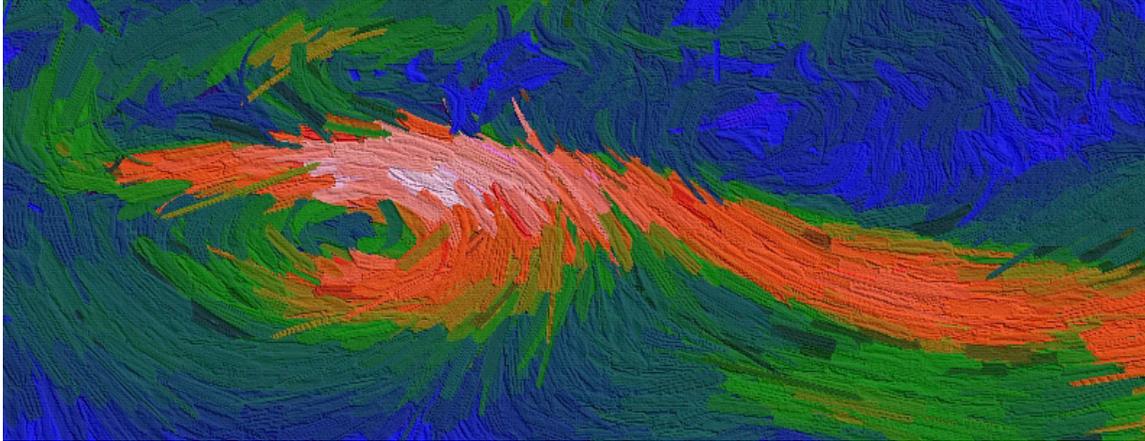


# Engaging Viewers Through Nonphotorealistic Visualizations

Laura G. Tateosian\*  
Department of Computer Science  
North Carolina State University

Christopher G. Healey†  
Department of Computer Science  
North Carolina State University

James T. Enns  
Department of Psychology  
University of British Columbia



**Figure 1:** A visual complexity style visualization of flow patterns in a 2D slice through a simulated supernova collapse, using the mappings: flow orientation  $\rightarrow$  stroke orientation, magnitude  $\rightarrow$  color (dark blue to bright pink for low to high), and pressure  $\rightarrow$  stroke size.

## Abstract

Research in human visual cognition suggests that beautiful images can engage the visual system, encouraging it to linger in certain locations in an image and absorb subtle details. By developing aesthetically pleasing visualizations of data, we aim to engage viewers and promote prolonged inspection, which can lead to new discoveries within the data. We present three new visualization techniques that apply painterly rendering styles to vary interpretational complexity (IC), indication and detail (ID), and visual complexity (VC), image properties that are important to aesthetics. Knowledge of human visual perception and psychophysical models of aesthetics provide the theoretical basis for our designs. Computational geometry and nonphotorealistic algorithms are used to preprocess the data and render the visualizations. We demonstrate the techniques with visualizations of real weather and supernova data.

**Keywords:** nonphotorealistic rendering, visualization, aesthetics, mesh simplification, Voronoi diagrams, NPR applications

## 1 Introduction

Visualizations enable scientists to inspect, interpret, and analyze large multi-dimensional data sets. We believe effective visualizations should both orient and engage (attract *and* hold) viewer attention, directing the viewer’s gaze in response to a visual stimulus,

\*e-mail: lauratateosian@yahoo.com

†e-mail: healey@csc.ncsu.edu

and then encouraging it to linger at a given image location. Research on human visual perception describes how to attract attention, using salient visual features, such as color and texture. Less is known about how to engage viewers with visualizations.

Recently, researchers have applied techniques from nonphotorealistic rendering in computer graphics to enhance and highlight important details in a visualization. We believe that nonphotorealism can also be used to increase the aesthetic merit of a visualization, which in turn may increase its ability to engage a viewer. To this end, we present three new visualization techniques designed with the goal of both orienting and engaging viewers. Orienting is controlled by painterly brush stroke glyphs whose visual properties vary to represent attributes of the underlying data. Human perception studies showed that the information-carrying capabilities of this painterly technique are consistent with more traditional glyph-based visualizations [Healey et al. 2004; Tateosian 2002]. To engage viewers, we incorporate visual qualities important to aesthetics into our visualizations. These visual qualities, interpretational complexity, indication and detail, and visual complexity, were chosen based on psychophysical models of aesthetics. Experiments conducted using these models provide strong evidence that varying image complexity can significantly impact a viewer’s aesthetic judgment.

We apply our visualization techniques to real meteorological and supernova data to explore their capabilities in a real-world setting. Anecdotal feedback from domain experts is strongly positive, supporting the hypothesis that enhancing the artistic merit of a visualization can result in more effective and more productive visual analysis.

The remainder of this paper describes our visualization techniques. Section 2 discusses related work in artistic visualization, nonphotorealistic rendering, and aesthetics. Section 3 introduces the visual qualities that form the basis of our visualizations. Sections 4 and 5 describe the algorithms that were used to preprocess and render the data. Section 6 shows examples of real world applications visualized with our techniques. Finally, Section 7 discusses future work.

## 2 Related Work

We reviewed previous research on aesthetics from the visualization, computer graphics, and human perception communities, then extended and combined results from each domain during the design of our aesthetic visualization algorithms.

### 2.1 Nonphotorealistic Visualization

Scientists in visualization are using ideas from NPR to inspire new visualization techniques. Artistic techniques, such as using abstraction to eliminate unimportant distractions, and sharpening details to draw attention to important areas, can help to convey information more effectively. [Laidlaw et al. 1998; Kirby et al. 1999; Healey et al. 2004; Tateosian 2002] use painting concepts to visualize data. [Laidlaw et al. 1998] visualizes diseased and healthy mouse spinal cords using a tonal underpainting for the anatomy of the brain matter and textured elliptical brush strokes for diffusion tensor data. [Kirby et al. 1999] visualizes air flow past a propeller with an underpainting, ranging from blue to yellow for clockwise to counter clockwise vorticity, and elliptical and arrow-shaped strokes representing additional flow information. The mapping is chosen to display velocity and vorticity prominently, while still showing deformation of the fluid elements. [Healey et al. 2004; Tateosian 2002] create painterly visualizations by mapping data attributes to the visual features of rectangular brush stroke glyphs. In these techniques, brush strokes are “painted” atop an underpainting and stroke features such as shape, color, transparency, orientation, and texture display local data attributes.

