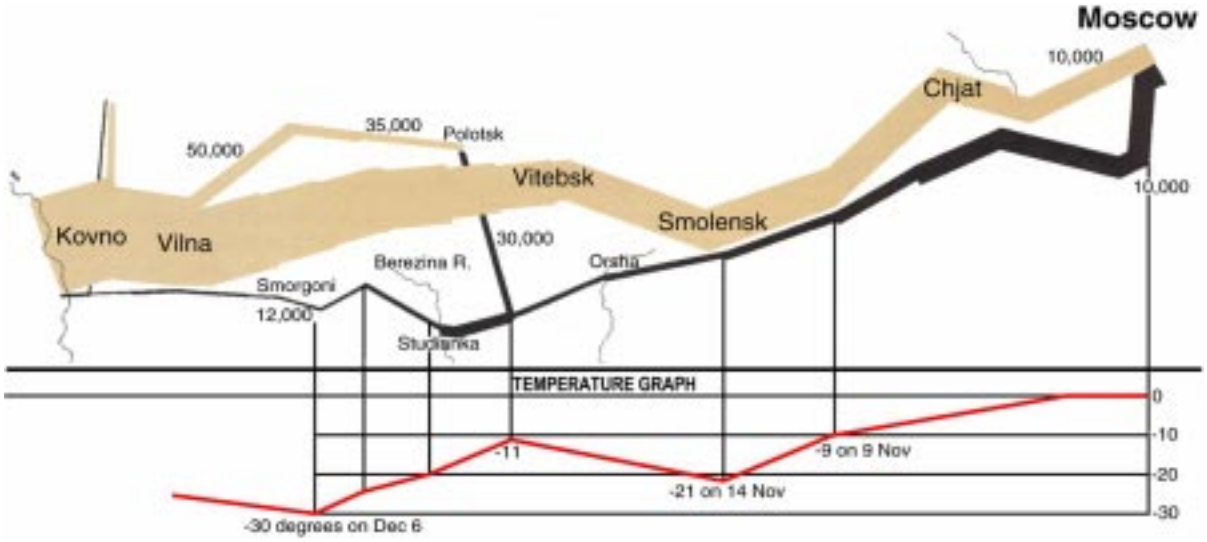
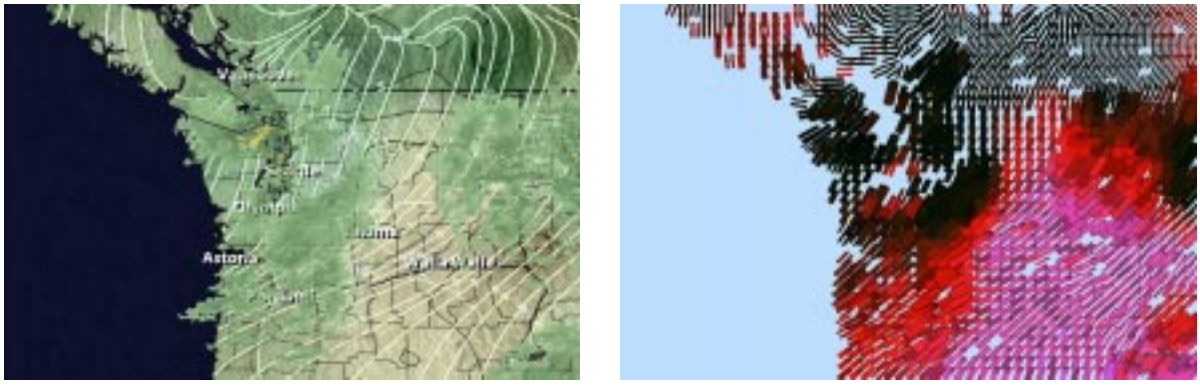


Figure 1.1: Charles Joseph Minard's 1861 map of Napoleon's 1812 campaign on Russia [Min61], is considered remarkable for how much information it encodes. E.R. Tufte describes the map [Tuf83]: "Beginning at the left on the Polish-Russian border, the thick band shows the size of the army (422,000 men) as it invaded Russia in June 1812. The width of the band indicates the size of the army at each place on the map. The retreat is depicted by the darker, lower band, which is linked to the temperature scale and dates at the bottom of the chart. It was bitterly cold and many froze on the march out of Russia. The army struggled back to Poland with only 10,000 troops remaining." $A = \{\ size\ of\ the\ army,\ 2D - location,\ direction\ of\ the\ army's\ movement,\ temperature\ during\ the\ retreat\}$ and $V = \{$ line thickness, xy-coordinates of top bands, color, y-coordinates of red line $\}$.

estimate the army to be 1/4 of its original size by the time it reaches Moscow, by comparing the thickness of the left and right ends of the top band. The luminance of the bands representing the army's advance greatly differs from the luminance of the bands representing the retreat, so that even color-blind viewers can perceive the difference.

Minard's visualization has not lost its effectiveness over time, perhaps because it lends itself well to our perceptual abilities. Visualization designers have begun to recognize the importance of studying human perceptual abilities. A 1998 Panel on Visualization identified it as one of the difficult problems in visualization. Specifically, they asked, "How can human perceptual and cognitive talents be enhanced and amplified through visualization?" [Zei98] We are asking similar questions: How can we harness human perception skills? Could we



(a) Mapping, M_a (b) Mapping, M_b

Figure 1.2: Different visualizations of the same dataset: [Cha01] (a) Mapping, M_a: $temperature \rightarrow$ color $\in$ [green,...,yellow], $wind\ speed \rightarrow$ luminance $\in$ [dark,...,bright], $wind\ direction \rightarrow$ directed contours, $\in [0°, ..., 360°]$, $precipitation \rightarrow$ semi-transparent color $\in$ [green,...,red] (b) Mapping, M_b: $temperature \rightarrow$ color $\in$ [dark green,...,bright pink], $wind\ speed \rightarrow$ density $\in$ [sparse,...,dense], $wind\ direction \rightarrow$ orientation $\in [0°, ..., 360°]$, $precipitation \rightarrow$ size $\in$ [small,...,large]

create more effective mappings by studying the low-level human visual system and how human perception works?

The mapping in figure 1.2b was created in such a fashion, though figure 1.2a was not. Figures 1.2a and 1.2b both visualize the same weather dataset, but the effect is very different. Each element in the dataset has the following attributes: $temperature$, $wind\ speed$, $wind\ direction$, and $precipitation$. Figure 1.2a was created by forming a composite of three traditional weather maps. The three traditional maps had the following individual mappings: Map1. $temperature \rightarrow$ color, Map2. $precipitation \rightarrow$ semi-transparent color, and Map3. $wind\ speed \rightarrow$ color and $direction \rightarrow$ directed contour lines. The result of combining these three maps is a familiar TV news-like weather display, but a rather faulty visualization. For example, the semi-transparent color used to indicate precipitation causes some problems. In areas where precipitation is high, is it cold? Is it windy? The overlap interferes with our ability to perceive other attributes. A simplistic alternative to using semi-transparent colors, such as

outlines tracing clouds, would suppress precipitation data. What would then indicate higher levels of precipitation shown now by the yellow clouds? Luminance was chosen to represent wind speed since opaque color was already being used for temperature. However, perceiving the changes in luminance is difficult.

The mapping used in figure 1.2b was designed with the aid of ViA, a visualization agent created in our lab [Cha01]. ViA suggests mappings based on perceptual guidelines. The agent chooses mappings that minimize interference and assign the most salient features to the attributes that are of greatest interest to the viewer. Data attributes were mapped to the color, size, orientation, and density of the small rectangles that comprise the visualization. The mapping is as follows: $temperature \rightarrow$ color, $wind\ speed \rightarrow$ density, $wind\ direction \rightarrow$ orientation, and $precipitation \rightarrow$ size. This mapping eliminates the interference of cloud coverage that occurred in the original mapping. To find cold regions in areas of high precipitation, for example, we look for large, dark, green rectangles. Finding combinations of wind speed and temperature means attending to density and color, as opposed to the luminance and color combination used in figure 1.2a. Warm, low wind speed areas have sparsely placed pink glyphs. The colors were selected to be easily differentiable and balanced to accurately reflect the changes in the underlying data values. This mapping appeals to the low-level human visual system by employing features like color, orientation, and size.

To enhance the effectiveness of the mappings we choose, we also want to entice viewers into prolonged inspection of the images. We believe that images that are engaging and attractive will encourage viewers to study them, thereby improving the likelihood that they may discover anomalies or trends not immediately obvious. This brings us to another question posed by the

1998 Visualization Panel: “How can the long and rich history of visualization in the arts be exploited by the information age?” [Zei98]

The influence of the fine arts on graphics can be seen in the relatively new field of graphics called non-photorealistic rendering. This is a diverse category of graphics, ranging from projects like simulating pencil marks to automating the cartoon creation process. For years scientists in (traditional) graphics modeling and rendering have studied the physics of light and color to develop techniques for creating images that look amazingly like photographs. Contrary to this trend some are studying the techniques of artists to create artistic images with computer graphics. These techniques lend themselves well to our goal of creating appealing visualizations. Also, there seems to be a correlation between the techniques used by artists and techniques revealed to be effective by studies of low-level human perception.

1.1 Main Contributions

The current data visualization environment as described in the previous section, points to the need for continued research on displaying multi-dimensional datasets. These displays must facilitate analysis and exploration and engage the viewer. These requirements motivated our work. Specifically, the goals of this project are to create visualizations which are:

- *Effective.* As discussed above, the need for effective visualizations is essential for the scientific community[Ger94, Lym00, Mcc87, Smi98].

- *Multidimensional.* Most applications generate datasets of high dimensionality [Kir99, Rhe01, Lym00]. Multidimensional visualizations provide important information about

relationships between attributes. The example shown in figure 1.2a gives a taste of the challenges involved in visualizing multiple attributes in a single display.

- *Engaging and aesthetically pleasing.* Creating aesthetically pleasing visualizations may enhance the effectiveness of the visualizations, since viewers will be encouraged to gaze at them for extended periods.

To achieve these goals, we plan to: (1) create mappings based on human perceptual strengths, (2) employ a painterly style based on artistic characteristics, and (3) test the resulting styles for effectiveness and aesthetic appeal.

We will create visualizations that look as if they are "painted" with textured glyphs that resemble brush strokes. The glyphs will be distributed as if they are being painted on a canvas. Data attributes will be mapped to the visual features of the glyphs (e.g. color, size) to support multi-dimensionality. Mappings will be determined based on human visual perception of color and texture so that our visualizations will take advantage of human perceptual strengths.

We will seek guidance for our painterly styles from groups in two distinct fields: computer scientists in nonphotorealistic rendering and artists in Impressionist painting. By studying the techniques in nonphotorealistic rendering, we can build on the research of scientists who are already using computer graphics to "paint." To create aesthetically pleasing "painted" images, we will study the techniques of master Impressionist artists. The characteristics of Impressionist artwork may provide insights on creating expressive and appealing visual displays.

Since we intend to "paint" our visualizations, the techniques we will use to create our visualizations will be different from existing ones. In order to confirm that existing human

perceptual guidelines apply to our techniques, we will conduct studies with images created in this painterly manner. Results from these studies will serve as an indication of the effectiveness of the technique. In order to test for aesthetic appeal, we will ask observers to judge our painterly images on artistic merit. We will also try to identify the basic emotional and visual factors that influence these judgments.

The results of this work may influence scientists in visualization to consider the impact of aesthetic appeal on scientific visualizations. This project raises some general questions. Could existing techniques be enhanced by adjusting visualizations to be more engaging? Could other new visualization techniques be discovered by studying artistic techniques along with perceptual guidelines? More exploration of this methodology may lead to the discovery of new ways to effectively visualize more dimensions simultaneously. Additionally, our studies of the effectiveness of new painterly styles may offer insights into how we perceive certain combinations of visual properties.

1.2 Paper Organization

The remaining chapters progress as follows: Chapter 2 surveys related work in nonphotorealistic rendering. Chapter 3 elaborates on the research in human visual perception guidelines on color and texture. Chapter 4 briefly highlights important Impressionist principles. Our studies on effectiveness and aesthetic judgment are discussed in Chapter 5. Chapter 6 describes the implementation of a system to realize nonphotorealistic visualizations and describes a practical application of the tool. Last, Chapter 7 discusses conclusions and future work.

Chapter 2

Nonphotorealistic Graphics

Nonphotorealistic graphics applies drafting and artistic techniques to imaging. Photorealistic rendering involves careful modeling how light interacts with the materials in the world and understanding how we perceive the light rays entering our eyes [Jen01]. Imagine modeling a tree realistically with computer graphics. A tree's bark is knotted, multi-colored, and creased. Thousands of unique leaves catch the sunlight at different angles and cast thousands of shadows. Scientists at the University of Calgary, for example, render plant topiaries like the one shown in Figure 2.1 [Pru94]. A great deal of detail must be included to create a realistic look. Although such renderings are a great accomplishment, nonphotorealistic renderings offer their own unique advantages [Win94, Goo01, Str02]. For example, nonphotorealistic images can:

- convey information better by omitting extraneous details
- focus attention on relevant features, by clarifying and simplifying shapes
- expose hidden features

Figure 2.1: Computer graphics rendering of the topiary garden at Levens, England, created by R. Mech, P. Prusinkiewicz [Mec96].

- utilize less storage space and be more easily reproduced and transmitted
- provide a more natural vehicle for conveying information at different levels of detail

Figure 2.2, a simple hand-painted watercolor tree, exhibits some of these characteristics. When artists depict foliage, they don't paint every leaf. Instead they use brush strokes to abstractly represent the leaves [Mei96]. Introducing some abstraction enables the image to be drawn without alluding to each minute detail. Released from physical constraints of realism, a rendering often more succinctly communicates its content.

Figure 2.2: Hand-drawn tree in watercolor markers by the author.

2.1 Methodology for Simulating Artwork

Many researchers in nonphotorealistic rendering create artistic images based on photographic images or 3D models. Even though our basis, i.e., datasets, is more abstract, their methodologies and techniques influence our work. Some of their projects are focused on producing convincing physical simulations of artistic media, such as pen-and-ink, colored pencil, or graphite pencils. Their approach involves choosing a specific artistic medium, compiling a list of properties exhibited by artistic works in that medium, and developing algorithms to draw images that conform to these rules.

Winkenbach and Salesin, for example, developed a pen-and-ink modeling system [Win94]. They began by listing the properties of pen-and-ink sketches. For example, they observed that lines vary in thickness, absence of detail indicates glare, crisp lines are used for glass, and sketchy lines are used for weathered materials. Next they selected algorithms to draw images that conform to these rules. A collection of hatching strokes vary in direction, style, and density to reproduce texture and tone differences. They also use outlines sparingly to indicate but not overdo boundaries. In later work, they added the ability to sketch curved objects by developing a method to vary stroke direction on curved surfaces and introducing a new controlled-density hatching algorithm [Win96]. Work on modeling pen-and-ink renderings has been continued by Salisbury et al [Sali97]. Figure 2.3 demonstrates a rendering of a difficult-to-model non-smooth furry surface.

Figure 2.3: Computer generated pen-and-ink style raccoon sketch created by Michael P. Salisbury, Michael T. Wong, John F. Hughes, David H. Salesin. [Sale02].

Other projects similarly base their work on artistic properties, but also model the microscopic media-paper interactions. Takagi et al. simulate colored pencils by modeling the microstructure of paper and particle dispersion when round-tipped colored pencils mark the paper [Tak99]. The model uses 3D pixels, i.e., voxels, to model the paper content (initially pulp fibers and talc) and record the changes occurring during the drawing process. Sousa and Buchanan model the interaction of graphite pencils, erasers, blenders, and paper based on electron microscope images of paper [Sou99a, Sou99b, Goo01]. The model allows for variation of a large number of parameters. How the pencil is held by the artist, the pressure on the pencil, pencil hardness and sharpness and paper roughness are simulated.

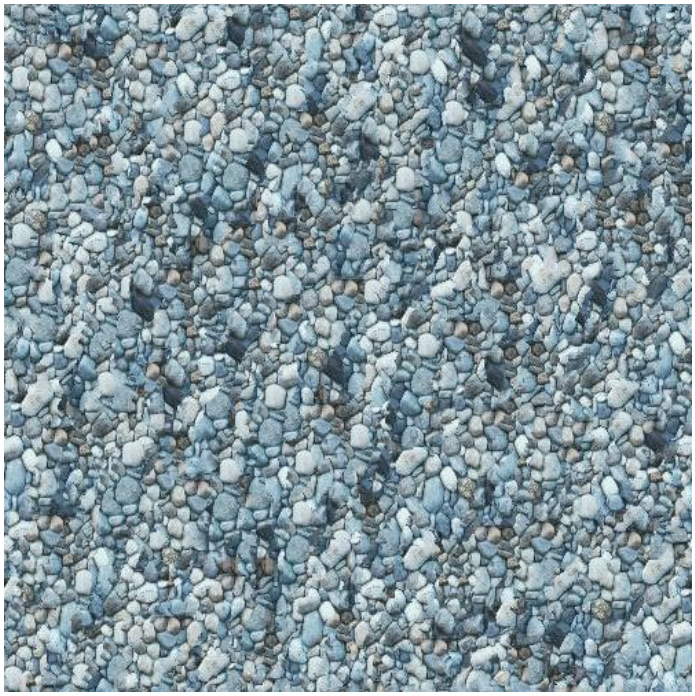
2.2 The Texture Approach to Painting

Other scientists approach nonphotorealistic rendering by focusing on texture. Inherent in hand-drawn images due to the paintbrush bristles, the pencil tip's wear, the paper's fibers, or the canvas weave, texture must be added as a separate step in computer graphics images. Texture mapping, a common technique for adding realism to computer-generated scenes, overlays an image onto an object. A wall may be drawn by laying a brick texture over a rectangle or a semi-transparent texture may be mapped to create clouds. Haeberli and Segal draw layers of translucent textures on a background image to simulate air-brushing and other types of paint strokes [Hae93].

The texture synthesis process involves generating texture based on a texture sample. Ashikhmin worked on synthesizing natural textures. Given a small image of pebbles, Ashikhmin

could create an image that covers a driveway with what look like authentic pebbles [Ash01]. Figure 2.4 shows an example of texture synthesis created by Ashikhmin. Figure 2.4a is the sample image used to create figure 2.4b, a synthesized pattern of pebbles. Using texture synthesis,

(a) Sample image



(b) Synthesized image

Figure 2.4: Example of texture synthesis applied to a natural texture.

Lewis developed an interactive painting system [Lew84]. Figure 2.5 shows some examples of images created with this system. In Lewis's program the painter designs a texture sample and specifies several variables that effect how the edges of the texture are drawn. The program generates textured areas, as if painting brush strokes of varying sizes, based on the size of the synthesis region.

Figure 2.5: Digital paintings with texture synthesis. The painter designs a sample texture and the program generates textured areas.