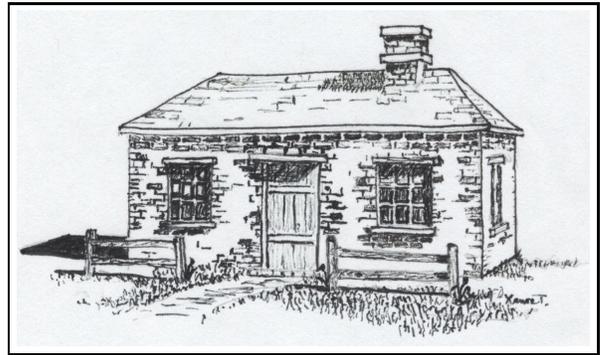
[Grigoryan and Rheingans 2002; Lu et al. 2002] stipple-render volumes. [Grigoryan and Rheingans 2002] uses multi-colored stippling, to show uncertainty on the surface of growing tumors as an intuitively uncertain, fuzzy surface of sparsely placed points. [Lu et al. 2002] uses pen-and-ink stippling, placing and sizing stipples to control both local and global tone. [Rheingans and Ebert 2001; Stoppel et al. 2002] use artistic techniques like silhouetting and outline sketches to enhance volume visualizations.

These visualization methods use artistic techniques to enhance their expressiveness, as do our new algorithms. One important contribution of our visualizations is a basis in psychological literature to vary complexity in ways specifically designed to improve a visualization’s aesthetic merit.

### 2.2 Nonphotorealistic Rendering Techniques

Existing work in NPR provides a rich vocabulary of artistic expression in computer graphics. We use several of the NPR concepts and algorithms described here. Sophisticated pen-and-ink illustration and stippling simulation systems have been developed [Winkenbach and Salesin 1994; Salisbury et al. 1994; Winkenbach and Salesin 1996; Salisbury et al. 1997; Secord 2002; Wilson and Ma 2004]. In our work, we make use of a classic pen-and-ink technique, called, “indication and detail,” that draws more detail in areas of interest and reduces detail in homogeneous regions [Winkenbach and Salesin 1994]. In [Winkenbach and Salesin 1994], users draw line segments on an image where high detail is desired. The level of detail is reduced as the distance from the line segments increases. The hand-drawn house in Fig. 2 shows results this system would produce if detail segments were drawn around the door and front windows.

Painting simulation techniques are of particular interest because of our brush-stroke based visualization style. [Lewis 1984; Strassmann 1986; Pham 1991; Hsu and Lee 1994] present approaches for modeling sophisticated individual strokes. Several systems use



**Figure 2:** A hand-drawn recreation of an image from Winkenbach and Salesin’s paper on computer-generated pen-and-ink drawings demonstrating the use of “indication and detail” [Winkenbach and Salesin 1994]. This technique avoids monotonous repetition by presenting a small amount of detail to indicate patterns that continue in the surrounding regions.

lists of brush strokes to create Impressionist style imagery [Haeberli 1990; Meier 1996; Litwinowicz 1997; Hertzmann 1998; Shiraishi and Yamaguchi 2000; Hertzmann 2002]. [Meier 1996] creates animated paintings and [Litwinowicz 1997] generates painterly videos. [Hertzmann 1998; Shiraishi and Yamaguchi 2000] simulate paintings of source photographs. [Hertzmann 1998] layers curved B-splined strokes, increasing detail on each layer. [Shiraishi and Yamaguchi 2000] use image moment functions to size and orient rectangular strokes. [Hertzmann 2002] extends [Hertzmann 1998] with a fast paint texture method to simulate the thickness of paint using a height field and lighting, a technique we implement to create textured visualizations.

### 2.3 Characterization of Beauty

To choose visual properties to vary in our visualizations, we reviewed models of aesthetics that identify parameters that are believed to affect aesthetic judgment. In an early attempt to model beauty with mathematical equations, Birkhoff calculated the aesthetic measure,  $M$ , of objects, such as polygons, tiles, and vases as a ratio of order to complexity [Birkhoff 1932]. Birkhoff theorized that heightened complexity and disorder induce feelings of discomfort and so aesthetic measure decreases as the ratio of order to complexity decreases. Observer rankings of Birkhoff’s polygons in a study conducted by Davis, suggest that aesthetic value is curvilinearly related to  $M$ , peaking when  $M$  is moderate [Davis 1936].

The same pattern is seen in Berlyne’s model of aesthetic pleasure as a function of arousal, which plots as an inverted U [Berlyne 1971]. Viewing beautiful images generally evokes a feeling of pleasure. The image provides an activating stimulus and the arousal induces pleasure. Thus, artistic value is identified with pleasure. In this model, as arousal increases, pleasure increases until it peaks and decreases toward indifference (zero pleasure) and displeasure (negative pleasure values). Structural properties, such as complexity, novelty, conflict, expectations, ambiguity, and instability increase arousal; whereas, familiarity, dominance, grouping and pattern, and expectations moderate arousal.

Arousal and pleasure axes also comprise Barrett and Russell’s affective model, a tool for specifying a wide range of emotional states. This does not imply that aesthetic judgment is solely an emotional process. Factor analysis on how paintings are judged revealed that a cognitive factor, made up of complexity, meaningfulness, interestingness, pleasantness, and familiarity, explained a

majority (51.4%) of the variance in judgment of paintings and an emotional factor, made up of warmth, emotionality, arousal, and dominance, explained 13.7% of the variance [Baltissen and Ostermann 1998]. The converse was true for emotional pictures (48.1% of the variance explained by the emotional scale and 14% explained by the cognitive scale). Viewing paintings may be largely a cognitive process, because it requires the viewer to identify the meaning, and emotions may arise merely as an after-effect associated with the cognitive process.

### 3 Designing Aesthetic Visualizations

Birkhoff, Berlyne, and Barrett and Russell reveal that complexity and closely related properties such as dominance play an important role in aesthetic judgment. Our three nonphotorealistic visualization techniques are designed to vary these properties.

**Interpretational Complexity (IC).** IC uses a layered approach to introduce complexity. Paintings are often developed in layers with an undercoating broadly defining the shapes in the image and details refined with subsequent layers. This aspect of paintings adds interpretational complexity, because the information provided by additional layers requires interpretation. Unlike the layering in [Tateosian 2002; Hertzmann 1998; Shiraishi and Yamaguchi 2000; Hertzmann 2002], we wanted our underpainting to contrast in display style and level of detail from the top layer, because extracting contrast reinforces attention [Ramachandran 2000]. Hence the underpainting, a colored canvas with sparse faint strokes, provides a lower level of detail than the highly detailed top level painted with wet strokes. The data to be visualized is filtered to detect regions of rapid change, as described in Section 4. Then distinct strokes are laid on a second layer to provide detail in rapid change areas, but not elsewhere so that the underpainting is not fully covered.

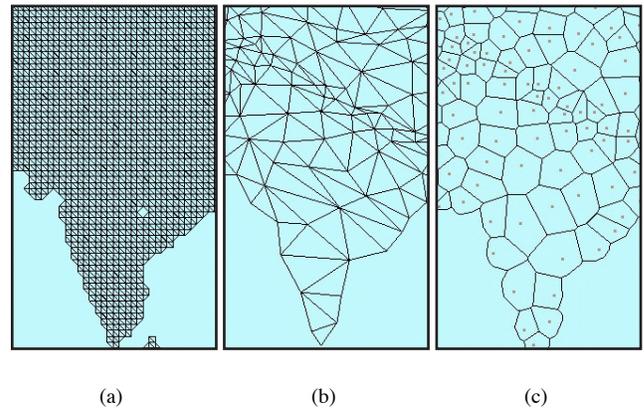
**Indication and Detail (ID).** ID provides focal points with high detail areas and omits monotonous details in homogeneous regions (Fig. 2). Tracking the eye movements of people viewing paintings has shown that viewers fixate on areas of high detail, such as faces [Wooding 2002; Holman et al. 2004]. The abstraction of detail in some areas allows other areas to be visually dominant, resolving the conflict created by competing stimuli. Short-term familiarity develops when a visual element appears, disappears, and reappears in an image after a short hiatus. Overly repetitive visual displays may be too simplistic to be engaging. But *variation*, repeating patterns that are alike in one respect and varying in another, is a powerful aesthetic device [Berlyne 1971]. We can reuse the filtering technique we developed for IC to preprocess the data. Then highly variable regions are drawn with high detail and clearly distinguishable brush strokes, and homogeneous areas are drawn with low detail and subtle strokes. Transitional strokes, something like the partially drawn bricks in Fig. 2, are drawn on the borders between low and high detail regions. In ID, stroke textures are outlined with a marker to create a congruous style, so that the detailed areas gradually evolve into abstracted areas, unlike IC where the layers are designed to contrast in style.

**Visual Complexity (VC).** In two patterns with the same number of visual elements, the one with the most similarity among its elements will be considered less visually complex [Berlyne 1971]. Inspecting master Impressionist artist paintings reveals a great deal of variation amongst the individual brush strokes. There are often differences in the color and thickness of paint even within a single stroke. These visual properties contribute to the complexity of the painting, but in a way that differs from interpretational complexity. VC involves local visual variations, where IC creates global trends. Psychologists consistently cite complexity as one of the components affecting the judgment of aesthetics. Thus, we decided

to study visual complexity, as well as interpretational complexity.

To inject visual complexity, we introduce new stroke properties that correspond to observable characteristics of brush strokes in master Impressionist paintings. To identify these properties, we consulted an artist who has a background in Fine Arts and is a practicing oil and watercolor painter. We told her we were seeking salient and aesthetically pleasing ways to vary our basic brush stroke style and asked her to look at several Van Gogh reproductions. We mapped her observations of the kinds of variations she saw in these paintings to six properties: varied stroke contour, variegation (paint color varying within one stroke), curved strokes, embossed strokes (paint thickness varying within one stroke, sometimes dramatically), varied aspect ratio (some strokes are much longer or fatter than others), and contrast strokes (sets of dark and light strokes moving in different directions, juxtaposed in places).

### 4 Data Preprocessing



**Figure 3:** Processing Indian weather data: a) Triangulate the data element locations. b) Simplify the mesh based on data attribute values. c) Find the corresponding Voronoi diagram.

The IC and ID techniques first identify areas of high spatial frequency within the data to be visualized. To do this we apply mesh simplification followed by spatial analysis with Voronoi regions.

Feature preserving mesh simplification reduces a mesh by removing polygon vertices in ways that minimize the surface error. For example, a simplified terrain map mesh can describe a plateau with just a few large polygons, whereas, many more polygons remain for craggy mountains. Mesh simplification algorithms that handle surface properties, such as RGB triples, surface normals, and texture coordinates, can be used to process a multidimensional data set, with each data element’s attribute values representing surface properties at the corresponding vertex [Walter and Healey 2001]. After simplification, the plateaus of our example, are characterized by a sparse distribution of data elements, while in regions of rapid attribute value change (craggy mountains) the data elements remain dense. We used a popular implementation called qslim, which applies iterative vertex contraction with a quadric error metric to simplify meshes with associated surface properties [Garland and Heckbert 1997; Garland and Heckbert 1998; Garland 1999].

To use qslim, we triangulated our data sets, set the surface properties to data attributes, and chose a desired face count. Fig. 3a shows triangulated data elements from an Indian weather data set. The output face count is reduced to 3% of the original face count (Fig. 3b). The data attributes, *radiation*, *wet day frequency*, and *diurnal temperature range* were set as surface properties, and used to determine the quadric error.

To measure the denseness of the mesh vertices, we use a nearest point Voronoi diagram, which is a partitioning of space around a set,  $S$ , of  $n$  sites,  $S = p_1, \dots, p_n$ , into  $n$  regions, such that each point inside the region containing  $p_i$  is closer to  $p_i$  than any of the other  $n - 1$  sites. Since the size of the Voronoi region decreases as density increases, the Voronoi diagram of our reduced mesh identifies regions of rapid change. We used an available implementation, qhull, to find our Voronoi diagrams with the results from qslim acting as input sites to qhull [Barber et al. 1993; Barber et al. 1996; Brown 1979]. Using our simplified Indian weather example in Fig. 3b as input, the resulting Voronoi diagram is shown in Fig. 3c. The region sizes are correlated with the density of the input mesh. For example, because the input attributes, *radiation*, *wet day frequency*, and *diurnal temperature range*, vary rapidly in the Himalayas, the Voronoi regions are much smaller in these mountainous areas.

## 5 Visualization Algorithms

Our three visualization techniques all create painted regions of strokes that vary in color, size, and orientation based on underlying data values. IC, ID, and VC share a common painting algorithm to lay strokes in a random manner while controlling coverage. As region  $S_k$  is painted, coverage is controlled by tracking the percentage,  $c$ , of coverage of the total area of  $S_k$ . A stroke is centered at a randomly chosen unpainted position within  $S_k$ . It is transformed and scan converted, and then retained or rejected based on whether there is too much overlap (when too much of a stroke falls outside  $S_k$  or overlaps previously laid strokes). Stroke properties such as size, orientation, and color, are determined by a data sample at the center of the stroke. Each sample is a vector of data attribute values at that position in the underlying data set, which are mapped to a vector of visual features. Stroke placement continues until the region’s desired coverage,  $c$ , has been met. The output is a list of the accepted strokes and their properties. Each stroke in the list is drawn as a textured quad. Stroke textures were mostly generated from scanned hand-painted strokes or strokes drawn with marker. The algorithm for creating a stroke list with coverage  $V_k$  of region  $S_k$  is:

```

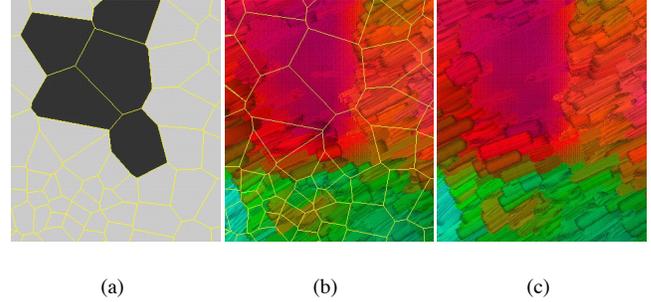
while  $c < V_k$  do
  Randomly pick an unpainted position  $p$  within  $S_k$ 
  Identify data element  $e_i$  associated with  $p$ 
  Set size and orientation of a new stroke,  $s$ , based on  $e_i$ 
  Center  $s$  at  $p$  and scan convert
  Compute amount of  $s$  outside of  $S_k$ , outside
  Compute amount of  $s$  overlapping existing strokes, overlap
  if outside or overlap are too large then
    Shrink  $s$  until it fits or it cannot be shrunk
  end if
  if  $s$  fits then
    Add to stroke list and update  $c$ 
  end if
end while