2.3 Painting Programs

We hope to visualize data using “painted” strokes, as if an Impressionist artist were rendering our datasets. The collection of projects described below, though diverse in focus and outcome, all synthesize brush strokes to “paint” non-photorealistic renderings. We are interested in their brush stroke techniques, as well as how they composed the overall image.

In 1986 Strassmann developed a system for painting in a style of Japanese art chosen for its minimalist qualities. Only a few brush strokes are used and ink is only shades of gray. This allows for focus on the quality of each stroke [Stra86]. The system uses four objects: brush, stroke, dip, and paper. The brush is composed of numerous bristles. Strokes have trajectory, color, and pressure. Dip is the set amount of ink in the well. Paper stores the results of each stroke.

Haeberli’s “Paint by Numbers” program allows users to interactively convert synthetic or natural scenes into Impressionist images [Hae93]. The user “paints” the strokes over the source image, varying the size, direction, and shape (footprint of the brush) using various mouse ges-

tures. The location and color of the stroke are based on the location of the mouse. The source image color at the current location establishes the color of the stroke, freeing the user from choosing the color as in previous painting programs. Advanced extensions include “pushing edges” and painting 3D synthetic scenes. “Pushing” an edge is a technique that exaggerates important edges. This is achieved by making dark edges slightly darker and light edges slightly lighter where dark and light edges meet.

Hsu and Lee produced a painting program with a style suitable for Chinese brush art and for cartoons [Hsu94]. Defined by a reference backbone and reference thickness, “skeletal strokes” constitute their vector-based drawing primitives. The strokes are based on a transformation of the localized parametric coordinate system. A user may choose to simulate various effects, such as, water-based ink, wood-cut, or flat-nib pen. Each stroke varies in thickness, an attribute not available in previous painting programs.

Meier used partial placement to create animated painterly renderings that have an Impressionist look [Mei96]. The program divides the surface of a source image into triangles that approximate the shape. Particles are then placed in each region. The number of particles in each region is based on the size of the region. The orientation, size, and color of the strokes may be stored within the particles. The particles are depth sorted and rendered as 2D strokes using a painter’s algorithm.

Painterly rendering programs usually apply some randomness to placement, color, and other stroke features to reduce the rigidity inherent in computer-generated images. This randomness can cause undesirable jumpiness when painterly images are animated. Thus, frame-to-frame coherence is a challenging issue. To maintain coherence, Meier stored a seed with

each particle so that the same random perturbations are used throughout the animation. In 1997 Litwinowicz presented another algorithm for producing painterly animation with an Impressionist style [Lit97]. While Meier used computer-generated animations as input, this program used video clips as input. brush strokes, oriented normal to the gradient direction of the original image, are clipped at detectable edges in the source image. The program maintains a list of strokes and uses optical flow fields to maintain temporal coherence.

In 1997 Curtis et al. designed a system for creating computer-generated watercolors [Cur97]. Though not focused on the medium, the previous two examples must have assumed oil or acrylic paint which yields defined, distinct brush strokes, whereas watercolor can be much more fluid. This project models the nuances of watercolor painting by simulating unique effects, such as edge-darkening, granulation, backruns, separation of pigment, and glazing. Three different applications were developed: an interactive watercolor system, an automatic image "watercolorization," and non-photorealistic rendering of 3D scenes.

Another program designed to paint a wide range of visual styles, from color wash to Impressionist, creates images with a hand-drawn look from photos [Her00]. The program "paints" with curved brush strokes rendered in layers. Larger strokes are drawn first, then the image is painted over and over with successively smaller strokes in areas with more detail. The designer can vary parameters to create different styles. For example, the painting can be sketchier or more precise and brush stroke curvature may be limited or exaggerated.

2.4 Visualization Applications

Scientists in visualization have recently begun to focus on how studies of human perception and nonphotorealistic rendering can be applied to visualization. Interrante investigated human perception of textures to create visualizations using natural textures, such as bubbles or bee hives or woven materials [Int00]. These textures offer intricate variety and subtle richness, qualities lacking in regular synthetic textures on a flat plane.

Several scientists have been investigating the use of nonphotorealistic rendering techniques to supplement established visualization techniques for applications such as flow and volume illustration. The results are promising.

For example, Ebert and Rheigans augment physics based rendering techniques with non-photorealistic techniques to create volume illustrations [Rhe01]. Depth-cueing dims the color of objects far from the viewer. Null-halos leave the areas just outside surfaces empty, even if an accurate depiction would show a background object there. Tone is varied based on orientation to the light. Surfaces facing the light get a warm cast and those facing away from the light get a cool cast. These are just a few examples of the enhancements employed. Their techniques are demonstrated on volume illustrations of human internal organs with impressive results. Figure 2.6 shows some examples. Figure 2.6a shows the original gaseous image of an abdominal CT Scan (Computed Tomography Scan). CT Scans are used for imaging soft tissue, bone, and blood vessels. Figure 2.6b demonstrates tone enhancement. Figure 2.6c demonstrates boundary and silhouette enhancement and halos.

Laidlaw et al. constructed a visualization technique for diffusion tensor images that may

(a) Original gaseous image of an abdominal CT Scan.

(b) Tone enhancement

(c) Boundary and silhouette enhancement and halos

Figure 2.6: Ebert and Rheigans enhance volume illustrations using nonphotorealistic techniques.

provide early diagnostic value for neurogenative diseases [Lai98]. The rate of diffusion of water molecules in a biological system can provide important information about the structure of the underlying tissue. The relevant data yields seven values in each spatial location. Applying concepts borrowed from artists, Laidlaw et al. created an effective visualization using varied, layered brush strokes.

These methods are extended to problems in fluid mechanics [Kir99]. A combination of discrete and continuous visual elements are arranged in multiple layers to illustrate attributes such as velocity and vorticity. The techniques also allow them to visualize several additional quantities that had rarely been visualized before. The visual features include a primer, an underpainting, an ellipse layer, an arrow layer, and a mask layer.

Laidlaw, involved in both of the latter two projects described above, made the following observations about what scientists in visualization can learn from paintings [Lai01]:

Paintings are multiscale. When viewed from different distances they can be seen and under-

stood differently.

Paintings have a temporal component. We see different aspects of an image at different viewing times. Some parts stand out quickly, like the overall composition or palette of a painting, and some take more time to become apparent, like texture or shape of individual strokes.

These concepts extend to reading painterly visualizations. Laidlaw expanded on this point with the following comments [Lai98]: brush strokes encode information individually showing specific values and collectively showing spatial connections and generating a texture and sense of speed corresponding to the speed of diffusion. To summarize, the strokes represent data more qualitatively from a distance and more quantitatively up close.

Laidlaw was particularly influenced by Impressionist artwork, as were several other scientists mentioned in the previous section [Hae93, Hsu94, Mei96, Lit97, Her00]. The next chapter describes the influence of Impressionism on our project.

Chapter 3

Painterly Style

The painterly style created by Impressionist artists, e.g., Monet, Renoir, Pisarro, Cezanne, and Van Gogh, provided inspiration for our project. The Impressionist art movement consisted of a small group of painters who departed from the dominant artistic practices of the time. The term "Impressionism" was derived from the Monet painting *"Impression Sunrise (Le Havre)"*, which was characterized by a sketchy form and attempted to capture certain atmospheric effects [Nov95]. Monet, whose work exemplifies the Impressionist principles, held that the first real look at the motif was likely to be the truest and most unprejudiced one [Per27]. The Impressionist style of painting is characterized chiefly by concentration on the general impression produced by a scene or object [Pio02].

We chose the Impressionist style to lend focus to our project, though we consider our methodology to be a general one that could use other painterly styles as a basis. Our design echos the Impressionist principles. The points below further describe the main features of the Impressionist style.

Typically, they used small, quick brush strokes to simulate the reflected light. Our tool applies a sprinkling of brightly colored strokes that portray high spatial frequency details.

Primary importance was given to tiny, moving, brightly colored strokes that depict movement and create form [Whi78]. Our visualization consists of small, directed, colorful brush strokes that vary in length and surface texture.

Monet said that the first (layer of a) painting should cover as much of the canvas as possible, no matter how roughly, so as to determine at the outset the tonality of the whole [Per27]. Our underpainting covers the canvas and sets the tone by deriving size and orientation from data that determines the size and orientation of the final coat.

Impressionist artists studied color and light methodically. Some studied results of color scientists like Chevreul and Rood, while others developed their own scientific models of color [Che67, Bro50, Roo79]. In a 1905 letter, Monet emphasized the use of colors that would make the painting brighter [Mon05]. Following their example, our palette consists of colors chosen from a carefully developed scientific model.

Finally, the following statement by art historian Fritz Novotny reveals the essence of the art form: "The highly ingenious artistic system of illusion in Impressionist painting ultimately depends on two types of purely formal structure: the structure of marks created by the visible pattern of brush strokes and the structure established by color relations [Nov95]." The next chapter describes the perceptual considerations that influenced the structure of the marks and colors that constitute our visualizations.

Chapter 4

Visual Perception

Human perception is based on a sophisticated and complex information processing system designed to optimally extract environmental information. A visualization's effectiveness, to some extent, depends on how well it is designed as an input to this system [Ira00]. To address this we study how visual perception operates. People commonly believe that they see "what is there" by merely opening their eyes and looking, like a camera taking a snapshot. However, human vision differs greatly from modern photography [Hea02]. Although a camera creates a two-dimensional replica of the scene, the human visual system cannot fully process all of its input [Wol94]. Healey compiled a list of significant findings that differentiate human vision from photography [Hea02]:

The retinal image is only fully processed at the fovea [Wol94]. Detailed form and color vision is only possible for a tiny window of several degrees of arc surrounding a gaze location.

Inspecting an entire scene requires a series of eye movements. Even when the eye seems to be fixed at one point, the eye jumps ballistically. Wooding et al. are interested in how people view paintings [Woo00]. They recorded observers' eye movements as they viewed images of paintings at the National Gallery in London. Figure 4.1 shows a trace of an observer's eye saccades (ballistic eye movements) as he inspects *The Execution of Lady Jane* by Grey Paul Delaroche.

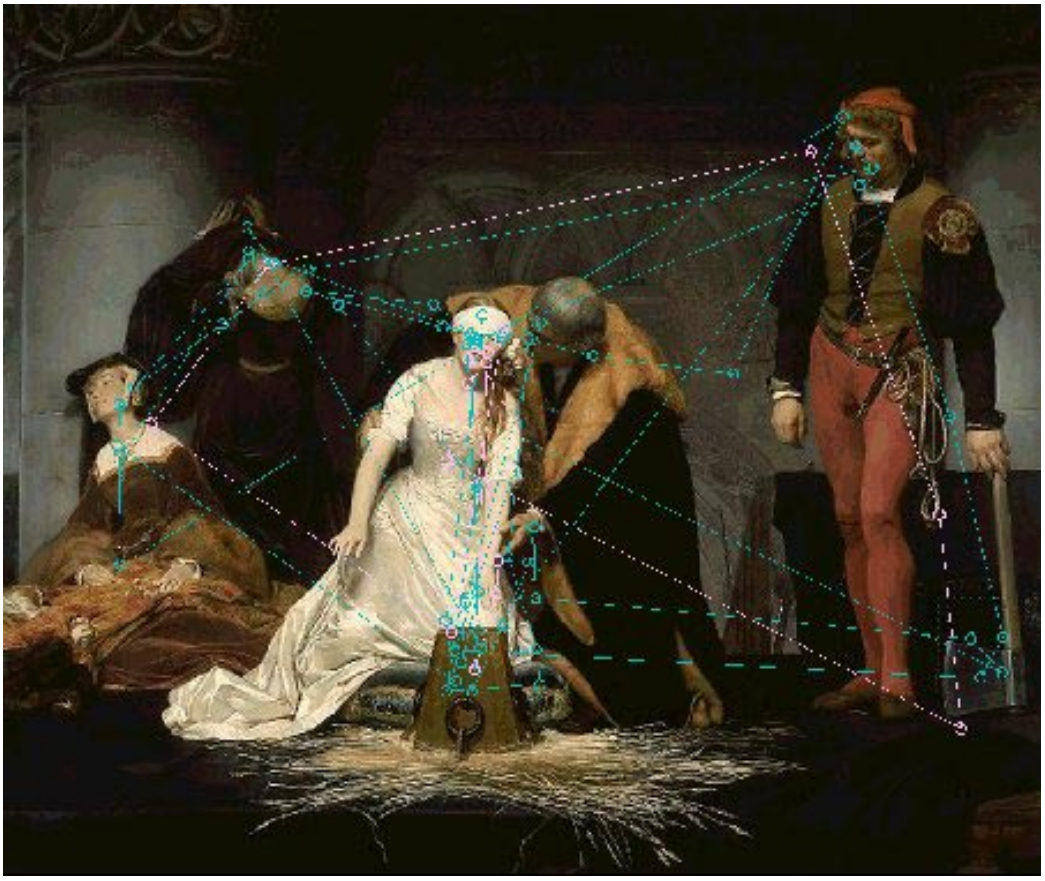


Figure 4.1: *The Execution of Lady Jane* by Grey Paul Delaroche with fixations and scan-paths of eye movements of an observer inspecting the painting. The series of discrete saccades that are required to see a "whole scene," are time-consuming.

The series of discrete saccades that are required to see a "whole scene," are time-consuming. While the eye is moving, image blur makes the visual system effectively blind. Saccadic suppression imposes an additional sensitivity loss for a period that outlasts the saccade by 50 ms or more [Tri02].

Memory of what is seen in a glance is extremely limited and dependent on where attention is directed and what is sought.

Human vision is honed to be drawn to objects that are very different from their surroundings, or that change or move, because of the different signals emanating from these locations. Objects that remain unchanged may escape notice.

Only a few basic features can be used to guide attention. The most prominent properties are hue and luminance. The next level includes orientation, motion, and texture, followed by length, area, and convexity.

Research in human perception indicates that visualizations should be carefully tailored to harness human visual perception skills [War00, Wol94]. Research in perception and visualization guided our work. Specifically, color and texture selection and visual feature perception hierarchy research steered the development of our visualization tool.

4.1 Color Selection

Color names are too imprecise to be useful in descriptions of color. Scientists have dealt with this issue by creating various color models that define colors precisely in terms of a physical

model and associated mathematical equations.

Early publications on the study of color include work by M.E. Chevreul in 1839 and A.H. Munsell in 1905. Chevreul's "Principles of Harmony and Contrast of Colors," a collection of notes from work in the color lab of his textiles factory, was a major influence on artists of the time, including Delacroix, Pisarro, Monet, and Seurat [Che67].

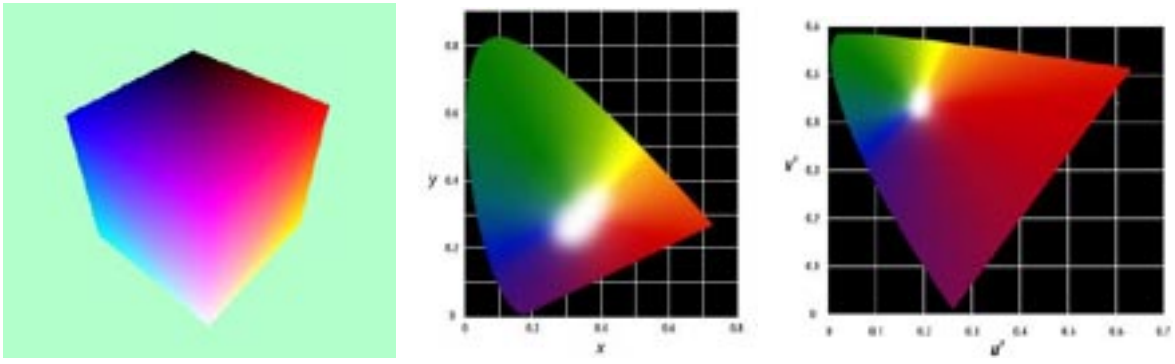
In order to describe the colors on his sketches to his color composition classes in definite terms, Munsell arranged colors on a sphere [Mun45]. In "'A Color Notation" he first describes his model as a peeled orange divided into five parts still connected at the bottom. One section contains all the greens. Another contains all the blues, etc. The sphere models hue, value, and chroma (i.e., saturation). Munsell describes hue, value, and chroma as the name of a color, the lightness of a color, and the strength of a color, respectively. In physical terms these correspond to dominant wavelength, excitation purity, and luminance and together they are necessary and sufficient to specify a color precisely.

The RGB (Red Green Blue) Color Cube is used to specify these three color components on computer monitors [Fol97]. An RGB cube is shown in Figure 4.2a. Each corner of the cube is a unique fully saturated hue. Colors are denoted by triples (r, g, b) for $0 \leq r, g, b \leq 1$ where r, g, and b represent the amount of red, green, blue. For example, $(1, 0, 0)$ represents red and $(1, 1, 1)$ represents white. The model is based on the notion that all colors of light can be specified by some combination of the three primary light colors red, green, and blue. Since this is not entirely true, no computer monitor can produce all visible colors and the subset that it does produce is called its gamut.

In the 1920's W.D. Wright and J. Gould employed this additive color notion [War00]. They

conducted a series of experiments on a large number of normal-sighted people to establish three color matching functions that show the amounts of red, green, and blue light needed for the average observer to match each color in the visible spectrum. Many of the colors could be matched by additive combinations of red, green, and blue. However, hues with wavelengths between 438.1 nm and 546.1 nm could not. To match these colors some red needed to be added to the color being compared. A segment of the red curve is negative. This corresponds to adding red to the target patch to produce the match. This problem was addressed by the Commission International D'Eclairage (French for "International Commission on Illumination").

In 1931 the Commission International D'Eclairage (CIE) applied a linear transformation to the original r, g, b curves to produce three new functions $x, y,$ and z. These new functions contain only positive values over the visible color spectrum. This resulted in the CIE XYZ model and later the CIE LUV model (shown in Figure 4.2 b and 4.2c, respectively [Ado02]) [Fol97]. Both models are horseshoe shaped color maps that specify the entire spectrum of colors. The CIE LUV model has the advantage of being roughly perceptually balanced, an important property for data visualization.



(a) RGB Cube (b) CIE XYZ (c) CIE LUV

Figure 4.2: Some examples of color models.

Perceptual balance is a measure of whether perceived color differences between pairs of colors correspond to the physical distances between the color pairs in a given model. Does the difference between yellow and red appear to be the same as the difference between black and red? The pairs are the same distance apart in the RGB Color Cube Model. To compare pairs of numerical values, we find the relative differences by subtracting. How our visual system compares pairs of colors is more mysterious. Mapping data values to colors to reflect data accurately is a challenge.