```

### 5.1 The Interpretational Complexity Algorithm

For IC, we wanted to build two layers, each with a distinct style. Master artists often use an underpainting to broadly define forms. The underpainting, varying in color, sometimes peeks through to higher detail layers. The individual strokes in the underpainting are difficult to discern. These patches of underpainting appear where there is little change or detail in the image.

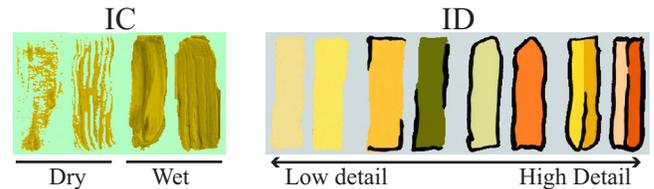
The Voronoi regions determine where the underpainting should show by partitioning the data into two sets:  $S_0$ , representing low spatial frequency changes in the data (i.e. large Voronoi regions), and  $S_1$ , representing high spatial frequency changes (i.e. small



**Figure 4:** The IC algorithm: a) Partition Voronoi regions into sets  $S_0$  (dark) and  $S_1$  (light) based on size. b) Paint layer  $S_0$ , then  $S_1$  (Voronoi regions overlaid). c) Render the results.

Voronoi regions). For each region, if its area exceeds a *threshold* percentage of the largest Voronoi region, it is placed in  $S_0$ , otherwise it is placed in  $S_1$  (Fig. 4a).

The Voronoi regions in  $S_0$  are tiled with colored canvas texture to establish the broad color changes of the underpainting. The tiles are smoothly shaded, with the nearest data element’s attribute values determining the color of each tile corner. A sparse collection of strokes are placed on the tiles by centering a stroke on each Voronoi vertex, and in the center of each Voronoi region (Figs. 4b and c). The strokes are textured with a dry brush appearance (Fig. 5), and seem faint since some of the underlying canvas shows through.



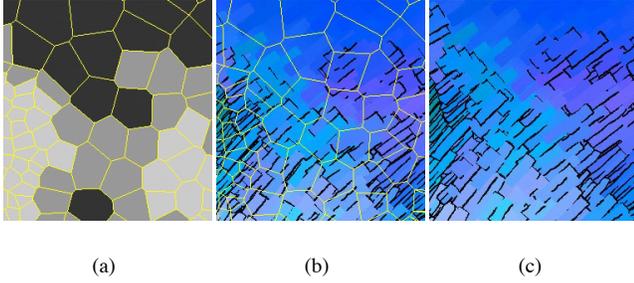
**Figure 5:** Sample stroke textures: left) dry and wet brush strokes for the IC visualizations. right) strokes with increasing detail for the ID visualizations.

Finally, Voronoi regions in  $S_1$  are painted with our painting algorithm set to produce full coverage. Each stroke is textured in a wet brush fashion (Fig. 5). This produces the necessary contrast with the dry brush strokes in regions from  $S_0$ . Fig. 4 demonstrates the IC process.

### 5.2 The Indication and Detail Algorithm

To create our stylized ID visualizations, we use the Voronoi regions to partition the data into three sets  $S_0$ ,  $S_1$ , and  $S_2$ , then paint with textured strokes like those in Fig. 5. We begin by identifying all Voronoi regions whose area  $A \geq \text{threshold}$ , and placing them in  $S_0$  (large Voronoi regions). For the remaining regions if the region has at least one neighbor in  $S_0$ , we place it in  $S_1$  (small Voronoi regions bordering large ones), otherwise we place it in  $S_2$  (small Voronoi regions).

Voronoi regions in  $S_0$  are drawn as smooth shaded polygons, with vertex colors based on the data element at their corresponding spatial position. The regions are then painted with one stroke at their center and one stroke at each vertex, rendered with low detail texture maps that have no outline (Fig. 5).



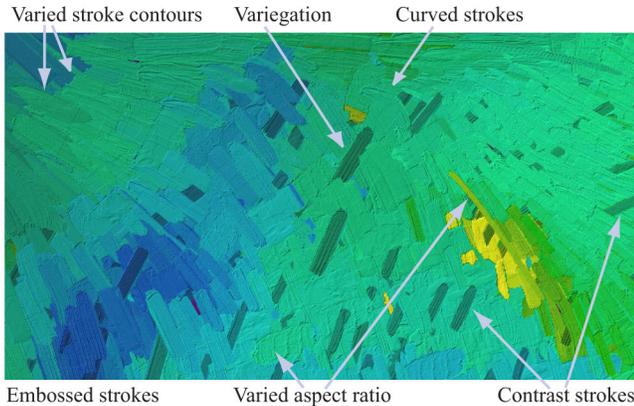
**Figure 6:** Generating increase in detail for the ID algorithm: a) Partition Voronoi regions into sets  $S_0$ ,  $S_1$ , and  $S_2$  (dark to light). b) Paint each region (Voronoi regions overlaid). c) Render the results

Voronoi regions in  $S_1$  and  $S_2$  are painted with our painting algorithm. Coverage for regions in  $S_1$  is set to  $V_1$ ,  $0 \ll V_1 < 1$ . The strokes are texture mapped to have partial outlines (Fig. 5).

Regions in  $S_2$  are painted with full coverage. Strokes in these regions are further partitioned based on local variations of the attribute mapped to color. First, the stroke is rotated based on the attribute value mapped to orientation. Next, the data is sampled  $\frac{1}{4}w$  from the stroke's center (where  $w$  represents stroke width) to locate flank colors  $(R_1, G_1, B_1)$  and  $(R_2, G_2, B_2)$ . If the difference between the flank colors  $\sqrt{\Delta R^2 + \Delta B^2 + \Delta G^2} \leq \delta$  for a small positive constant  $\delta$ , the stroke is textured with a complete outline but no center vein (Fig. 5). Otherwise, the stroke is rendered as two quads colored  $(R_1, G_1, B_1)$  and  $(R_2, G_2, B_2)$ , and textured with a complete outline and a center vein.

With this design, the stroke count increases from  $S_0$  to  $S_1$  and again from  $S_1$  to  $S_2$ . The addition of stroke outlines reinforces this gradual increase in level of detail. Fig. 6 shows an example of the ID algorithm.

### 5.3 The Visual Complexity Algorithm



**Figure 7:** Stroke properties in our VC algorithm

VC varies properties of the individual strokes. The VC algorithm places every data element in a single region  $S_0$ , then applies our painting algorithm with full coverage to create a list of strokes, varying the stroke properties as the list is created.

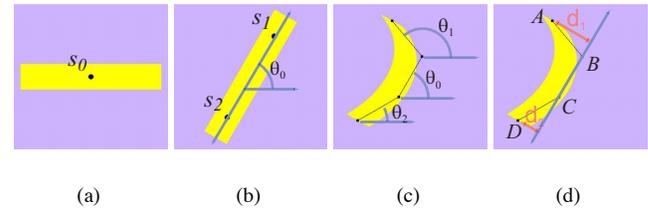
**Varied Stroke Contours.** We scanned hand-painted strokes to construct our texture maps. This introduces more variability to the

footprint of the stroke, compared to rectangular, computer generated strokes. For example, some of the strokes have a rounded end, while others are angular (Fig. 7).

**Variegation.** Strokes are drawn as three quadrilaterals  $q_1$ ,  $q$ , and  $q_2$ , of equal length. The data is sampled at the center of the stroke to choose a primary stroke color  $C = (R, G, B)$ , and the length  $l$  and width  $w$  of the stroke.  $C$  is varied randomly to produce  $C_1 = (R_1, G_1, B_1)$  and  $C_2 = (R_2, G_2, B_2)$  in a small interval around  $C$ 's RGB values.  $w$  is also varied to generate widths  $w_1$  and  $w_2$  such that  $0 \leq w_1, w_2 < \frac{1}{8}w$ .

$q$  is then drawn centered on the sample point with length  $l$ , width  $w - (w_1 + w_2)$ , and color  $C$ .  $q_1$  is drawn next to  $q$  with length  $l$ , width  $w_1$ , and color  $C_1$ , and  $q_2$  is drawn opposite of  $q_1$  with length  $l$ , width  $w_2$ , and color  $C_2$ . This simulates different paint colors being pushed to the margins of a stroke, as often occurs when the brush has residual paint from a previous dip. The randomly colored flanks provides important color variations.

**Curved Strokes.** We render strokes as Bézier curves with four control points,  $A$ ,  $B$ ,  $C$ , and  $D$ . The configuration of a curve is controlled by the data attribute mapped to orientation. We query the attribute values at three sample points,  $s_0$ ,  $s_1$ , and  $s_2$ , then compute the associated orientations to position the stroke's control points.



**Figure 8:** Placing control points,  $A$ ,  $B$ ,  $C$ , and  $D$ : a) Get data sample,  $s_0$ . b) Rotate by  $\theta_0$  and get samples,  $s_1$  and  $s_2$ . c) Place control points, so that  $\overline{BC}$  has orientation  $\theta_0$ ,  $\overline{AB}$  has orientation  $\theta_1$ , and  $\overline{CD}$  has orientation  $\theta_2$ . d) If  $\text{MAX}\{d_1, d_2\}$  is large, draw stroke as a curve.

$s_0$  is taken at the center of the stroke to define its orientation  $\theta_0$  (Figs. 8a).  $s_0$  also determines the stroke's color and size. This is consistent with how rectangular strokes are colored and sized. Since both the curve and  $s_0$  will lie in the convex hull of the control points, the curve will either pass through  $s_0$  or close to it. After the stroke is rotated by  $\theta_0$ , samples  $s_1$  and  $s_2$  are taken  $\frac{1}{4}l$  from the ends of the stroke, yielding orientations  $\theta_1$  and  $\theta_2$ , respectively (Fig. 8b). The control points are then positioned such that that line segment  $\overline{BC}$  has orientation  $\theta_0$ , line segment  $\overline{AB}$  has orientation  $\theta_1$ , and line segment  $\overline{CD}$  has orientation  $\theta_2$  (Fig. 8c). Since Bézier curves are approximating, not interpolating, the curve may not pass through  $B$  and  $C$ , but it will be pulled in their direction. Curved strokes are only drawn where the underlying data attribute mapped to orientation varies rapidly (Fig. 8d).

**Embossed Strokes.** We simulate variations in the thickness of the paint on each stroke using Hertzmann's fast paint texture technique [2002]. The algorithm assigns a height map to each stroke, then generates an overall height field of the image as it is painted. When the scene is lit, shadows generated by the height field simulate the undulations of paint on a canvas. This procedure is meant to avoid rendering expensive three-dimensional mesh models of the strokes, while producing many of the same visual results.

Hertzmann's height field takes into account both the texture of each individual stroke produced by the bristles as they spread the paint and the overlap of paint as strokes are laid atop each other. We

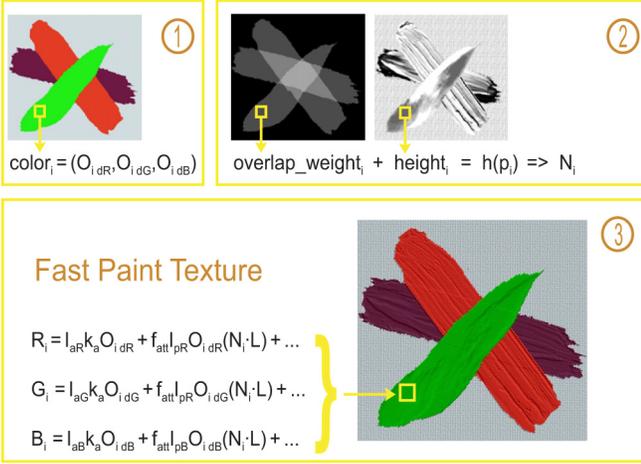


Figure 9: Implementation of Hertzmann’s fast paint texture.

use texture maps paired with opacity maps to calculate these values. We refer to the texture maps as “height maps” here, since the greyscale values are used as height values (dark to light for low to high). The contrast on wet brush stroke texture maps like those in Fig. 5 was increased, to provide more dramatic height differences within the height maps. We create an opacity map, which encodes the stroke’s footprint, to correspond to each height map, by coloring all the points within the stroke white and setting the remaining background transparent. The opacity maps are used to compute a stroke overlap count. The final height field value for a given image position  $(x, y)$  is the sum of the stroke’s height map value at  $(x, y)$  plus a weight representing the number of strokes overlapping at  $(x, y)$ .