Researchers have built systems to let users interactively explore different color mapping schemes. Rheingans and Tebbs developed a tool for dynamic exploration of color mappings [Rhe90]. The tool displays the current color mapping on a dataset, a three-dimensional color space, and a chart which relates distance traveled along a color path to data value increments. The color space is a three-dimensional cloud of color samples from a color model. The system allows the user to choose an RGB cube, an HSV cone or an HSL double cone for the color model. A curve inside the color space shows the color path which defines the sequence of colors used to represent the data values. The user could also select a linear or exponential mapping. Each setting could expose different aspects of the data. For example, an exponential mapping would map most of the data values to a relatively small range of colors, mapping the remaining portion to a larger part of the path, exposing subtle details in the high values of data. Bergman et al. presented a rule-based tool for assisting colormap selection [Berg95]. The tool assists the user in finding color mappings to display data faithfully by giving advice based on characteristics of the dataset and the kinds of information the user wants to be revealed.

Other scientists propose color mappings based on principles of color perception. Colin

Ware, an experimental psychologist and computer scientist, developed guidelines for choosing color sequences for visualization [War00]. He recommends a spiral in color space, so that as the sequence varies through a range of colors, the luminance increases as well. The monotonically increasing luminance facilitates the perception of forms. This model also addresses the perceptual problems caused by contrast. A patch of color may appear to have different luminance or to be a different hue when placed on different backgrounds. Figure 4.3 shows an example [Kai02]. The circles are all the same color, though we perceive the one on the far left to be brighter than the one on the far right. This effect, undesirable for visualizations, is called simultaneous contrast. Errors such as this are reduced by choosing a cyclic pattern in color space.

Figure 4.3: An example of simultaneous contrast: The circles are all the same color. The luminance of the grey squares behind the circles is affecting our perception of the circles' colors.

Healey and Enns describe a technique for choosing multiple colors to maximize the number of distinguishable colors that can be displayed [Hea96, Hea99]. They indicated a need to consider the three separating criteria: color distance, linear separation, and color category. Color distance is the Euclidean distance between two colors in a perceptually balanced color model. Linear separation requires every color to be separable by a single straight line from all others. A color's category means the name of the color region in which it resides in the model. Tests employing this method of color selection yielded successful results for rapid, accurate

color detection.

Our tool employs elements of each of the techniques described to map data to color effectively. Our colors are selected along a spiral about the luminance pole in the monitor's gamut within the CIE LUV color model. As in Ware's guidelines, we use an upward spiral, so that luminance increases as data values increase. Uniform sized groups of colors reside in different color categories, maintaining color name separation. The technique also reduces simultaneous contrast effects, since the colors are chosen cyclically. Overall, the result is a perceptually balanced selection of colors.

4.2 Texture Selection

Webster's dictionary defines texture as the visual or tactile surface characteristics and appearance of a surface or object. A surface may feel smooth or coarse and look spotted or striped, for example. In this context "visual texture" is a perceptual property analogous to color. Just as color can be described in terms of hue, luminance, and saturation, texture can be described by various properties, such as size, slant, density, and regularity.

Grinstein et al. use the texture properties shape, length, thickness, and slant to visualize data [Gri89]. They built a visualization tool called EXVIS (Exploratory Visualization). EXVIS maps data to glyphs called "stick-figure icons." The stick-figures have various basic configurations determined by the family to which they belong, i.e., they may have two, three, or four limbs on one end of the body attached in various ways. Data attributes mapped to parts affect each limb's length, width, slant, and color, so that data elements with similar attribute values

produce visually similar icons. When the icons are arranged spatially, spatial coherence and boundaries between groups of elements with different attribute values are visible. Figure 4.4 demonstrates the application of EXVIS to five-parameter data, weather satellite imagery of the Great Lakes region. The basic stick-figure icon is shown in the upper-right corner. The upper four segments are limbs. Properties of these limbs vary across the image to display the data. This visualization conveys boundaries patterns in data attributes.

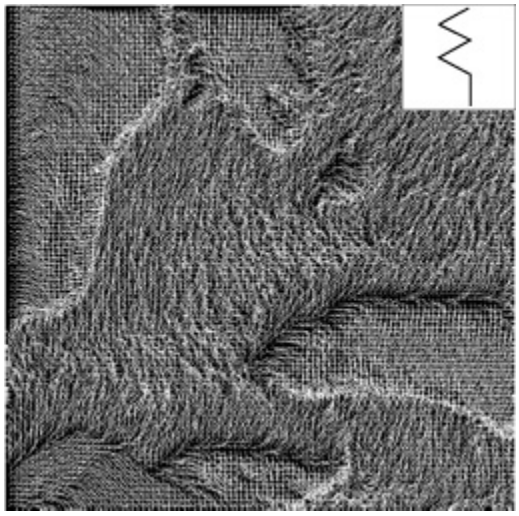


Figure 4.4: An image created with EXVIS. This iconographic image visualizes a five-attribute dataset: weather satellite imagery of the Great Lakes region. The insert shows the basic stick-figure icon that is varied across the image.

Scientists are interested in which texture properties and combinations of properties can be detected at a glance. This is called *preattentive detection*. An eye movement takes at least 200 ms. A visual task completed in less than 200 ms is considered to be processed preattentively. Researchers investigating the saliency of visual features typically conduct experiments in which viewers search for a particular element in a field of distracter elements. For example, the observer might attempt to find a diagonal element in a field of vertical elements.

Treisman ran experiments with grids of glyphs to test the preattentive processing of human vision [Tre85]. She tested skills like boundary detection and target detection. She suggested that early visual analysis results in separate mental maps for separate properties that pool their activity across locations allowing rapid access to information about the presence of a target. Ware and Knight developed a mathematical model based on neurological and psychophysical research to describe the orientation, size, and contrast of glyphs [War95]. Healey and Enns created visualizations of multidimensional data with glyphs varying in height, regularity, and density and then tested them for effectiveness [Hea00]. Though regularity is a commonly used texture property, it did not perform as well as expected in their experiments. However, results showed that visualizations employing height and density could be rapidly, accurately, and effortlessly analyzed. Weigle et al. focused specifically on texture orientation [Wei00]. Their research indicated that a target oriented $\pm 15^{\circ}$ or more from its background resulted in discrimination with high accuracy and fast response times.

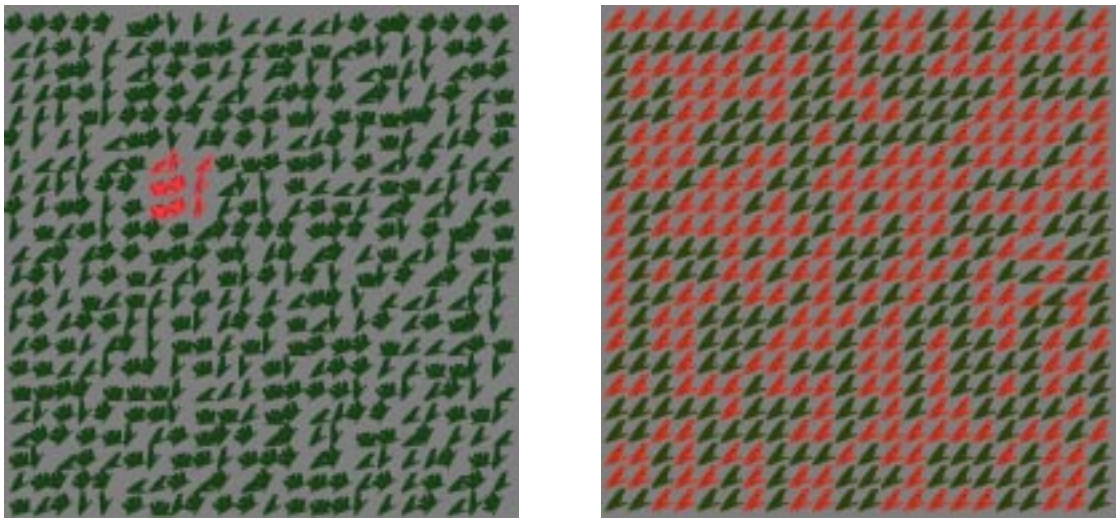
Our visualization tool employs several textural properties. Data is mapped to brush strokes varying in location, size, orientation, and coverage (percentage of canvas covered). Figure 4.5a shows a small patch of brush strokes varying in size, orientation, and coverage. Also, a texture map created from real brush strokes is mapped to each stroke. This serves two purposes: to synthesize painted brush strokes, and to increase discernibleness of features in densely packed monochromatic areas. Figure 4.5b shows the same sample before brush stroke texture was added.

(a) Strokes with texture mapping (b) Strokes without texture mapping

Figure 4.5: A patch of brush strokes generated by our tool: (a) and (b) show the same patch of strokes. But the stroke size and orientation are difficult to detect upper right corner of (b), where the strokes are monochromatic and densely packed.

4.3 Feature Hierarchy

The previous sections discussed issues like choosing distinguishable colors and perceptible orientation variations. Since we are visualizing multidimensional datasets, we are also interested in how visual features interact. In figure 4.5 the addition of brush stroke texture maps enhances the saliency of the size and orientation of each stroke. In other cases, additional features may interfere with the perception of existing ones. Evidence indicates that the low-level human vision system filters preattentive input, ordering visual features hierarchically. Researchers investigating feature hierarchy and interference typically conduct boundary or target detection experiments, similar to those described in the texture selection section. For example, figure 4.6 shows some textures with several varying features. In each, one feature is held constant over the entire field, except for a 3 x 3 target in which that feature is set to some other value. Different features vary in the background (e.g., orientation varies in figure 4.6a and color varies in figure 4.6b). Though the target in figure 4.6a is quickly evident, the viewer may need to conduct a serial search to detect the target figure 4.6b. Experimental results explain this phenomenon.



(a) Color target (b) Orientation target

Figure 4.6: Target detection examples: (a) The target of pink frogs pops out from a background of randomly oriented frogs. (b) The target of frogs oriented 5° does not pop out from the background of frogs oriented 20°. The target is masked by the random coloring.

The results indicate that color produces a small interference effect during texture segmentation [Sno98, Hea99]. Color variation interferes with an observer's ability to see textural regions. On the other hand, variations in texture have no effect on color region detection. Figure 4.6 demonstrates this asymmetric relationship. Even though texture orientation and shape vary randomly in figure 4.6a, the pink area can be detected preattentively. The color target jumps out despite the random texture variations of the frogs. But the tilted patch of frogs may be difficult to detect in figure 4.6b. All but the target are at a 20° angle, and the target orientation is 5°. Despite the 15° difference in orientation, the color pattern masks the texture property. The relationship between hue and luminance is similar to the one between texture and color. Luminance variations interfere with spatial patterns formed by hue, though random variations in hue have no effect on the viewer's ability to see luminance patterns [Cal90]. According to these results, in visualizations data attributes of most interest to the observer should be mapped

to luminance, then hue, and then various texture features.

Chapter 5

Effectiveness and Aesthetics Studies

Two main questions emerged as we developed our tool:

1. Would the perceptual guidelines discussed in the previous chapter apply to our painterly images?

2. Would observers find artistic merit in our computer-generated displays and what emotional and visual factors influence this judgment?

In conjunction with the psychology department at the University of British Columbia, we conducted two experiments to address these questions.

5.1 Effectiveness Studies

The effectiveness studies were designed to answer the first question by measuring speed and accuracy of preattentive perception. The tests were similar to the perception experiments described in the previous chapter, except that the textures used in our tests were patches of brush

strokes like the brush strokes used by our visualization tool. The brush stroke colors and textures varied according to which properties were being tested. Figures 5.1, 5.2, 5.3 and 5.4 show a few of the trials used in the experiment. Some contained a patch of strokes, called a target, with a unique color or orientation. The target was a randomly located 3x3 patch of strokes within a 22x22 patch. For example, Figure 5.1 has a target patch of green stokes in a pink background and Figure 5.3 has a target patch of $45°$ in a $30°$ background. Some trials like the ones shown in Figure 5.1f and 5.3f did not contain a target. Observers were shown each trial briefly (200 milliseconds), then asked to report the presence or absence of the target. The images were maximized on a 17 inch screen with observers sitting at a normal distance from the screen. The following sections describe the configuration of the texture patches in the two effectiveness studies.

5.1.1 Design

In experiment 1 observers were asked to report the presence or absence of a color target. Color, orientation, density, and regularity were varied over the brush strokes to test for feature interactions. The factor levels are listed in Table 5.1.1.

The trials shown in Figure 5.1 have constant orientation. Those in 5.2 have random orientation. Columns one, two, and three of figures 5.1 and 5.2 show dense, sparse, and very dense trials, respectively. The brush strokes are jittered in b, c, and d of both figures 5.1 and 5.2.

The color pairs, orange with pink and green with orange were chosen based on results from earlier studies that showed they were rapidly distinguishable from one another when no other visual features varied [Hea99, Hea00]. Displays were calibrated to ensure accurate color

Table 5.1: Four factors were varied to create the trials: a color, orientation, density, and regularity. Each factor has two or three settings. All combinations of levels of these factors generate 24 unique conditions.

Experiment 1			
Factors	**Levels**		
color	orange target in a pink background		green target in an orange background
orientations	constant		random
densities	sparse	dense	very dense
regularities	regular grid pattern		jittered randomly

reproduction. We used a Macintosh computer with a 24-bit color display.

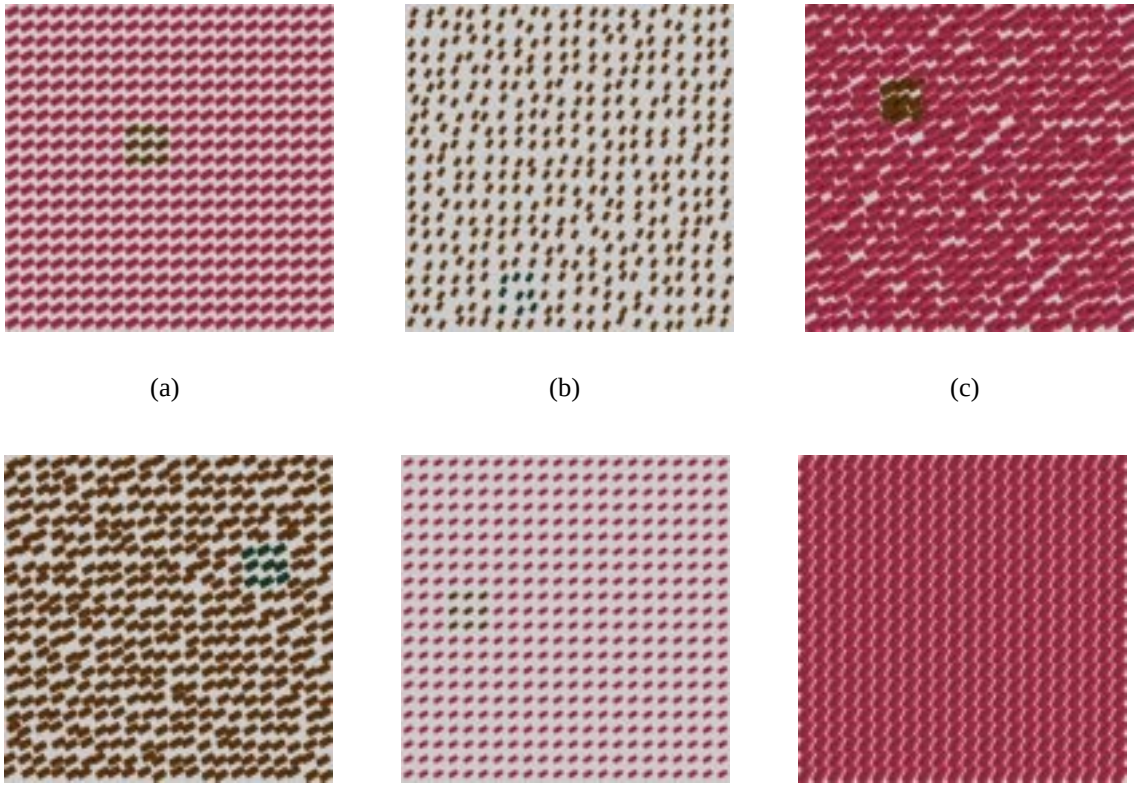
Each observer was shown 192 trials. Half of these contained a target. The other half didn't. These were created by varying the color (2 levels), the orientation[1] (2 levels), the density (3 levels), and the regularity (2 levels) to generate 24 unique conditions. We created 8 sets of these variations, 192 trials. For each condition the target was present in 4 of the trials and absent in the other 4 trials.

Experiment 2 was like experiment 1, except the roles of orientation and color were swapped (See Table 5.1.1). Color was a secondary feature, held constant or varied randomly[2] and targets were sets of strokes oriented differently from the rest. Observers attempted to detect the presence or absence of an orientation target.

Figures 5.3 and 5.4 shows some examples of the trials that were used. Figure 5.3 shows trials with constant color. Figure 5.4 shows trials with random color. Columns one, two, and three of figures 5.3 and 5.4 show dense, sparse, and very dense trials, respectively. The brush

[1]Constant orientation was a constant $45°$ angle or a constant $60°$ degree angle. Random orientation was randomly $30°$ and $45°$ or $45°$ and $60°$.

[2]Constant color means every stroke was the same color, either green or pink. Random color means the strokes were randomly colored green and orange or randomly colored orange and pink.



(a) (b) (c)

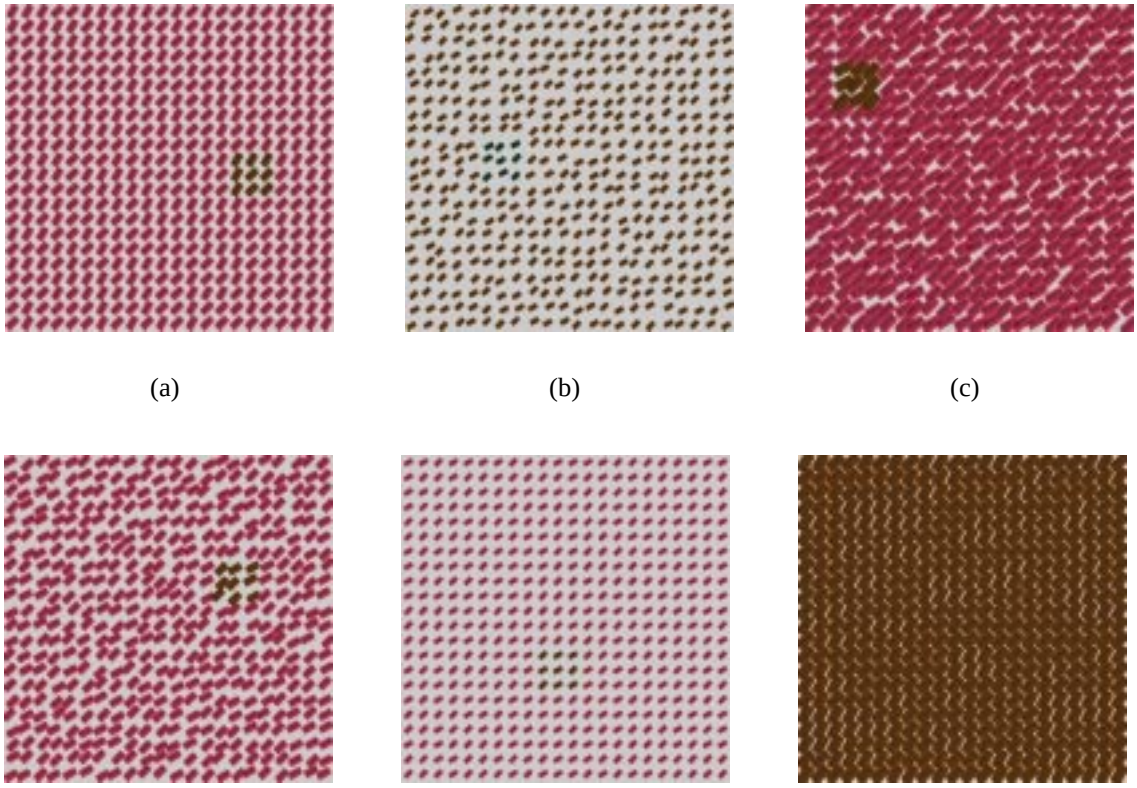
(d) (e) (f)

Figure 5.1: Examples of color target detection with constant orientation. (a)-(e) Green or orange target (f) target absent.

strokes are jittered in b, c, and d of both figures 5.3 and 5.4.

Orientation differences of 15° were chosen based on the texture orientation results described in the previous chapter [Wei00]. This research indicated that a target oriented $\pm 15^\circ$ or more from its background (with all other visual features held constant) was detected with high accuracy and fast response times. Again, 192 trials were created, half of which contained targets.

The participants and procedure were the same in both experiments. Eighteen observers (six males and 12 females ranging in age from 18 to 28) with normal or corrected acuity and normal color vision participated. Each observer met or exceeded our minimum accuracy requirement



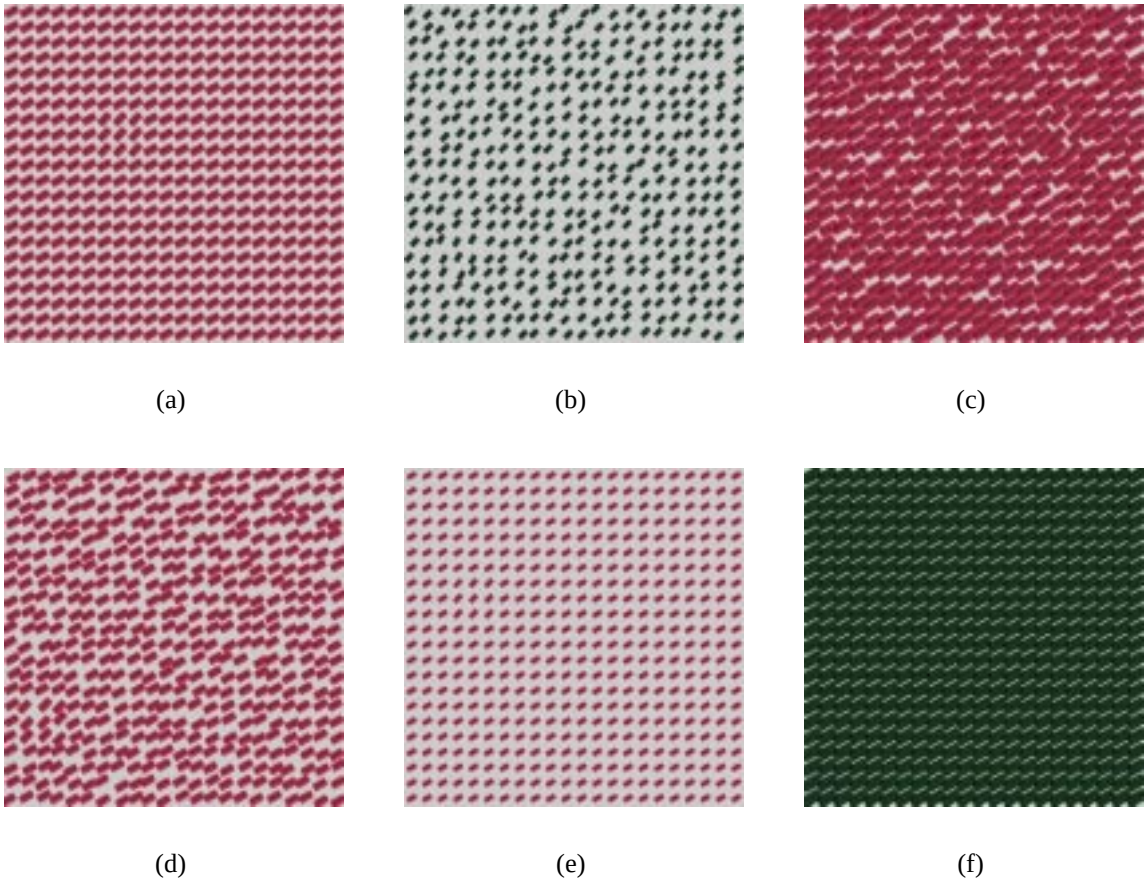
(a) (b) (c)

(d) (e) (f)

Figure 5.2: Examples of color target detection with random orientation. (a)-(e) Orange or green target (f) target absent.

(60%) on both trials.

Half the observers participated in experiment 1, followed by experiment 2. The other half did them in reverse order. Each completed a practice session of 24 trials before each experiment type (color or orientation). Observers were told that half of the trials would contain a target and half would not. Each trial was shown on the screen for 200 milliseconds. Then the screen was cleared and the system waited for a response of “target present” or “target absent” to be registered. Observers were instructed to respond quickly while maintaining a high level of accuracy, not sacrificing accuracy for time.



(a) (b) (c)

(d) (e) (f)

Figure 5.3: Examples of orientation target detection with constant color. (a)-(e) 45° target in a 30° background or a 60° target in a 45° background. (f) target absent.

5.1.2 Results

Response time, in milliseconds, and accuracy, 0 for incorrect or 1 for correct, were recorded for each trial. An *average response time*, t and an *average accuracy*, a, over all trials were computed for each experiment condition (i.e., for all trials where the choice of target and background was identical for all subjects who participated in the experiment). Preliminary tests showed a high inverse correlation between t and a (i.e.,an increase in t corresponded to a decrease in a), so we measured our results in terms of the ratio of t to a. This ratio is called the *inefficiency measure*, e. $e = t/a$ is commonly used when the direction of change in accu-

Table 5.2: The roles of color and orientation were swapped in experiment 2

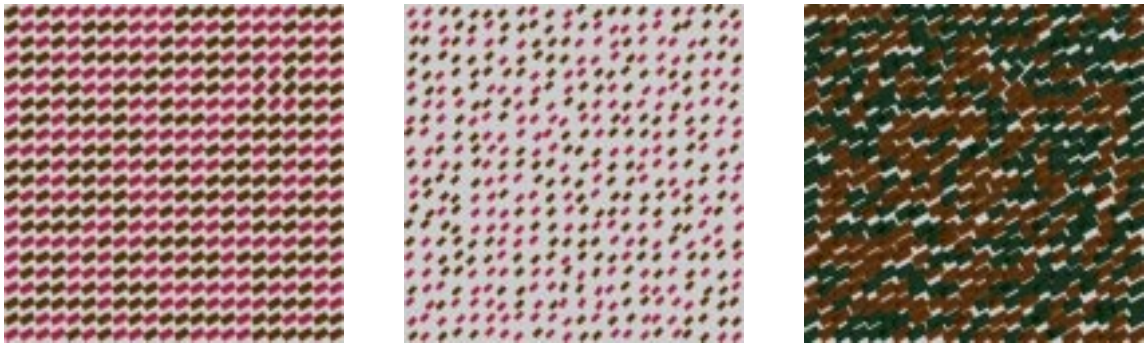
Experiment 2			
Factor	**Levels**		
color	constant	random	
orientations	45° target in a 30° background	60° target in a 45° background	
densities	sparse	dense	very dense
regularities	regular grid pattern	jittered randomly	

racy and the direction of change in time are consistent across experimental conditions. Notice that both an increase in time or a decrease in accuracy cause inefficiency to rise. Also, when accuracy is perfect (i.e., $a = 1$), $e = t$.

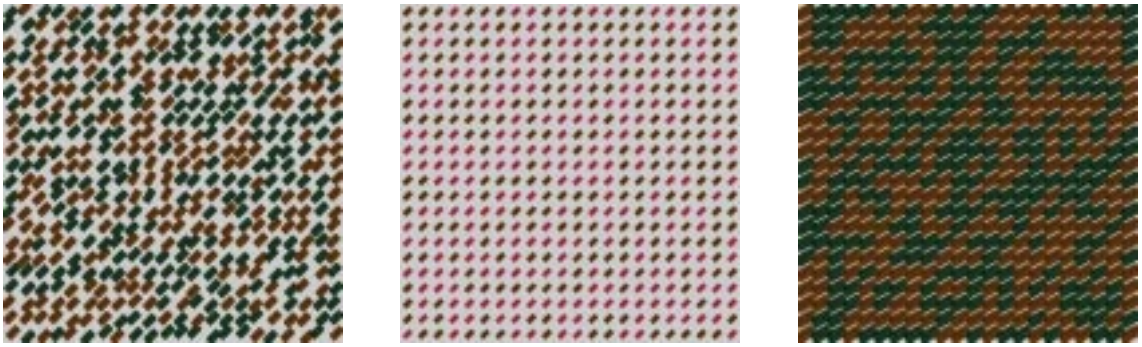
Average values of e for each condition are shown in Table 5.1.2. Target presence versus target absence is not listed, because preliminary analysis showed that this factor was not significantly related to our measures of performance. The target background pairings (i.e., orange target in pink background versus green target in an orange background, or 45° target in a 30° background 60° target in a 45° background) were not included for the same reason.

Looking at the table, we can see that color targets were easy to detect for all tests (average over all conditions, $e = 811.9$, $a = 91.1\%$). Orientation targets were easy to detect in a constant color background ($e = 1327.7$, $a = 71.9\%$), although performance was not as good as for color targets. Significant differences in mean values were identified using analysis of variance (ANOVA) tests.

The values of e are mean values of a finite sample in an infinite population. When comparing two different values of e, some analysis is necessary to decide if the population means are also different. Statisticians use ANOVAs to make this decision. We conducted ANOVAs



(a) (b) (c)



(d) (e) (f)

Figure 5.4: Examples of orientation target detection with constant color. (a)-(d) 45° target in a 30° background or a 60° target in a 45° background. (e)-(f) target absent.

on e using a standard 95% confidence interval. The ANOVA results for random background, variations in density, variations in regularity, and density $\times$ regularity interaction are listed here:

- *A random background had a significant effect on orientation targets, but not on color target performance.* Random orientation had no negative effects on color target detection ($F(1, 17) = 0.01, p < 0.94$, with $e = 813.7$, $a = 91.2\%$ for constant orientation, and $e = 810.2$, $a = 90.9\%$ for random orientation). Random color had a significant effect on orientation target detection ($F(1, 17) = 8.08, p < 0.05$, with $e = 1327.7$, $a = 71.9\%$ for

Table 5.3: Mean values of inefficiency measure $e = t/a$ under 24 different conditions (t = average response time for a treatment and a = average accuracy for a treatment). s, d, and v stand for sparse, dense, and very dense. The density is varying column-wise and the regularity varies row-wise.

Mean Inefficiencies for Color Targets						
	Constant Orientation			Random Orientation		
	s	d	v	s	d	v
regular	893.8	784.9	690.3	919.2	776.5	670.1
jittered	1007.3	770.1	735.4	995.3	763.4	736.6
Mean Inefficiencies for Orientation Targets						
	Constant Color			Random Color		
	s	d	v	s	d	v
regular	1276.6	1105.6	1046.1	1336.9	1477.9	1117.1
jittered	1699.0	1537.9	1264.9	1732.3	1409.3	1499.0

constant color and $e = 1437.8$, $a = 67.9\%$ for random color).

- *Density had a significant effect on both color and orientation targets* ($F(2, 34) = 30.84$, $p < .001$ for color targets and $F(2, 34) = 7.85$, $p < .01$ for orientation targets). Denser displays produced improvement in performance.

- *Regularity had a significant effect on both color and orientation targets* ($F(1, 17) = 5.10$, $p < .04$ for color targets and $F(1, 17) = 24.89$, $p < .001$ for orientation targets). Irregular displays caused a reduction in performance.

- *Density $\times$ regularity interaction had a significant effect on performance in color targets* ($F(2, 34) = 5.34$, $p < .01$) *and a marginally significant effect for orientation targets* ($F(2, 34) = 2.93$, $p < .07$). Variation in performance was larger for more difficult trials. For example, the negative effect of irregularity was larger in sparse color trials (like Figure 5.1b), compared to very dense color trials (like Figure 5.1c). Recall that

sparse trials were less efficient (i.e., more difficult) than dense trials. Similarly, the effect of density was larger in irregular color trials, compared to regular ones.

5.1.3 Interpretation

Recall that we set out to determine if the perceptual guidelines discussed in the previous chapter apply to our painterly images. Our results support this hypothesis. Feature hierarchy results indicated that color interfered with texture. Color outperformed orientation in our experiments ($F(1, 17) = 71.51, p < .01$). Also, psychophysical and visualization results show that random variation in color interfere with an observer's ability to see texture features, but variations in texture have no effect on color tasks. Thus, our results mirror the previous findings.

Another notable result was the improvement in performance of color and orientation target detection for increased density. We had been concerned that for brush strokes drawn closer together or overlapping, as in paintings, orientation might not be as easily discernible, especially for monochromatic patches. The results imply that our style of glyphs with brush stroke texture mappings facilitates discrimination of individual strokes in densely packed patches.

The list in the previous section also indicates a decline in performance for irregularity. We believe that trials with regularity may have provided the observer with a secondary clue, due to the underlying spatial arrangement of the glyphs. For example, compare Figure 5.3a with 5.3d. Both contain a 45° target in a 30° background, but in Figure 5.3a the presence of the target patch breaks the underlying regularity pattern, providing an additional visual signal of the target's presence. Jittering the strokes, like in Figure 5.3d, removes this extra cue. So we view the decline in performance for irregularity not as a measure of interference, but rather as

the removal of a secondary cue that regularity provides.

Our overall conclusion is that existing perceptual guidelines can be applied to construct perceptually salient data-feature mappings, even when used in a nonphotorealistic setting.

5.2 Aesthetic Judgment Studies

Our effectiveness studies confirm that we can follow existing perceptual guidelines to create effective mappings, but the question of aesthetic appeal still remains. The goals of our initial research on the aesthetics of visualization images are as follows:

- To determine if a group of relatively homogeneous observers find any artistic merit in our computer-generated displays.
- To identify the basic emotional and visual factors that influence these preferences.

If successful this would provide a solid foundation to guide further, more detailed studies on the effects of the individual differences and past experiences of the viewers.

Since it is generally accepted that art is beautiful and appealing when it evokes a sense of pleasure, an intuitive theory of aesthetics might begin with some measurement of emotional reaction. In fact, Berlyne proposed just such a theory when he tried to position aesthetic experience within a two dimensional model of emotionality [Berl70]. The axes of Berlyne's model represent hedonic value (or pleasure) and autonomic activity (or arousal). Others have since followed suit, for example the two dimensional pleasure-arousal model of affect proposed by Barrett and Russell (shown in Figure 5.5) [Bar99, Rus80]. In spite of these promising results, it

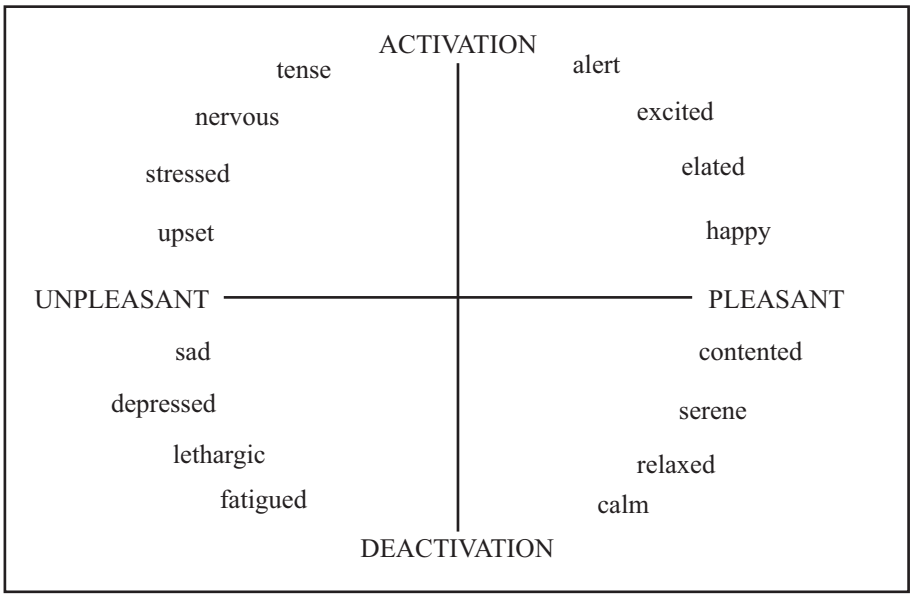


Figure 5.5: The structure of consciously experienced affective states modeled by Barret and Russell with two dimensions: pleasure and arousal. [**?**]

is generally agreed that there is more to visual aesthetics than emotionality alone; visual factors play a role. A study by Baltissen and Ostermann illustrates this point [Ost98]. These authors compared aesthetic judgments of famous artistic paintings and emotionally provocative photographs. They found that although emotionality was an important factor in the ratings of all the images, the judgments of the artistic paintings involved an additional cognitive-evaluative factor that is influenced by properties like meaning, familiarity, complexity, and interest. This finding is consistent with previous attempts to measure visual aesthetics using concepts like order and complexity [Bir33].

Based on these ideas, we chose to begin our study of the artistic merit of our visualizations by conducting a set of aesthetic judgment studies designed to study three important problems:

- How do observers rank images created with our computer painting method relative to paintings by master artists?

- Can we identify any fundamental emotional factors that predict when viewers will perceive an image to be artistic?

- Can we categorize individual viewers as preferring different types of art (e.g. realism or abstractionism), and how do these preferences impact the emotional responses that predict artistic rankings?

To investigate these questions we designed experiments in which observers were asked to assign numeric rankings to a collection of images taken from four separate sources: Impressionist works by famous masters, Abstractionist works by famous masters, nonphotorealistic data visualizations made by our computer algorithms, and nonphotorealistic renderings of natural scenes made by our computer algorithms. Each observer evaluated the images based on their artistic beauty, two emotional factors: pleasure, arousal, and two compositional factors: meaning, and complexity. Results were used to measure the level of artistic merit the observers attached to each type of image, and to identify the fundamental image properties that influenced these judgments.