To implement fast paint texturing, we draw our stroke list three times. Strokes are rendered differently each time, to collect different information about the scene. The first pass records the colors. The next two passes record the overlap count and the stroke height map information that is used to create the height field. The scene is rendered pixel by pixel, using the color buffer, the height field, and a Phong shading model to calculate each pixel’s color:

1. *Color Pass*: Draw each stroke colored and textured with an opacity map and save the RGB components in the *color* buffer.
2. *Overlap Pass*: Draw each stroke white and textured with the same opacity maps as those used in color pass. The strokes are alpha blended with transparency  $0 < \alpha \ll 1$  to count the number of overlapping strokes at each pixel. Given source and destination blending factors,  $\alpha$  and  $1 - \alpha$ , and source and destination colors,  $(R_s, G_s, B_s, A_s)$  and  $(R_d, G_d, B_d, A_d)$ , respectively, we obtain:

$$R_i = R_s \alpha + R_d (1 - \alpha) \quad (1)$$

$$G_i = G_s \alpha + G_d (1 - \alpha) \quad (2)$$

$$B_i = B_s \alpha + B_d (1 - \alpha) \quad (3)$$

$$A_i = \alpha^2 + A_{id} (1 - \alpha) \quad (4)$$

Starting with a black background sets the initial  $(R_d, G_d, B_d, A_d) = (0, 0, 0, 1)$  at every pixel  $p_i$ . Since each stroke’s color is set to  $(R_s, G_s, B_s, A_s) = (1, 1, 1, \alpha)$ , we can compactly define  $R_{i,k}$ , the red component of pixel  $p_i$

where  $k$  strokes have been drawn, as:

$$R_{i,k} = \begin{cases} 0, & \text{if } k = 0 \\ \alpha + R_{i,k-1}(1 - \alpha), & \text{otherwise} \end{cases} \quad (5)$$

In this way, the final value of any of the color components of a pixel enables us to count the number of strokes that overlap there. We choose to save the red component of the color buffer in the *overlap\_weight* buffer.

3. *Height Pass*: Draw each stroke white and textured with the height map paired with the opacity map used in color and opacity passes. Each pixel  $p_i$  in the resulting color buffer contains the grey value (i.e. the height value) of the last stroke drawn at  $p_i$ . We save the red component of the color buffer in the *height* buffer.
4. Compute the height value at each pixel  $p_i$  as:

$$h(p_i) = \text{overlap\_weight}_i + \text{height}_i \quad (6)$$

Once heights are established, we calculate per-pixel normals using the directional derivatives of the height field.

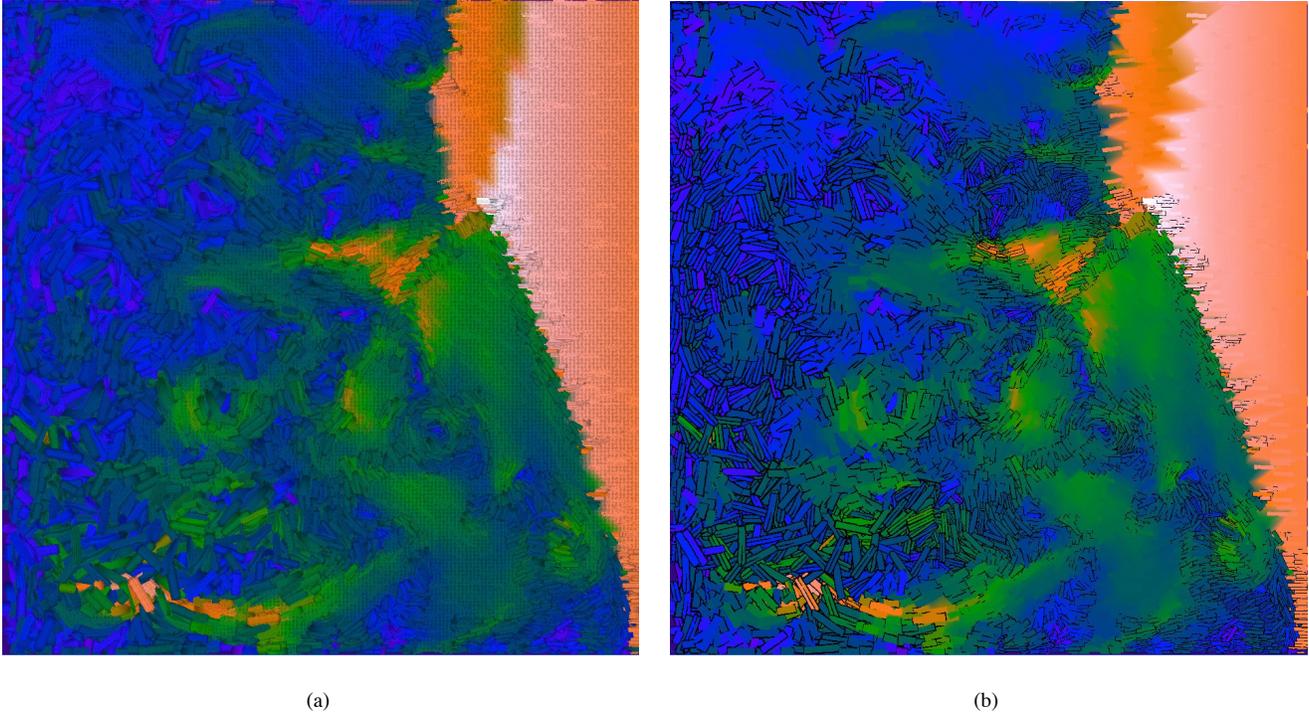
5. For each pixel  $p_i$  we combine the stroke’s diffuse color components in *color<sub>i</sub>* and the surface normal using the Phong model to obtain an RGB color value at  $p_i$ . These value are displayed in the final rendered image (Fig. 9).

**Varied Stroke Aspect Ratio.** This visual feature varies the ratio of stroke length  $l$  to stroke width  $w$ . For a majority of the strokes,  $l$  and  $w$  are selected based on the data attribute  $A_i$  mapped to size.  $A_i$ ’s value determines  $w$ , with  $l = cw$  for some positive constant  $c$ . For a minority of the strokes, however, we let a different attribute  $A_j$ ,  $j \neq i$  control the aspect ratio. If  $A_j$ ’s value is below a preset threshold  $t \gg 0$ ,  $w$  is set based on  $A_i$ ’s value, and  $l = cw$  with  $c = 4$ . If instead  $a_j < t + \frac{\text{MAX}(A_j) - t}{2}$ ,  $w$  is set to a constant value and  $l = c_1 w$  with  $c_1 < 1$ , producing strokes that are wider than they are long. The paint texture reinforces the stroke direction, so that it is not misinterpreted as a stroke with  $l = \frac{1}{c_1} w$ . If neither condition holds,  $w$  is set to a small constant value and  $l = c_2 w$  with  $c_2 \gg c$  to yield strokes that are much longer than they are wide.

**Contrast Strokes.** In Van Gogh wheat field paintings, burgundy strokes contrast in color and direction with a field of golden strokes. We also used the color and orientation of a minority of the strokes to contrast with the majority. A contrast property, either luminance, luminance and orientation, or color, is chosen. If attribute  $A_i$  is mapped to color and  $A_j$  is mapped to orientation,  $A_k$  is mapped to contrast ( $k \neq i \neq j$ ). Stroke color  $C$  and orientation  $\theta$  are selected based on  $A_i$  and  $A_j$ . 10% of the strokes with an  $A_k$  value above a preset threshold are selected to act as contrast strokes. If the contrast property is luminance, the stroke’s luminance is shifted by a constant. If the contrast property is luminance and orientation, the luminance is shifted and the stroke’s orientation is reset to  $-\theta$ . If the contrast property is color, color values in a small interval around  $C$  are assigned to the stroke.

## 6 Practical Applications

We used two real-world data sets to test the practical capabilities of our visualization algorithms: flow from a simulated supernova collapse, and historical weather conditions calculated for positive elevations throughout the world.



**Figure 10:** Visualization of supernova data set, with mapping  $\Delta x$  and  $\Delta y \rightarrow$  orientation, magnitude  $\rightarrow$  color, density  $\rightarrow$  size. a) IC visualization. b) ID visualization.

## 6.1 Supernova Application

A supernova is a massive explosion that occurs at the end of a star's lifetime. Researchers in the Physics Department at NC State University are studying these phenomena. Their current interests include simulating how a supernova collapses. The resulting supernova data sets have time steps and describe a three-dimensional volume of data. Astrophysicists often look at two-dimensional slices of the data at a particular time step to analyze a snapshot of the flow. The slices we are visualizing contain a  $500 \times 500$  regular grid of sample points composed of the attributes  $\Delta x$  and  $\Delta y$  (direction of the velocity vector), *magnitude* (of the velocity vector), *pressure*, and *density*.

We created visualizations of this data using each of our new techniques. To preprocess the data, we first created a mesh by triangulating the sample points and storing *magnitude*, *pressure*, and *density* at each vertex so the simplified mesh would be dense in areas where one or more of the attribute values change rapidly, and sparse in areas where they are relatively constant. Next we calculated the Voronoi diagram of the simplified mesh. This Voronoi diagram was used to partition the data for the IC and ID visualizations.

The IC visualization in Fig. 10a maps *magnitude* to a perceptually balanced color ramp (dark blue to bright pink for low to high).  $\Delta x$  and  $\Delta y$  determine orientation. *Density* is mapped to stroke size (small to large for low to high). *pressure* is not visualized, however, we know that *pressure*, *magnitude*, and *density* are relatively constant where the underpainting is visible. The bright pink area on the right represents a shock wave. The strokes form circular patterns where there are vortices. In Fig. 10b, the data is visualized with the ID technique and the same data-feature mapping. The low and high detail region IC textures in Fig. 10a contrast each other as sharp spatial boundaries. In ID visualization, stroke styles gradually transform across levels of detail.

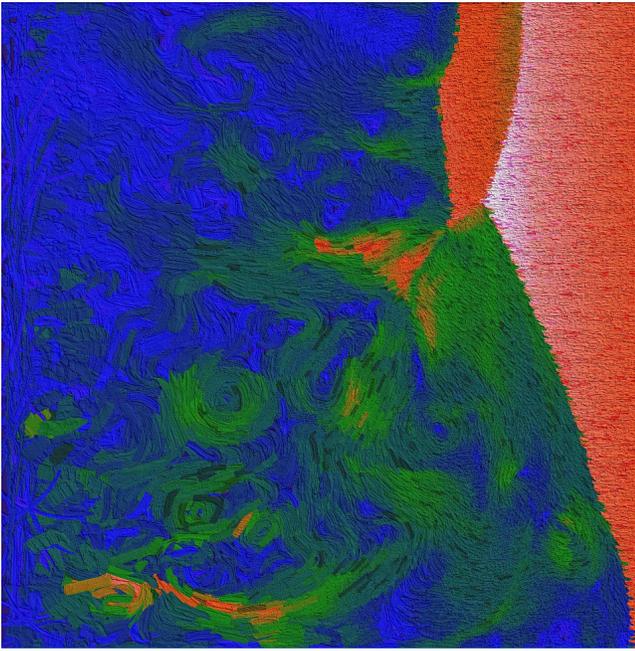
To create a VC visualization, we again mapped *magnitude* to color,  $\Delta x$  and  $\Delta y$  to orientation, and *density* to stroke size. Contrast is used to reinforce magnitude (10% of the strokes have reduced luminance where magnitude  $> 20\%$ ). Pressure is mapped to aspect ratio ( $70\% \leq \text{pressure} < 85\% \rightarrow$  wide stroke and  $\text{pressure} \geq 85\% \rightarrow$  long stroke). The resulting visualization is shown in Fig. 11. Wide and long strokes appear on the left where pressure is the highest. The shock wave has contrast strokes, reinforcing the high magnitudes that occur in this region.

Dr. John Blondin, an astrophysicist from the physics department, provided the data and gave us anecdotal feedback on our visualizations. He found the use of glyphs to show multiple attributes in all three techniques assists in the investigation of attribute interactions. He thought visual complexity was the most appealing technique for this application, since the continuity of the glyphs reflects the continuous flow. For example, on the right side of Fig. 11 where the shock wave impact occurs, he could see how the particles were coming together and causing vortices. The sense of flow was reduced in the corresponding regions in Figs. 10a and b. His preference for VC is interesting because he also claimed to find this visualization the most aesthetically pleasing.

## 6.2 Weather Application

Weather data is a second convenient data source, since weather data is plentiful and weather attributes, such as temperature, precipitation, and wind speed, are commonly referenced in day-to-day settings. The data used here, collected by the International Panel on Climate Change between the years 1961 and 1990, consists of 30 year mean monthly weather conditions sampled on a  $\frac{1}{2}^\circ$  latitude by  $\frac{1}{2}^\circ$  longitude regular grid of positive elevations throughout the world.