5.2.1 Design

For each question, observers were asked to rank 28 images on a seven point scale (1 = lowest to 7 = highest). Seven images were presented from four different categories: nonphotorealistic visualizations (*visualization*), master Abstractionist works (*abstractionism*), painterly render-

ings (*nonphotorealism*, and master Impressionist works (*impressionism*) The nonphotorealistic visualizations were computer-generated visualizations of a weather dataset using the painterly techniques developed in this thesis. The initial visualizations look like weather maps with clearly defined landforms. To create the images we used here, we cropped the images so that no land formations were apparent, resulting in an abstract collection of strokes. Although real weather conditions are being represented (*temperature* by color, *wind speed* by coverage, *pressure* by size, and *precipitation* by orientation), no explanation was provided to an observer about what was being depicted. Figure 5.6a shows one of the visualizations used in the experiment. We paired these abstract visualizations against reproductions of seven paintings by Abstract masters:

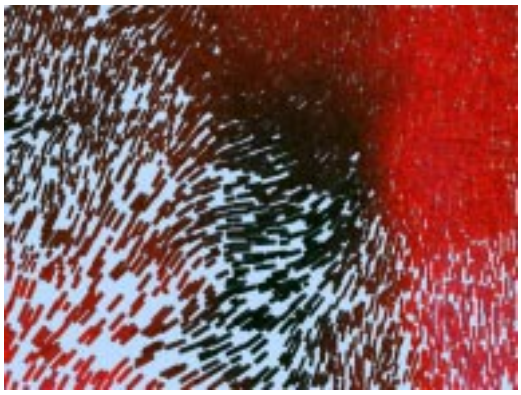
- *Untitled XX* by Willem de Kooning (Staatsgalerie, Stuttgart)
- *Corpse and Mirror* by Jasper Johns (University of Oklahoma Museum of Art)
- *Map* by Jasper Johns (Museum of Modern Art, New York)
- *Untitled, 1995* by Franz Klien (Kunstsammlung Nordrhein-Westfalen, Dusseldorf)
- *Untitled, 1916* by Kazmir Malevich (Peggy Guggenheim Collection)
- *Broadway Boogie-Woogie* by Piet Mondrian (Museum of Modern Art, New York)
- *Lavender Mist Number 1* by Jackson Pollock (National Gallery of Art, Washington, D.C.)

Since many of our painterly styles are derived from Impressionist paintings, we also included reproductions of the following seven paintings by Impressionist masters in our study:

- *Montagnes en Provénce (Mountains in Provénce)* by Paul Cezanne (National Gallery, London)
- *Water Lilies* by Claude Monet (The Art Institute of Chicago)
- *Cache-cache (Hide-and-Seek)* by Berthe Morisot (Mrs. John Hay Whitney Collection)
- *Les Chataigniers a Osny (The Chestnut Trees at Osny)* by Camille Pissarro (Private Collection, New Jersey) (See Figure 6.1.)
- *The Lighthouse at Honfleur* by George Seurat (National Gallery of Art, Washington, D.C.)
- *L'automne: Bords de la Siene pres Bougival (Autumn: Banks of the Seine near Bougival)* by Alfred Sisley (Museum of Fine Arts, Montréal)
- *Olive Trees with the Alpilles in the Background* by Vincent van Gogh (Mrs. John Hay Whitney Collection)

We generated painterly renderings to pair against these master works, by creating non-photorealistic renderings based on photographs. We applied our tool to the RGB data in the photographs. The resulting images are realistic in nature, since they portray a real underlying scene. Figure 5.6b shows one of the painterly renderings we used in our studies.

Twenty-five observers (6 males and 19 females aged 17 to 26) with normal or corrected vision participated. Each observer was given 28 printed images that they could spread out on a table and move around to compare. They were asked to rank the images five times based on five different questions that asked about *artistic beauty*, *pleasure* (how emotionally pleasing



(a) A painterly visualization

(b) A painterly rendering

Figure 5.6: The images used in the study were printed on 8.5×11-inch glossy film ink-jet paper at 1400×720 dots-per-inch resolution using an Epson 900N ink-jet printer. Example images: (a) A painterly visualization of a weather dataset. Temperature is mapped to color, wind speed to coverage, pressure to size, and precipitation to orientation. (b) A painterly rendering based on a photograph of Lake Moraine in Banff, Canada.

the images were), *arousal* (how active the images were), *meaning* (how meaningful the images were), and *complexity*. For example, during the ranking of artistic merit, observers were given the instructions below. The remaining four questions were framed in a similar manner.

> "As a first step, I would like you to look through this entire set of pictures in order to choose one picture that you like the best. This is a picture that you would like to place as art somewhere in your house or at your place of work. Its the one you think is the best example of 'good art.'
>
> Now look through the remaining pictures and choose the one that you think is the worst example of art.
>
> I would like you to go through the rest of these pictures in the order in which they come up and assign each one a number from 1 to 7. If the picture is as good as the one you chose to be 'best,' then give it a 7. If it is as bad as the one you chose to be 'worst,' give it a 1. If it is somewhere in between, then choose an appropriate number between 1 and 7. Remember, 7 represents the best art. Please use the range of numbers to the best of your ability."

5.2.2 Results

Each observer rating was classified by content of image (*realistic* or *abstract*), source of image (*master* or *computer generated*), and specific image within each category (arbitrarily numbered 1 through 7). Figure 5.7 shows overall rankings for each category. These results were tested for significance with a multi-factor analysis of variance (ANOVA) and multiple regression analysis (MRA). We used a standard 95% confidence interval to denote significant variation in mean ratings. In summary, our results showed:

- Viewers rated the paintings made by master artists and those with realistic content as generally more artistic overall, and as more pleasurable, more meaningful, and more complex.

- Viewers made an exception to this pattern when it came to the emotional quality of arousal. For this scale they rated the visualizations as most arousing and the nonphoto-realistic images as least arousing. This means that the pleasure and arousal scales were applied differentially by the viewers.

- Only one of the emotionality scales (pleasure) and one of the compositional (meaning) were needed to predict artistic beauty rankings with a high degree of accuracy. These two scales were themselves not highly correlated, meaning that viewers were distinguishing between their emotional and visual experiences when they chose their rankings.

- Viewers were not homogeneous in their artistic ranking of our images. A small subset of the viewers identified the visualizations as “better than average” relative to the nonphoto-

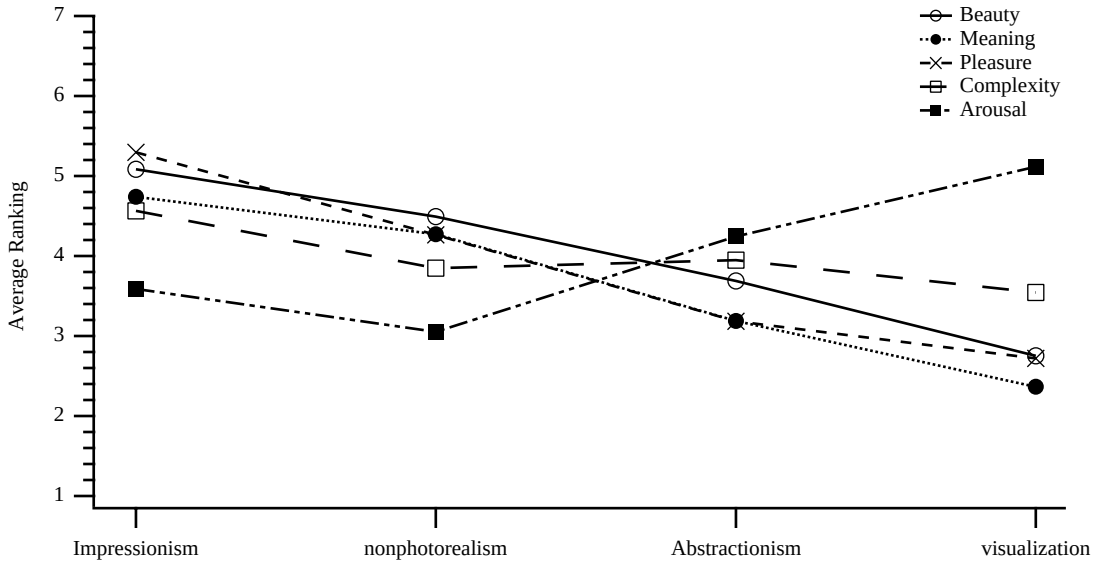


Figure 5.7: Summary of the aesthetic judgment results for the four image categories.

realistic images and master paintings. The emotional property of arousal influenced these viewers' artistic beauty judgments. The higher level of arousal for the visualizations was therefore a positive feature that improved their artistic beauty ranking.

5.2.3 Analysis

We first conducted ANOVA's to examine how the viewers rated each of the four categories. The dependent measure was the average ranking on a scale from 1 to 7. Categories were examined as a factorial design involving content (*realistic* or *abstract*) and source (*master* or *computer*). The mean rankings for artistic beauty are shown in Figure 5.8. The ANOVA showed that viewers ranked master paintings as more artistic than computer-generated paintings, $F(1, 24) = 24.56$, $p < .01$. Similarly, the ANOVA showed that viewers ranked master paintings as more pleasing and more meaningful than computer-generated paintings ($F(1, 24) = 24.56$,

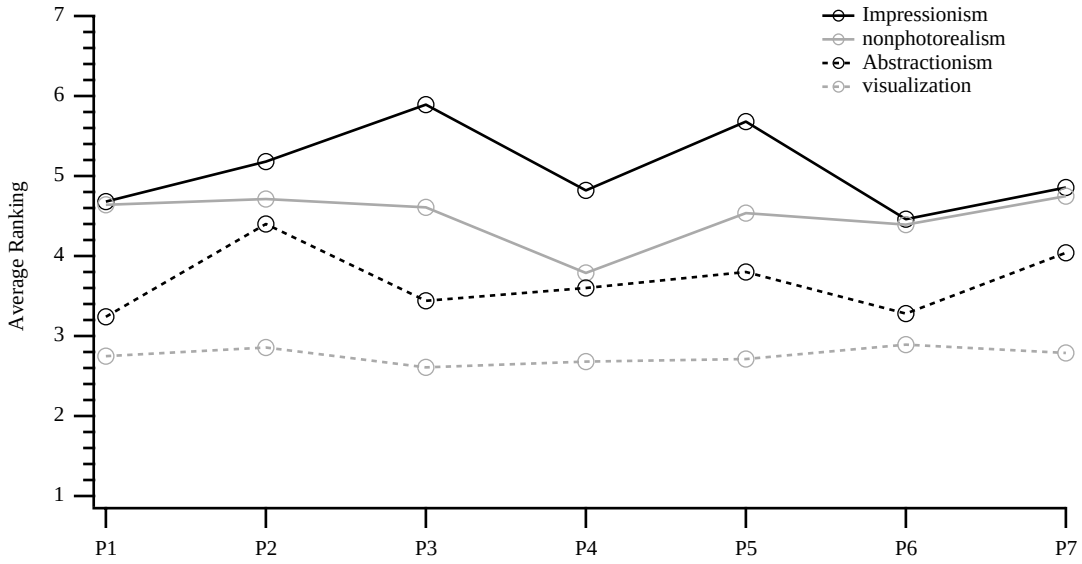


Figure 5.8: Artistic beauty rankings of the seven images for the four categories.

$p < .01$ and $F(1, 24) = 16.49$, $p < .01$, respectively), and that realistic content was judged more pleasant and more meaningful than abstract content ($F(1, 24) = 90.54$, $p < .01$ and $F(1, 22) = 43.79$, $p < .01$, respectively; see also Figures 5.10 and 5.11). Mean rankings for the arousal scale are shown in Figure 5.9. The ANOVA reported that viewers rated Impressionist and Abstractionist paintings as equally arousing, regardless of content, but that the visualizations were significantly more arousing than the nonphotorealistic images. This was reflected in a significant source $\times$ content interaction, $F(1, 24) = 23.20$, $p < .01$, in addition to a main effects of content, $F(1, 24) = 44.05$, $p < .01$. Finally, mean rankings for complexity are shown in Figure 5.12. Although there was less variation in these rankings than in the other scales, the ANOVA showed that viewers judged paintings by the masters as more complex than computer-generated paintings, $F(1, 24) = 8.73$, $p < .01$. There was no effect of content; realistic and abstract paintings were seen as equally complex.

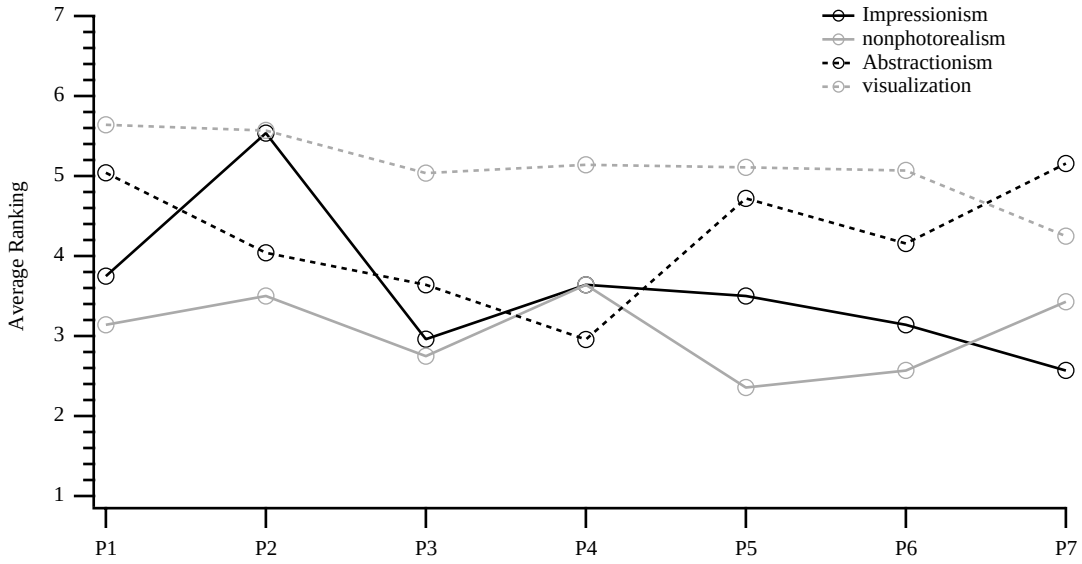


Figure 5.9: Arousal rankings of the seven images for the four categories.

5.2.4 Relationships Between Scales

Correlation analysis was used to make an initial inspection of the relationships between the two emotionality scales and the two visual composition scales. The independent measure was the mean ranking for each of the 28 different images, calculated across all four image categories. Results showed that pleasure and arousal were negatively correlated, $r = -.63$. This implies that paintings with high arousal rankings tended to be those with low pleasure rankings. The meaning and complexity ratings, on the other hand, were moderately correlated in the positive direction, $r = .55$. Paintings with high meaning also tended to be those with higher complexity.

When all four scales were tested as predictors of artistic beauty using a simultaneous multiple regression, they accounted for 91% of the variability in the rankings ($R\text{-}squared = .91$, $F(4, 23) = 58.01, p < .001$). An examination of the relative contributions of each of the four scales to the overall artistic beauty rankings revealed that only two made significant contri-

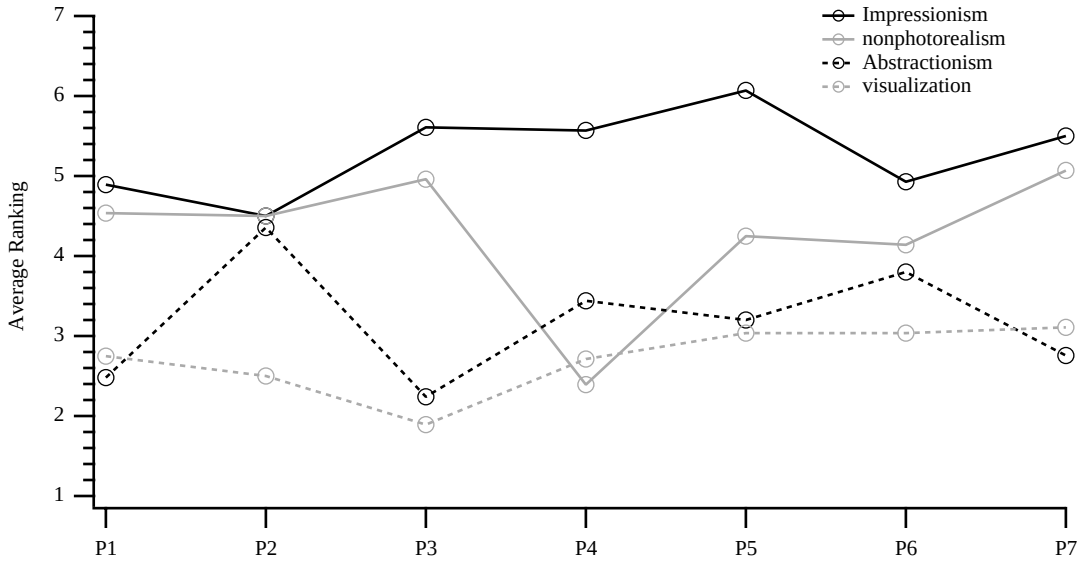


Figure 5.10: Pleasure rankings of the seven images for the four categories.

butions: pleasure (partial $r = .592$, $t(23) = 3.53$, $p < .01$) and meaning (partial $r = .450$, $t(23) = 2.42$, $p < .02$). When these two scales alone were used to predict the artistic beauty they accounted for almost the same amount of variability as when all four scales were included (*R-squared* $= .90$, $F(2, 25) = 118.86$, $p < .001$). Importantly, these two scales were themselves not highly correlated (partial $r = .122$). This indicates that these scales were used by the viewers to index different aspects of their artistic judgments.

5.2.5 Individual Differences

The preceding results were derived from treating all 25 viewers as an undifferentiated group. However, an inspection of the data revealed that there were clear differences in the way some groups of viewers experienced the images. For example, although most viewers (20 of 25) ranked the Impressionist images as "better than average" overall, there was a subset of viewers

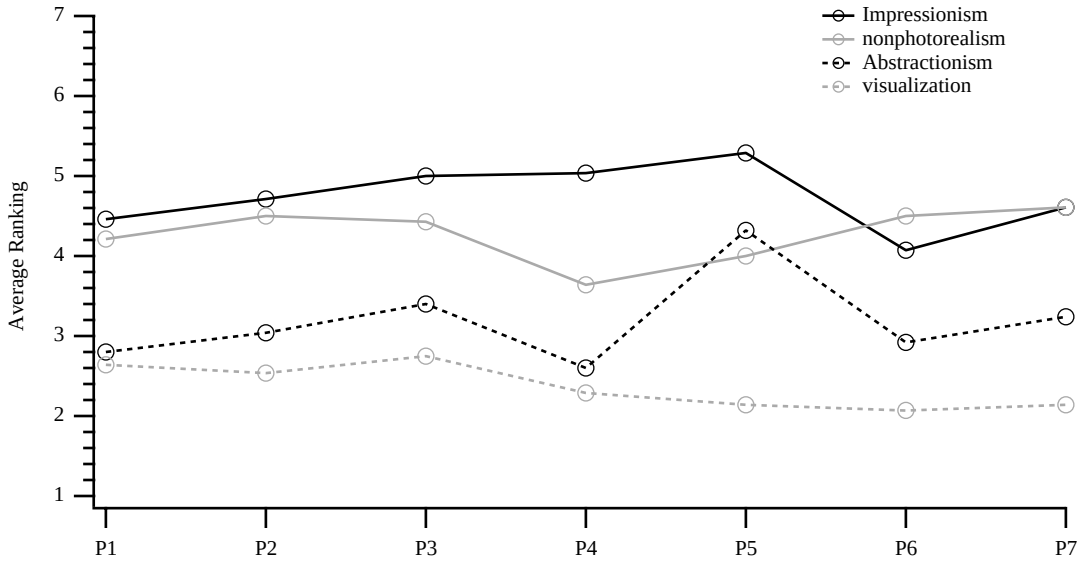


Figure 5.11: Meaningfulness rankings of the seven images for the four categories.

(6 of 25) who ranked the visualizations as "better than average art," and another subset (8 of 25) who rated the Abstractionist paintings as "better than average art." To examine the underlying factors behind these rankings, we repeated the MRA calculations for each subset.

The analysis for viewers who rated the visualizations as better than average revealed that the arousal rankings were able to predict artistic beauty rankings almost entirely on their own. Unlike the trend among viewers as a whole, this group had a positive correlation between arousal and artistic beauty. An MRA model that included all four scales as predictors had a modestly good fit ($\textit{R-squared} = .33$, $F(4, 23) = 2.80$, $p < .05$). Only the arousal scale made a significant contribution (partial $r = .510$, $t(23) = 2.81$, $p < .01$) A model that included only arousal was still a significant fit to the artistic beauty rankings ($\textit{R-squared} = .18$, $F(1, 26) = 5.21$, $p < .03$).

Finally, the analysis for viewers who ranked the Abstractionist paintings as better than

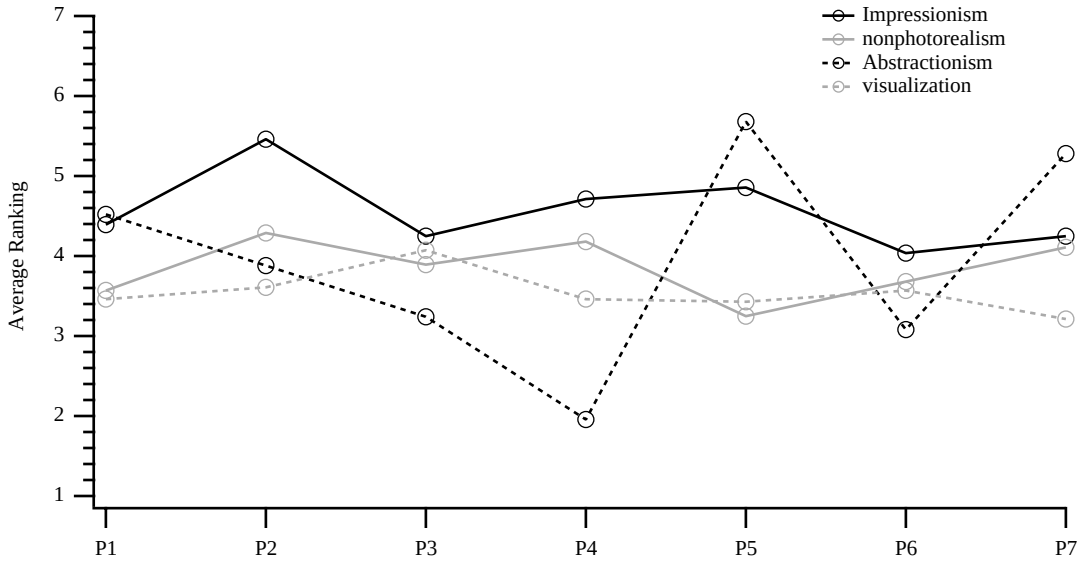


Figure 5.12: Complexity rankings of the seven images for the four categories.

average revealed that meaning was able to predict artistic beauty almost entirely on its own. The MRA model for this group of viewers that included all four scales as predictors had a modestly good fit ($\textit{R-squared} = .59,\ F(4, 23) = 8.21,\ p < .01$). Only the meaning scale made a significant contribution (partial $r = .550,\ t(23) = 3.16,\ p < .01$). The model that included only meaning was still a significant fit to the artistic beauty rankings ($\textit{R-squared} = .56,\ F(1, 26) = 32.58,\ p < .01$).

5.2.6 Interpretation

Our results indicate that it is possible to measure the relative aesthetic beauty of a set of images in a systematic and reliable way. Viewers' rankings of artistry were influenced reliably by both the source and the content of the images they were shown. Viewers tended to prefer images with realistic or recognizable content over more abstract patterns. They were also sensitive to

the skills of the artist. For example, nonphotorealistic images of natural scenes are not rated as artistic as paintings of similar scenes by the masters, even when they are both rendered with impressionistic brush strokes. Presumably this is because the master painter knows something about composition that the computer algorithm has not learned (or cannot learn) to capture.

In this context, our images based on data visualization fared least well in most viewers' judgments of artistic merit. This does not mean the visualizations were all seen as "poor art," but rather that many viewers preferred other types of images over the visualizations. The data hint at several reasons why this is so. First, a high degree of meaningfulness is important in most viewers' judgments of artistic beauty. The visualizations were rated among the lowest of all images on the meaning scale. Second, most viewers indicated that arousal was negatively correlated with their experience of artistic beauty. Since the visualizations were among the most arousing images, their artistic beauty rankings suffered as a result. Third, within the context of the 28 images tested, the seven data visualizations were the most homogeneous of the four categories. This may have had a negative effect on the rankings of this group as a whole. Further studies will be necessary to explore these hypotheses.

Chapter 6

Nonphotorealistic Visualization

In order to realize nonphotorealistic visualizations of real datasets, we implemented a visualization system based on our perceptual brush stroke model. Our visualizations consist of an underpainting, an overpainting, and a highlight layer. We created the first two layers using a segmentation algorithm and a painting algorithm. The highlight layer was created with an error diffusion algorithm. The next sections will describe the motivation for and steps of each algorithm.

6.1 Segmentation

To render a two-dimensional visualization of a dataset, we must first choose two data attributes to map to x and y coordinates. For spatial datasets these values are normally encoded in the data itself either explicitly or implicitly (e.g., by the elements' ordering). We divide the resulting two-dimensional space into 4×4 grids with a data sample at each corner to discretize our

canvas. Positions interior to each grid are assigned attribute values by linearly interpolating the known values at the corners. This scaling of the dataset is performed to construct an appropriately sized canvas on which to paint our brush strokes. Our algorithm uses the nonspatial data attributes to segment this canvas and place a set of brush strokes in each segment. The first challenge is segmenting the data.

In Impressionist paintings, the scene contents provide a natural subdivision of two-dimensional space into distinct regions. For example, in the painting by Pissarro shown in figure 6.1, the grassy area and tall chestnut trees in the foreground and the fields and ridges in the background act as delimiters.

Data values may also provide natural subdivisions, though care must be taken to create connected segments that are not undesirably coarse or fine. Figure 6.1a illustrates a coarse subdivision of Pissarros painting into foreground and background. The foreground is isolated in figure 6.1b. The background is isolated in figure 6.1c. An observer might view figure 6.1c as a collection of segments, since there are five disconnected portions. To avoid this ambiguity, we build connected segments, i.e., there exists a path from any point in the segment to any other point in the segment such that every point on the path is a segment member. Figure 6.1d shows the same painting more finely divided into twenty-one connected segments.

(a) Coarse Segmentation

(b) Foreground Segment

(c) Background Segments

(d) Fine Segmentation

Figure 6.1: *Les Chataigniers a Osny (The Chestnut Trees at Osny)* by Camille Pissarro. The scene can be thought of as being subdivided by the content. (a) A coarse segmentation of the scene outlined in white. The two segments are foreground and background. (b) The background segment isolated. (c) The foreground segment isolated. (d) Another possible segmentation (with finer granularity) of the scene outlined in white.

We use the following algorithm to subdivide a dataset with m non-spatial attributes into connected segments:

While unsegmented elements exist
  Start new segment R_k
  Select a starting element e that does not belong to any segment
  Initialize median vector med to e's attribute values
  Add e to queue
  While queue not empty
    Remove next item q from queue
    Add q to current segment
    For each neighbor nbr of q
      If $|nbr_i - med_i| \leq \delta_i \; \forall i = 1, ..., m$
        Add nbr to queue
        Update m based on nbr's attributes

The algorithm grows each segment around a starting element by considering its neighbors for membership. Neighbors are the eight adjacent elements. A neighbor is accepted if all its attribute values are "close enough" to the median vector values. An element that is accepted is added to the segment and its neighbors are then considered for membership. The algorithm ensures that the segments are connected by only considering neighbors of existing segment members. The segment stops growing when no more neighbors are eligible. If unsegmented elements still exist, the algorithm grows another segment in the same manner.

The membership criterion, "attribute values are *close enough* to the median vector values," is defined precisely by $|nbr_i - med_i| \leq \delta_i \; \forall i = 1, ..., m$, where nbr_i is the i^{th} attribute value of the neighbor element under consideration, med_i is the i^{th} median value, and δ_i is a user-defined percentage of the range of the i^{th} attribute. δ_i is fixed whereas med_i is updated as new elements are added. The granularity of the segments, which can be regulated by δ_i, can also be regulated by themethod used to update the median vector values. In the series of examples below, we segment a JPG file using different update methods to illustrate this point.

The datasets in practical applications are not usually JPG files, but using them as an example provides a lucid explanation, because the segments can be shown with color.

JPG files consist of pixels that encode three data values: red, green, and blue. Figure 6.2 shows the results of applying several different procedures for updating the median. The large image of poppy petals on the left was the basis for these segmentations. The top row of smaller images shows segments in grey, the results of four different median update procedures. The bottom row shows these segments laid over the original image.

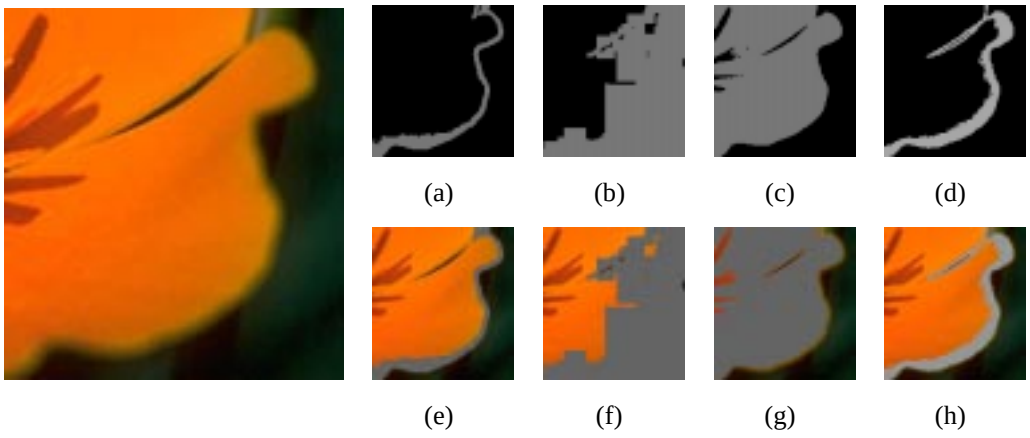


Figure 6.2: Examples of different segmentation algorithms applied to an RGB image of a golden poppy; (a,b,c,d) segments in grey with a fixed, running average, weighted average with $r = 1$, and weighted average with $r = \frac{7}{8}$, respectively; (e,f,g,h) segments overlaid on the original RGB image.

In the first example, we consider a fixed median, i.e., suppose the median is initialized, but not updated. Since the median vector is initialized to the starting element's values, the results of this method depend very heavily on the choice of initial element. Figures 6.2a and 6.2e show that using a fixed median can lead to a segmentation which is too fine. Since a randomly selected element may not be truly representative of the region, this method is too sensitive to the initial element. We adjust for this problem by allowing every new element to influence the

median.

A simple approach to allowing a new element to influence the median is to average the new element's attribute values with the current median:

$$med_i = \frac{med_i + nbr_i}{2}, \forall i = 1, ..., m. \tag{6.1}$$

With this method, the new element has a great effect on the median. In particular, the new element may tug m's value up or down enough so that e's neighbors are very likely to be included. The result is very coarse segmentation. Figures 6.2b and 6.2f show the results of using this method on the poppy JPG. We want the incoming elements to affect the median, but not as strongly.

The algorithm we implemented allows new elements to have a monotonically decreasing influence on the median vector. With this method the amount of influence wielded by each incoming element is determined by a variable, r. When the k^{th} element is being added, the median is updated as follows:

$$med_i = \frac{1}{\sum_{x=0}^{k-1} r^x}[r^0 e_{1,i} + r^1 e_{2,i} + ... + r^{k-1} e_{k,i}], \forall i = 1, ..., m \tag{6.2}$$

where $e_{j,i}$ is the value of the i^{th} attribute of the j^{th} member of the segment. Dividing by the geometric series, $\sum_{x=0}^{k-1} r^x$, ensures that med_i lies within the range of the data (i.e., $min_i \leq med_i \leq max_i$). $med_i \leq max_i$, since,

$$
\begin{aligned}
med_i &= \frac{1}{\sum_{x=0}^{k-1} r^x}[r^0 e_{1,i} + r^1 e_{2,i} + ... + r^{k-1} e_{k,i}] \\
&\leq \frac{1}{\sum_{x=0}^{k-1} r^x}[r^0 max_i + r^1 max_i + ... + r^{k-1} max_i] \\
&= \frac{1}{\sum_{x=0}^{k-1} r^x}[max_i][\sum_{x=0}^{k-1} r^x] \\
&= max_i.
\end{aligned}
\tag{6.3}
$$

An analogous argument holds for $min_i \leq med_i$.

Figures 6.2c, 6.2d, 6.2g, and 6.2h were created using this method. Figures 6.2c and 6.2g show an example in which $r = 1$. When $r = 1$, $\sum_{x=0}^{k-1} r^x = k$. So that at step k, med_i is the average of the k members ($med_i = \frac{1}{k}[r^0 e_{1,i} + r^1 e_{2,i} + ... + r^{k-1} e_{k,i}] = \frac{1}{k}[e_{1,i} + e_{2,i} + ... + e_{k,i}]$).

With $0 < r < 1$, the terms are monotonically decreasing. Elements that join late in the process wield less influence on the median values than early members. Since new elements are processed in order of proximity, the greater the Euclidean distance from the initial value, the less the impact on the median. This falloff averts the problem encountered in the method where the new elements pushed the median, stretching the segment too much. In figures 6.2d and 6.2h, $r = 0.875$. The falloff in this case means that the fifth element added has a 5.6% impact ($0.875^4 / \sum_{x=0}^{4} 0.875^x = .056$). The segment is much smaller than when $r = 1$. Yet the segment is larger than the one created with the constant method shown in figures 6.2a and 6.2e. These examples demonstrate that this update method balances the desire to reduce the importance of the choice of starting element and the need to temper the influence of incoming elements. The variable r also retains some flexibility for the outcome.

Finally, note that this segmentation algorithm can easily be extended to an arbitrary dataset with any number of attributes m. Each segment represents a collection of elements whose m

attribute values are “relatively similar” to one another. This is how we segment a multidimensional dataset.

6.2 Painting the Segments

Once we have segmented the data, we paint each segment by randomly placing brush strokes inside it, so that a certain percentage of the segment is covered. We assign one pixel in the final painting to each element in a segment. Therefore, coverage can be computed as the percentage of a segment’s pixels covered by brush strokes. In our procedure we have associated coverage with our segments; whereas, in the examples in figure 6.2, the segments were displayed with color. In paintings, the depiction of objects subdivides the scene. Artists reinforce these subdivisions with the visual features in each segment. In figure 6.1a the segmentation into foreground and background is a logical division of a scene, reinforced by the artist. Foreground strokes are well-defined and separately distinguishable. Those in the background blend together. The trees in the background are only distinguishable as overall shapes; whereas, individual branches are drawn in the foreground. In figure 6.1d, the grassy region in the foreground is separated from the chestnuts. The fields, ridges, and sky make up their own regions. Again, the artist communicates this segmentation by using colors, styles, and direction of brush strokes within each region that are varying, but relatively similar.

Visual features provide unification within segments of paintings. We use coverage to help to unify the members of a segment. Coverage is the percentage of pixels covered by brush strokes. Coverage is a property which is global to the entire segment, unlike visual features

such as size, color and orientation which vary locally for each brush stroke. The data-feature mapping M defines a data attribute to map to coverage. The average value of that attribute within segment R_k determines the desired coverage, C_k, for that segment. Coverage gives different segments individual definition, without creating the illusion of sharp changes where none exist.

To paint the data, we create a list of strokes for each segment. As strokes are added, the current coverage, c, is computed. This continues until the segment's coverage has been met. The following algorithm creates a stroke list for segment R_k:

While $c < C_k$
    Randomly pick an unpainted position p within R_k
    Identify element e_i associated with p
    Set size and orientation of new stroke s base on e_i
    Center s at p and scan convert
    Compute amount of s outside segment, $outside$
    Compute amount of s overlapping existing strokes, $overlap$
    If $outside$ or $overlap$ are too large
        Shrink s until it fits or it cannot be shrunk
    If s fits, add to stroke list and update c

To "paint" a segment, the algorithm randomly picks uncovered positions to place strokes within the segment until the desired coverage is met. A stroke s is not accepted if it overlaps existing strokes too much or if too much of s lies outside of the segment. To determine if the stroke is acceptable, it is centered on position p, scaled, rotated, and finally, scan converted. (The size, orientation, and color are determined by attribute values of the element associated with position p.) Scan converting s identifies which pixels it covers, allowing us to compute $outside$ and $overlap$. If either is too large, s is reduced in size until it fits or it can no longer be made smaller. If s fits, it is added to the stroke list. When the segmented layer is painted,

the strokes in the lists for each segment are rendered. Our painting algorithm distributes the strokes randomly within each segment and maintains an accurate representation by ensuring that strokes do not completely obscure one another and that the borders of the segments are respected (although some flexibility is permitted in both cases to achieve a painted look.

6.3 Underpainting

The segmentation algorithm and the painting algorithm are used to create the first two layers of our three layer display, the underpainting layer and overpainting layer. The underpainting is an addition inspired by the underpainting technique used by many artists. Monet, for instance, said that the first layer should cover the entire canvas to define the tonality of the scene [Per27]. We paint our entire canvas using a color not available in our overpainting. The underpainting is drawn first, and then the overpainting is drawn over top. The underpainting shows through the top coat in areas of low coverage and around the edges. Since stroke orientation and size are determined by the same attributes used in the overpainting, the underpainting reinforces information about these attributes. This information would otherwise be missing in low coverage sections of the overpainting.

The stroke lists for the underpainting are created using of the same segmentation computed for the overpainting. In the underpainting each segment is painted with a coverage of 100%. This means the entire canvas is covered by underpainting strokes (in our case, “the entire canvas” means the entire data sample area). Figure 6.4 shows how we set the tonality for the painting. The underpainting in figure 6.4a covers the sample area with strokes of varying size

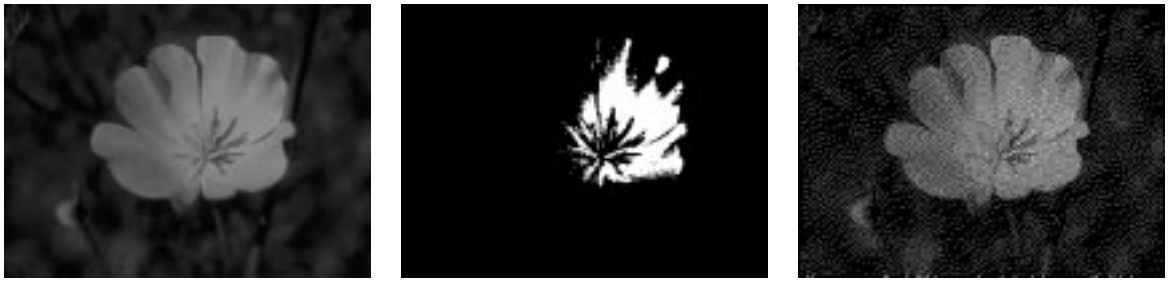
and orientation.

6.4 Highlighting Algorithm

The final layer in our painting is the highlight layer. This layer produces an additional visual feature designed to visualize a high spatial frequency attribute, i.e., an attribute whose values have a high rate of change within a given unit area of the canvas. We use a sprinkling of bright strokes across the canvas to display the trends in this attribute.

To create this layer we were faced with the challenge of producing a global pattern of highlights that accurately captures the local variations of an attribute. This problem is similar to halftoning in computer graphics. Halftoning produces approximations of greyscale image using only black and white dots (see examples in Figure 6.3b and 6.3c). This technique is commonly used in newspapers. When the dots are spaced closely together, they merge to form the appearance of a greyscale image. A naive solution to this problem uses a threshold: If a pixel is more than 50% white, display it as a white dot, otherwise display it as a black dot. Figure 6.3b shows the results of using a threshold method on the greyscale image in Figure 6.3a. Because this method approximates, without accounting for error, it gives poor results.

Error occurs for any pixel in the original greyscale image that is not completely black or completely white. For example, suppose we approximate a pixel that is 75% white with a white dot. This introduces a 25% error at that position in the image. Thresholding ignores these errors, producing local regions with an average ratio of black and white (i.e., greyscale level) that does not match the corresponding ratio in the original image.



(a) Original continuous greyscale image

(b) Threshold

(c) Floyd-Steinberg error diffusion

Figure 6.3: Halftone approximations.

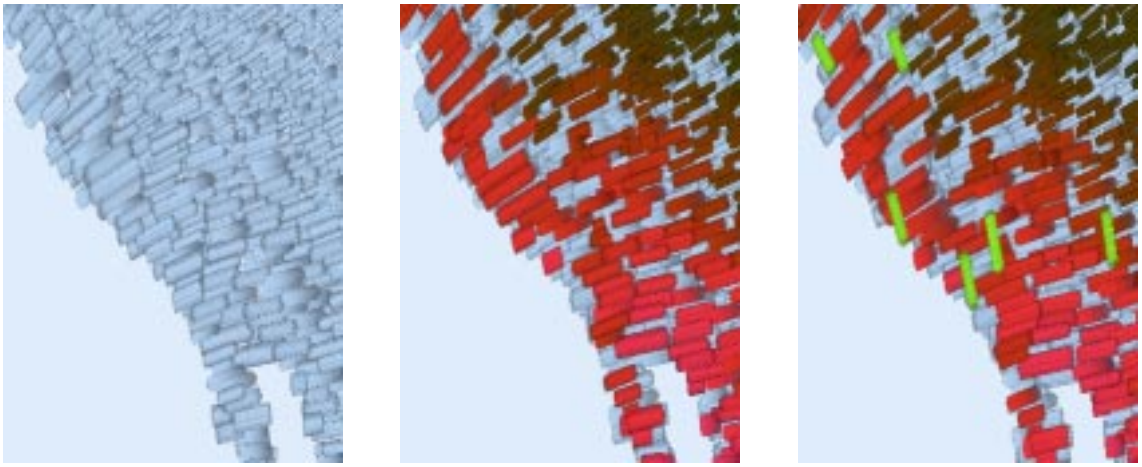
A better solution is to diffuse error forward during halftoning. Error diffusion algorithms address the issue of approximation by diffusing the rounding error across the data. Floyd-Steinberg, for example, sweeps from left to right and top to bottom across the images. As each pixel is approximated, any error that results is propagated forward to neighboring pixels that have not been processed yet (i.e., to the pixels to the right, directly below, to the bottom-left, and to the bottom-right). For example, if a pixel that was 75% white was approximated with a white dot, the 25% error would be propagated to the three unprocessed neighbors by darkening them by a combined total of 25%. This increases the likelihood that these pixels will be approximated with a black dot, thereby, maintaining the proper greyscale in the local area around the current pixel. This produces a much more accurate representation of the original image (see Figure 6.3c, for example).

Choosing locations for highlight strokes is very much like halftoning. Halftoning decides where to place black and white dots to properly recreate a target greyscale image. Highlight strokes must be placed to accurately represent an underlying data attribute. We used Floyd-Steinberg error diffusion to solve our problem. The algorithm can be applied almost directly to

normalized data attribute values, just as it applies to greyscale data. Given a grid of normalized attribute values, Floyd-Steinberg traverses this grid and adds highlight strokes at high attribute value positions exactly like it adds white dots to light pixel positions of a greyscale image. The main difference lies in the intended use of our display. We desire only a small number of highlight strokes, so that the layers below are not obscured and so that the highlights retain their ability to stand out. To achieve this sparseness, we multiply the normalized attribute values by a fractional constant h, so that each value begins on the range between zero and h. This would be equivalent to lightening an image prior to halftoning. More white space would appear in the resulting halftone. Figure 6.6 shows the results of varying h on a visualization. The number of highlight strokes increases as h increases.

When the highlight stroke list is complete, the stroke positions are jittered, so that they do not lie exactly on a regular grid. Highlight strokes are representative of the general trend in the attribute mapped to the highlight layer. Drawing strokes exactly on a grid could imply the existence of some horizontal or vertical boundaries; whereas, the grid merely functions as a convenient tool for processing the data.

The same attributes that control size and orientation in the overpainting coat are mapped to size and orientation of the highlight strokes. The rotation is then incremented by 90° so that the highlight strokes produce cross-hatching. The highlight strokes are drawn on top of the underpainting and overpainting in a bright color not used by the other layers. Figure 6.4 demonstrates the process. Figure 6.4a shows the underpainting. The overpainting is added in figure 6.4b. Finally, the highlight layer is added in figure 6.4c. In this application the highlights indicate high levels of precipitation. Figure 6.4 represents average weather conditions for De-



(a) Underpainting (b) Overpainting added (c) Highlights added

Figure 6.4: December weather data on the southern California coastline. The matte light blue on the left is the Pacific Ocean. Three layers are painted: (a) the underpainting covers the land mass where data samples were taken, (b) the overpainting encodes is painted next with coverage varying according to data values, and (c) the highlight layer is painted, providing an additional visual feature, creating the finished visualization.

cember in southern California. The weather data visualized here is described in more detail in section 6.5.

6.5 Practical Application

This section describes an application of our tool to a weather dataset. Weather data is a convenient example, since real data is plentiful and weather attributes, such as temperature, precipitation, wind speed, and pressure, are commoly used in day-to-day settings. This dataset, collected by the International Panel on Climate Change between the years 1961 and 1990, consists of 30 year mean monthly weather conditions. Figure 6.5 shows an example where March mean weather conditions (calculated as the averages of March conditions between 1961 and 1990) are visualized. *Mean temperature, precipitation, wind speed, pressure,* and *wet day*

frequency are visualized separately in figures 6.5a-6.5e using a simple coloring, ranging from bright pink for high values to dark green for low values. This visualizes the patterns of each attribute in isolation. In the largest map, temperature, precipitation, wind speed, pressure, and wet day frequency are displayed simultaneously with the following mapping:

$temperature \rightarrow$ color $\in$ [dark green,...,bright pink]

$precipitation \rightarrow$ highlighting $\in$ [absent, present]

$wind\ speed \rightarrow$ coverage $\in$ [low,...,high]

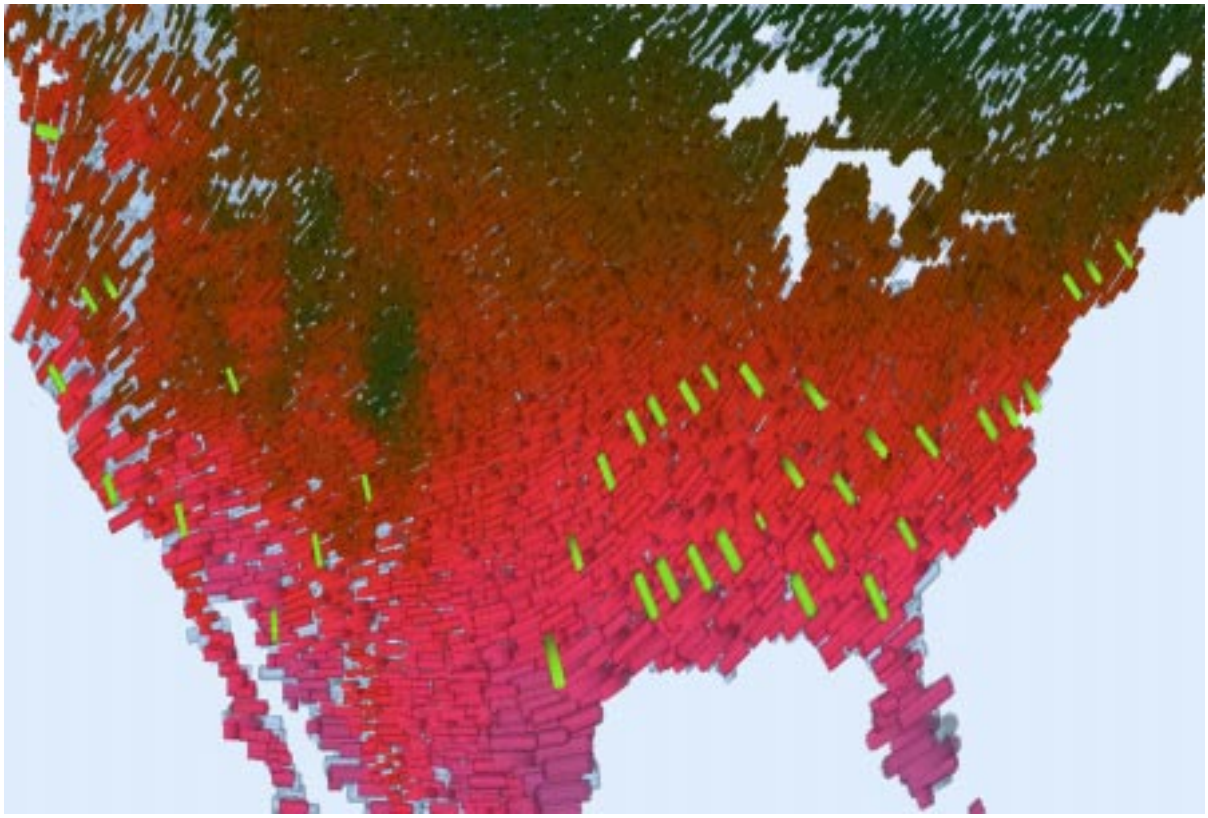
$pressure \rightarrow$ size $\in$ [small,...,large]

$wet\ day\ frequency \rightarrow$ orientation $\in$ [horizontal,...,vertical]

The colors range from bright pink in the south to dark green in the north. Though March weather is already quite warm in the south, the northern temperatures remain cold. $Temperature$ rises consistently from north to south, with the exception of some green spots in the West. The Rocky Mountains explain these cold areas.

In figure 6.5f, $precipitation$ is represented by green highlight strokes. Figure 6.6a magnifies the precipitation pattern shown in figure 6.5b. The same highlight strokes used in figure 6.5f are also included in figure 6.6a, so that both background color and highlights represent precipitation. The highest precipitation occurs in the Northwest, where this map is bright pink. Yellow areas in the Southeast also indicate a region with high levels of rainfall relative to surrounding areas. The amount of highlight stroke coverage is limited by the variable, h, discussed in section 6.4. Figures 6.5f and 6.6a were created with $h = 12\%$. Figures 6.6b and 6.6c show

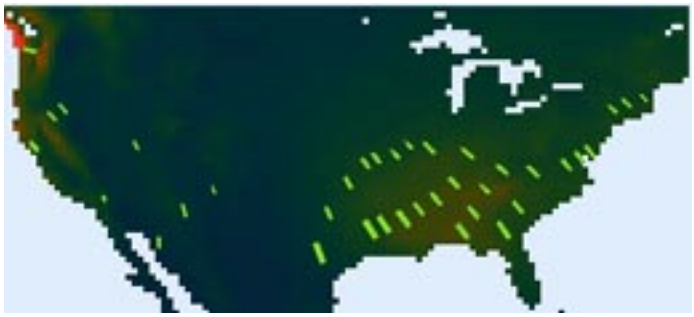
(a) *Temperature* (b) *Precipitation* (c) *Wind speed* (d) *Pressure* (e) *Wet day frequency*



(f)

Figure 6.5: Mean March weather conditions in North America. (a)-(e) visualize one attribute each with color. (f) visualizes the five attributes simultaneously, mapping *temperature*, *precipitation*, *wind speed*, *pressure*, and *wet day frequency* to color, highlighting, coverage, size, and orientation, respectively.

the change in highlight coverage as a result of varying h. The value of h was decreased by 5% in Figure 6.6b and increased by 5% in Figure 6.6c. h affects the sensitivity of the highlight process. Higher values of h allow lower values of the attribute to trigger highlight stroking. Thus, the number of highlight strokes is directly proportional to h for a given attribute.



(a) $h = 12\%$ (the same as in figure 6.5f)

(b) $h = 7\%$

(c) $h = 17\%$

Figure 6.6: All three maps show March precipitation levels represented by both background color and highlights, with different highlight coverage, h in a, b, and c. Colors range from bright pink for very high precipitation to dark green for low precipitation. The number of highlight strokes generated by high precipitation areas is affected by the value of h.

Wind speed is represented by coverage in figure 6.5f. The coverage is the percentage of the canvas covered by the overpainting layer, i.e., coverage is lower where more of the underpainting is visible. The high mid-continent coverage corresponds to high wind speed (also seen as bright pink patches in Figure 6.5c). The coverage is lower in the areas with lower wind speed in the west and northeast (shown as dark green in the windspeed map in Figure 6.5c.

Stroke size is used to represent *pressure*. High pressure areas appear on the southern coasts and Florida. Figure 6.5f indicates these high pressure areas with large strokes (red areas in Figure 6.5d.) The stroke sizes decrease towards the north, where the pressure is low.

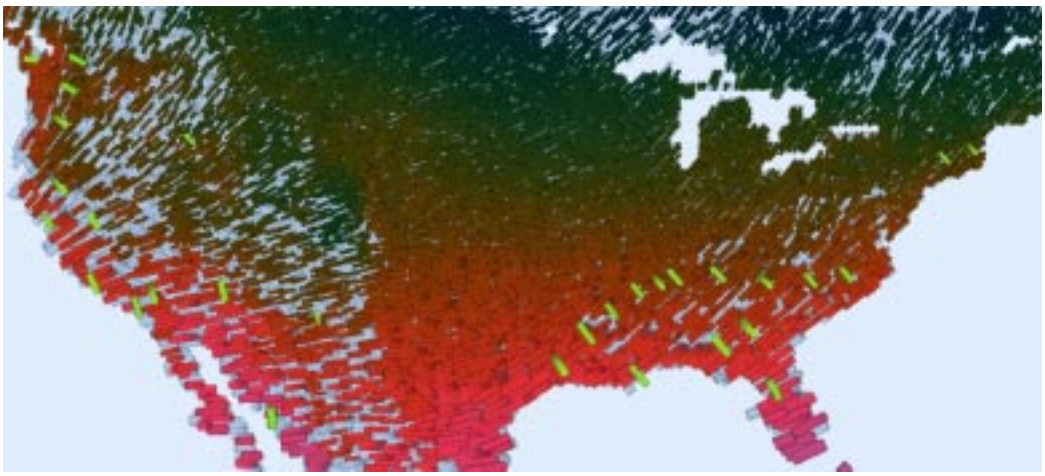
Higher wet day frequency appears in the northeast and further north in the west. Figure 6.5f represents this pattern with the orientation of the brush strokes. Horizontal strokes in the south represent low wet day frequency (shown in green in Figure 6.5e); whereas, strokes are close to vertical in the higher wet day frequency areas (shown in pink and red in Figure 6.5e).

Figure 6.5 shows the early spring weather conditions. Figure 6.7 visualizes winter and summer conditions from the same dataset. Figure 6.7a and 6.7b show January and July weather conditions, respectively. These visualizations apply the same mapping as the one used to create figure 6.5f.

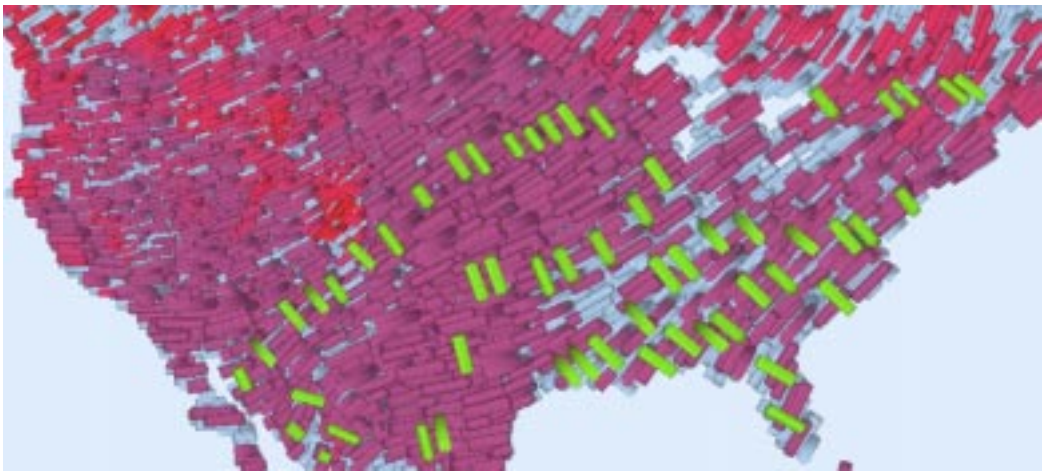
Figure 6.7a shows the cooler weather covering more of the land with darker colors stretching further south. Precipitation patterns, shown by highlights, are similar to those seen in March. Wind speeds are high mid-continent. The east coast shows lower wind speed in January than in March. Wind speeds north of the Great Lakes and west of the Rockies are even lower than those on the east coast. The low coverage in these regions indicates these conditions. Low wet day frequency dominates the west coast with the exception of a high wet day frequency in Washington state, where the strokes are drawn vertically. Pressure is fairly low, except in Florida which is covered with larger strokes.

With respect to figure 6.7a, the strokes in figure 6.7b are mostly larger, indicating a higher pressure in July than in January. The overall pinkish color shows that the temperatures are much higher at every data sample, and precipitation is higher south and east of the Rockies. The

Rockies, colored a dark pink, are still cooler than surrounding regions. Wind speeds are quite low in some portions of the east and northeast where coverage is very low. *Wet day frequency* shows a sharp change in the southwest, with vertical strokes (high wet day frequency) on the coast changing abruptly to horizontal strokes (low wet day frequency) as they move away from the coast. Visualizations such as these could provide experts in climate change with insights into patterns and anomalies that might not otherwise be perceived. Additionally, they could provide a means of communicating trends to non-expert observers.



(a) January weather conditions



(b) July weather conditions

Figure 6.7: Visualizations of winter and summer conditions: The mapping is the same as the one used to create figure 6.5f: $temperature \rightarrow$ color $\in$ [dark green,...,bright pink], $precipitation \rightarrow$ highlighting $\in$ [absent, present], $windspeed \rightarrow$ coverage $\in$ [low,...,high], $pressure \rightarrow$ size $\in$ [small,...,large], and $wet\ day\ frequency \rightarrow$ orientation $\in$ [horizontal,...,vertical].

Chapter 7

Conclusions

We have created a new visualization method that uses “painted” brush strokes to represent the data elements of large multidimensional datasets. In the formulation of this project, we identified three desirable characteristics for our visualizations: effectiveness, multidimensionality, and aesthetic appeal. To this end, we created nonphotorealistic visualizations influenced considerably by Impressionism, and tailored to the strengths of human visual perception. We conducted studies to objectively measure the effectiveness and aesthetic appeal of our visualizations (multidimensionality is inherently linked with both, since introducing higher dimensions to a visualization increases the complexity of both issues). The remaining sections discuss our progress toward each goal and directions for future work.

7.1 Effectiveness

Experimental results support the hypothesis that existing guidelines on human visual perception of color and texture hold for our painterly styles. This means that we can create effective visualizations by employing these guidelines to map data attributes to the visual features of our brush strokes. As we continue to develop and implement new painterly techniques, further testing may be required to ensure continued compliance with the guidelines.

7.2 Multidimensionality

Our nonphotorealistic visualizations map data element attributes to the physical features of brush strokes. Thus, we can visualize multiple data attributes (five non-spatial attributes per element).

Our system currently supports variations in the color, size, orientation, and coverage of brush strokes, as well as undercoating and highlight layers. We would like to build upon our current system by identifying some additional visual features that can be used to effectively display data attributes. To this end, we are pursuing two complementary approaches: (1) examining visual features that are known to be perceptually salient that may correspond to new painterly properties, and (2) reviewing literature on technical and stylistic characteristics of Impressionist art.

The brush strokes in Impressionist paintings offer a rich variety of styles. Figure 7.1 offers a close view of an acrylic Impressionist-style painting. Some brush strokes are long and curved; others are rounded and staccato. The paint is thinly applied in some areas, while other areas

Figure 7.1: A close-up of an acrylic painting that demonstrates the variety brush strokes in an Impressionist painting. The entire painting is shown in the upper left corner. This piece was painted by the author.

show heavier strokes, with thick blobs of paint. Some strokes are course and grainy, while others are smooth. We are considering adding curved strokes, coarseness, and weight to our system.

These additions may be difficult to implement using the simple texture map that we currently apply. We may need to take a different approach to modeling our brush strokes. These are some of the alternatives that we are considering: (1) creating a larger library of texture mapped brush strokes that explicitly vary the styles that are not easy to modify within an individual brush stroke, (2) modeling the brush strokes with spline surfaces to construct continuous representation of the multiple styles in a brush stroke, and (3) modeling the brush strokes with a physical simulation to vary painterly styles and construct visually realistic strokes.

7.3 Aesthetic Appeal

Our results show that some observers identified our visualizations as "better than average" relative to nonphotorealistic images and master Abstract and Impressionist paintings, though they fared least well in most viewers' judgments of artist merit. We are interested in learning more about the properties that influence viewers' judgments. Preliminary analysis of the data indicates that we should begin by exploring two factors:

1. In the context of the 28 images, the seven data visualizations were the most homogeneous of the four categories. We would like to explore the effect of presenting a more heterogeneous set of data visualizations. To do so, we can visualize datasets with sharp discontinuities. Visualizations of such data would exhibit abrupt changes within the image.

2. Many viewers felt highly artistic paintings were also highly meaningful. We will investigate the effect of increasing the meaningfulness of the nonphotorealistic visualizations. By informing the viewers that the images are data visualizations and by describing the dataset and data mappings to the observers, we hope to put the images in a meaningful context. This will allow us to test for increased meaning rankings for the visualizations, and for any corresponding increases in artistic beauty that may result.

Bibliography

[Ado02] Adobe Products (2002). Color Models. *Adobe Technical Guides* retrieved from http://www.adobe.com/support/techguides/color/colormodels/ on [8/2/02].

[Ash01] Ashikhmin, M. (2001). Synthesizing Natural Textures. *The proceedings of 2001 ACM Symposium on Interactive 3D Graphics* p.217-226.

[Bar99] Barrett, L.F., and Russell, J.A. (1999). The structure of current affect: Controversies and emerging consensus. *Current Directions in Psychological Science*, Vol.8, No.1 p.10-14.

[Berg95] Bergman, L.D., B.E. Rogowitz, and L.A. Treinish (1995). A Rule-based Tool for Assisting Colormap Selection. *Proceedings Visualization '95,* p118-125.

[Berl70] Berlyne, D.E.(1971). *Aesthetics and Psychobiology*, Appleton-Century-Crofts, New York, NY.

[Bir33] Birkhoff G.D. (1933). *Aesthetic Measure*, Harvard University Press, Cambridge, Massachusetts.

[Bro50] Brown, R. (1950). Impressionist Technique: Pisarro's Optical Mixture in *Impressionism in Perspective,* B.E. White, Ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978, p.114-121.

[Cal90] Callaghan, T.C. (1990). Interference and Dominance in Texture Segregation. In *Visual Search,* D. Brogan, E. Publisher: Taylor and Francis, New York, New York, 1990, p.81-87.

[Cha01] Chang, J. (2001). A Perceptual Visualization Assistant for Multi-dimensional Data Visualization. Master's Thesis, North Carolina State University.

[Che67] Chevreul, M.E. (1967). *The Principles of Harmony and Contrast of Colors and Their Applications to the Arts*, Schiffrel Publishing Limited, West Chester, PA.

[Cur97] Curtis, C.J., S.E. Anderson, J.E. Seims, K.W. Fleisher, and D.H. Salesin (1997). Computer-generated Watercolor. *SIGGRAPH 97 Conference Proceedings,* p.421-430.

[Fol97] Foley, J.D., A.van Dam, S.K. Feiner, and J.F. Hughes (1997). *Computer Graphics Principles and Practice,* Addison-Wesley Publishing Company.

[Ger94] Gershon, N.D., R.M. Friedhoff, J. Grass, R. Langridge, H.P. Meinzer, and J.D. Pearlman (1994). Is Visualization REALLY necessary? The Role of Visualization in Science, Engineering, and Medicine *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques,* Vol. 10, Issue 1, p.499-500.

[Goo01] Gooch, B. and A. Gooch (2001). *Non-photorealistic Rendering,* A K Peters, Ltd.,Natick, Massachusetts, 2001.

[Gri89] Grinstein, G., R.M. Pickett, and M.G. Williams (1989). An Exploratory Data Visualization Environment. *Proceedings Graphics Interface '89,* p.254-259.

[Hae93] Haeberli, P. (1993). Texture Mapping as a Fundamental Drawing Primitive. *Fourth Workshop on Rendering,* p.259-266.

[Hea96] Healey, C.G. (1996). Choosing Effective Colours for Data Visualization. *Proceedings IEEE Visualization,* p.363-270.

[Hea00] Healey, C.G. and J.T. Enns (2000). Building Perceptual Textures to Visualize Multidimensional Datasets. *Proceedings Visualization '98,* p.111-118.

[Hea99] Healey, C.G. (1999). Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. *IEEE Transactions on Visualization and Computer Graphics,* Vol.5, No 2, p.145-167.

[Hea01] Healey, C.G. (2001). Formalizing Artistic Techniques and Scientific Visualization for Painted Renditions of Complex Information Spaces*International Joint Conference on Artificial Intelligence,* Vol. 1, p.371-376.

[Hea02] Healey, C.G. and J.T. Enns (2002). Perception and Painting: A Search for Effective, Engaging Visualizations. *IEEE Computer Graphics and Applications,* Vol.22, No.2, p.10-15.

[Her00] Hertzmann, A. (1998). Painterly Rendering with Curved Brush Strokes of Multiple Sizes. *SIGGRAPH 98 Conference Proceedings,* p.453-460.

[Hsu94] Hsu, S.C. and I.H.H. Lee (1994). Drawing and Animation Using Skeletal Strokes. *SIGGRAPH 94 Conference Proceedings,* p.109-118.

[Ira00] Irani, P., Ware, C. (2000). Programs Based on Structural Object Perception. *Advanced Visual Interfaces 2000,* p.61-67.

[Int00] Interrante,V. (2000). Harnessing Natural Textures for Multivariate Visualization. *IEEE Computer Graphics and Applications,* Vol.20, Issue 6, p.6-11.

[Jen01] Jensen, H.W. (2001). *Realistic Image Synthesis Using Photon Mapping,* A.K Peters, Ltd, Natick, MA.

[Kai02] Kaiser, P. (2002). Simultaneous Contrast Example. *The Joy of Visual Perception*, a web book retrieved from http://www.yorku.ca/eye/simcont2.htm on [09/07/02]

[Kei02] Keim, D.A. (2002). Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics,* Vol. 8, No. 1, p.1-8.

[Kir99] Kirby, R.M., H. Marmanis, and D.H. Laidlaw (1998). Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting. *Proceedings Visualization '99,* p.333-340.

[Lai98] Laidlaw, D.H., E.T. Ahrens, D. Kremers, M.J. Avalos, R.E. Jacobs, and C. Readhead (1998). Visualizing Diffusion Tensor Images of the Mouse Spinal Cord. *Proceedings Visualization '98,* p.127-134.

[Lai01] Laidlaw, D.H. (2001). Loose, Artistic "Textures" for Visualization. *IEEE Computer Graphics and Applications,* Vol.21, Issue 2, p.6-9.

[Lew84] Lewis, J.P. (1984). Texture Synthesis for Digital Painting. *SIGGRAPH 84 Proceedings, Computer Graphics,* Vol.18,Issue 3, p.245-252.

[Lit97] Litwinowicz, P. (1997). Processing Images and Video for an Impressionist Effect. *SIGGRAPH 97 Conference Proceedings,* p.407-414.

[Lym00] Lyman, P. and H.R. Varian (2000). How Much Information? *Journal of Electronic Publishing,* Vol. 6, Issue 2. retrieved from http://www.sims.berkeley.edu/how-much-info on [6/6/02].

[Mcc87] McCormick, B.H., T.A. DeFanti, and M.D. Brown (1987). Visualization in Scientific Computing *Computer Graphics,* Vol. 21, No.6, p. 1-14.

[Mec96] Mech, R. and P. Prusinkiewicz (1996). Reproduction of the Topiary Garden in Levens, England retrieved from http://pages.cpsc.ucalgary.ca/ jungle/gallery/index.html on [8/31/02].

[Mei96] Meier. B.J. (1996). Painterly Rendering for Animation. *SIGGRAPH 96 Conference Proceedings,* p.477-484.

[Min61] Minard, C.J. (1861). Map of Napoleon's 1812 Campaign on Russia retrieved from http://www.napoleonic-literature.com/1812/1812-t.htm on [6/6/02].

[Mon05] Monet, C. (1905). Monet's Letter to Paul Durand-Ruel. in *Impressionism in Perspective,* B.E. White, Ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1878, p.15.

[Mun45] Munsell, A.H. (1946). *A Color Notation,* Munsell Color Company, Inc.

[Nov95] Novotny, F. (1995). *The Great Impressionists,* Prestel-Verlag, Munich and New York.

[Ost98] Ostermann, R. and B.M. Baltissen (1998). Are the Dimensions Underlying Aesthetics and Affective Judgement the Same? *Empirical Studies of the Arts,* Vol. 16, No.2, p.97-113.

[Per27] Perry, L.C. (1927). Reminiscences of Monet in *Impressionism in Perspective,* B.E. White, Ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1878, p.14-15.

[Pio02] Pioch, N. (2002). Impressionism. *WebMuseum, Paris,* retrieved from http://www.ibiblio.org/wm/paint/glo/impressionism/ on [7/20/02].

[Pru94] Prusinkiewicz, B., M. James, and R. Mech (1994). Synthetic Topiary. *SIGGRAPH 94 Conference Proceedings,* p.351-358.

[Rhe90] Rheingans, P. and B. Tebbs (1990). A Tool for Dynamic Explorations of Color Mappings. *ACM SIGGRAPH Computer Graphics* Vol.24, No.2, p.145-146.

[Rhe01] Rheingans, P. and D. Ebert (2001). Volume Illustration: Nonphotorealistic Rendering of Volume Models. *IEEE Transactions on Visualization and Computer Graphics,* Vol.7, No.3, p.253-264.

[Roo79] Rood, O.N. (1879). *Modern Chromatics, with Applications to Art and Industry,* Appleton, New York, New York.

[Rus80] Russell, J.A. (1980). A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, Vol.39, p.1161-1178.

[Sale02] Salisbury, M.P. (2002). Pen-and-Ink Raccoon. *Images-GRAIL* retrieved from http://grail.cs.washington.edu/cool-images/ on [9/1/02]

[Sali97] Salisbury, M.P., M.T. Wong, J.F. Hughes, and D.H. Salesin (1997). Orientable Textures for Image-Based Pen-and-Ink Illustration. *SIGGRAPH 94 Conference Proceedings,* p.401-406.

[Sou99a] Sousa, M.C. and J.W. Buchanan (1999). Observational Model of Blenders and Erasers in Computer-generated Rendering. *Graphic Interface Proceedings,* p.157-166.

[Sou99b] Sousa, M.C. and J.W. Buchanan (1999). Computer-generated Pencil Rendering of 3D Polygonal Models. *Proceedings Eurographics '99)*, Vol.18, Issue 3, p.195-208.

[Smi98] Smith, P.H. and J. Van Rosendale (1998). Data and Visualization Corridors Report on the 1998 CVD workshop series (sponsored by DOE and NSF). Teck.Rep. CACR-164, Center for Advanced Computing Research, California Institute of Technology, 1998.

[Sno98] Snowden, R.J. (1998). Texture Segregation and Visual Search: A Comparison of the Effects of Random Variations Along Irrelevant Dimensions. *Journal of Experimental Psychology: Human Perception and Performance,* Vol.24, No.5, p.1354-1367.

[Stra86] Strassmann, S. (1986). Hairy Brushes. *SIGGRAPH 86 Conference Proceedings,* Vol. 20, Issue 4, p.225-232.

[Str02] Strothotte, T. and S. Schlechtweg (2002). *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation,* Morgan Kaufmann Publishers, Inc., San Francisco, California.

[Tak99] Takagi, S., M. Nakajima, and I. Fujishiro (1999). Volumetric Modeling of Colored Pencil Drawing. *Proceedings Pacific Graphics,* p.250-258.

[Tre85] Treisman, A. (1985). Preattentive Processing in Vision. *Computer Vision, Graphics and Image Processing,* Vol.31, p.156-177.

[Tri02] Triesch, J., B.T. Sullivan, M.M. Hayhoe,and D.H. Ballard (2002). Saccade Contingent Updating in Virtual Reality. *Proceedings of the Symposium on ETRA 2002,* p.95-102.

[Tuf83] Tufte, E.R. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, Connecticut, 1997.

[Van84] Van Dam, A. (1984). Computer Graphics Comes of Age: An Interview with Andries Van Dam. *Communications of the ACM,* Vol. 27, Issue 7, p.638-648.

[War95] Ware, C. and W. Knight (1995). Using Visual Texture for Information Dispay. *ACM Transactions on Graphics,* Vol.14, No.1, p.3-20.

[War00] Ware, C. (2000). *Information Visualization: Perception for Design,* Morgan Kaufmann Publishers, Inc.

[Wei00] Weigle, C., W.G. Emigh, G. Liu, J.T. Enns, and C.G. Healey (2000). Oriented Texture Slivers: A Technique for Local Value Estimation of Multiple Scalar Fields. *Proceedings Graphics Interface 2000,* p.163-170.

[Whi78] White, B.E. (1978). *Impressionism in Perspective,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, p.24.

[Win94] Winkenbach, G. and D.H. Salesin (1994). Computer-generated Pen-and-ink Illustration. *SIGGRAPH 94 Conference Proceedings,* p.91-100.

[Win96] Winkenbach, G. and D.H. Salesin (1999). Rendering Free-form Surfaces in Pen-and-Ink. *SIGGRAPH 96 Conference Proceedings*, p.469-476.

[Woo00] Wooding, D., M. Mugglestone, K. Purdy, and A. Gale with the National Gallery (2000). *Telling Time Exhibit* and *The Execution of Lady Jane Grey by Paul Delaroche with Eye Traces* retrieved from http://ibs.derby.ac.uk/gallery/ on [09/10/02].

[Wol94] Wolfe, J.M. (1994). Guided Research 2.0 A Revised Model of Visual Search. *Psychonomic Bulletin and Review,* Vol.1, No.2, p.202-238.

[Zei98] Zeitzer, D., A.M. Biszantz, K. Lenk, J.D. Mackkinlay, and R.W. Simons (1998). Visualization (Panel): The Hard Problems. *Siggraph Conference Abstracts and Applications,* p. 177-179.