Figs. 13a and b and 14 show visualizations of this weather data



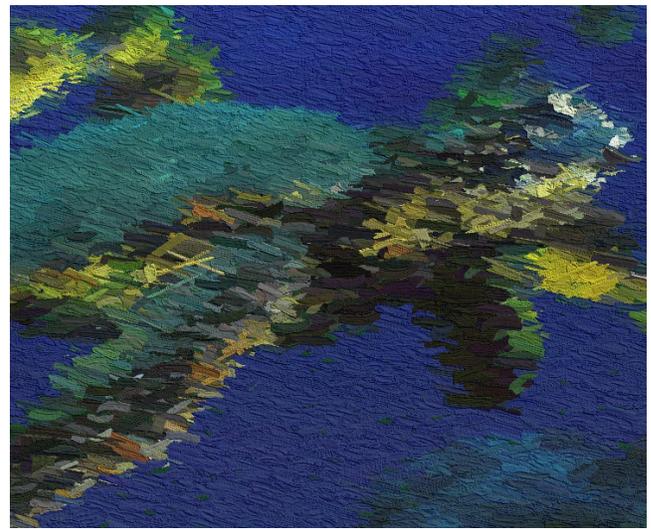
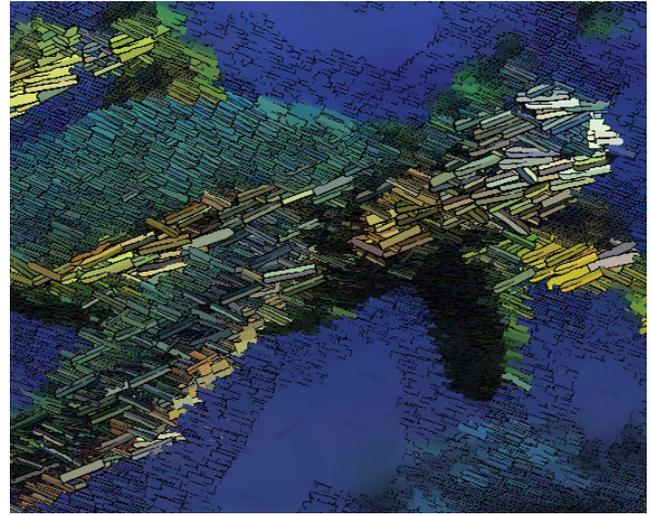
**Figure 11:** VC visualization of supernova data set, with mapping  $\Delta x$  and  $\Delta y \rightarrow$  orientation, magnitude  $\rightarrow$  color, density  $\rightarrow$  size, pressure  $\rightarrow$  aspect ratio; magnitude also controls contrast strokes.

using our three techniques. In Fig. 13a, average African weather conditions in the month of May are visualized. Here *cloud cover*, *mean temperature*, *pressure*, and *diurnal temperature range* are visualized with hue, luminance, size, and orientation. The underlying Voronoi diagram was created with our usual mesh reduction technique, with *temperature*, *wind speed*, and *pressure* stored at each vertex. The luminance is generally higher in northern Africa, indicating the higher temperature there. The canvas shows in interior regions that have relatively constant *temperature*, *wind speed*, and *pressure*.

Mean January South American weather patterns are visualized with our indication and detail technique in Fig. 13b. Here *wind speed*, *minimum temperature*, *wet day frequency*, and *precipitation* are visualized with hue, luminance, size, and orientation. Mesh simplification was performed with *wet day frequency*, *pressure*, and *diurnal temperature range*. The low detail region in the interior of Brazil shows highlight these attributes have low variability. In the Andes, we can see hue variations and a dark luminance, indicating variable wind speeds and low minimum temperatures relative to the rest of the continent.

In Fig. 14, mean weather conditions for January in southwest Canada are visualized with our visual complexity technique. *Mean temperature*, *pressure*, *wet day frequency*, *wind speed*, and *precipitation* are visualized with color, size, orientation, contrast, and aspect ratio, respectively. The temperature is increasing from north to south with high wind speeds in south central regions where contrast strokes are drawn with reduced luminance and opposing orientation. Wide and long strokes appear in the southwest where precipitation is high.

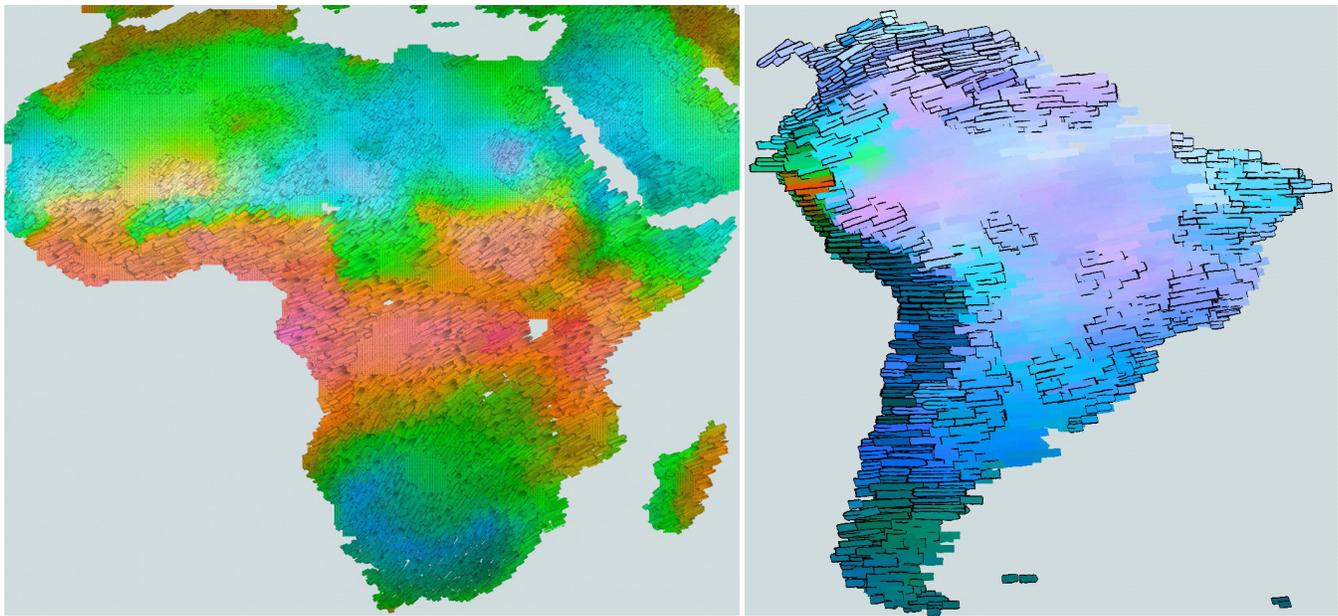
Our techniques are not limited to scientific data visualizations. They can also be applied to other types of input, for example, photographs interpreted as a regular grid of RGB data samples. Fig. 12 shows two examples of an input photograph of a sea turtle, rendered using the ID and VC techniques.



**Figure 12:** A photograph of a sea turtle rendered using: Top the ID algorithm. Bottom the VC algorithm.

## 7 Conclusions

This paper presents new artistic styles for visualizing large scientific data. The styles are designed to both orient viewers with the deliberate use of salient visual features and to engage viewers with aesthetic appeal. We designed these styles by studying psychophysical models of aesthetics and combining them with expressive NPR techniques. We used the new styles to visualize real world data and received encouraging feedback from our domain experts. In fact, our collaborators have started to use our visualizations in their presentations and publications. We are now conducting and analyzing controlled experiments to measure how viewers rate aesthetic, emotional, and visual composition properties of our NPR visualizations relative to real paintings by Abstractionist and Impressionist Masters. Preliminary results have been promising, particularly for IC and VC. This suggests that our techniques can positively impact aesthetic judgment. Given the apparent increase in aesthetic merit produced by our visualizations, the next step in our research will study whether we are increasing engagement, again through the use of controlled experiments.



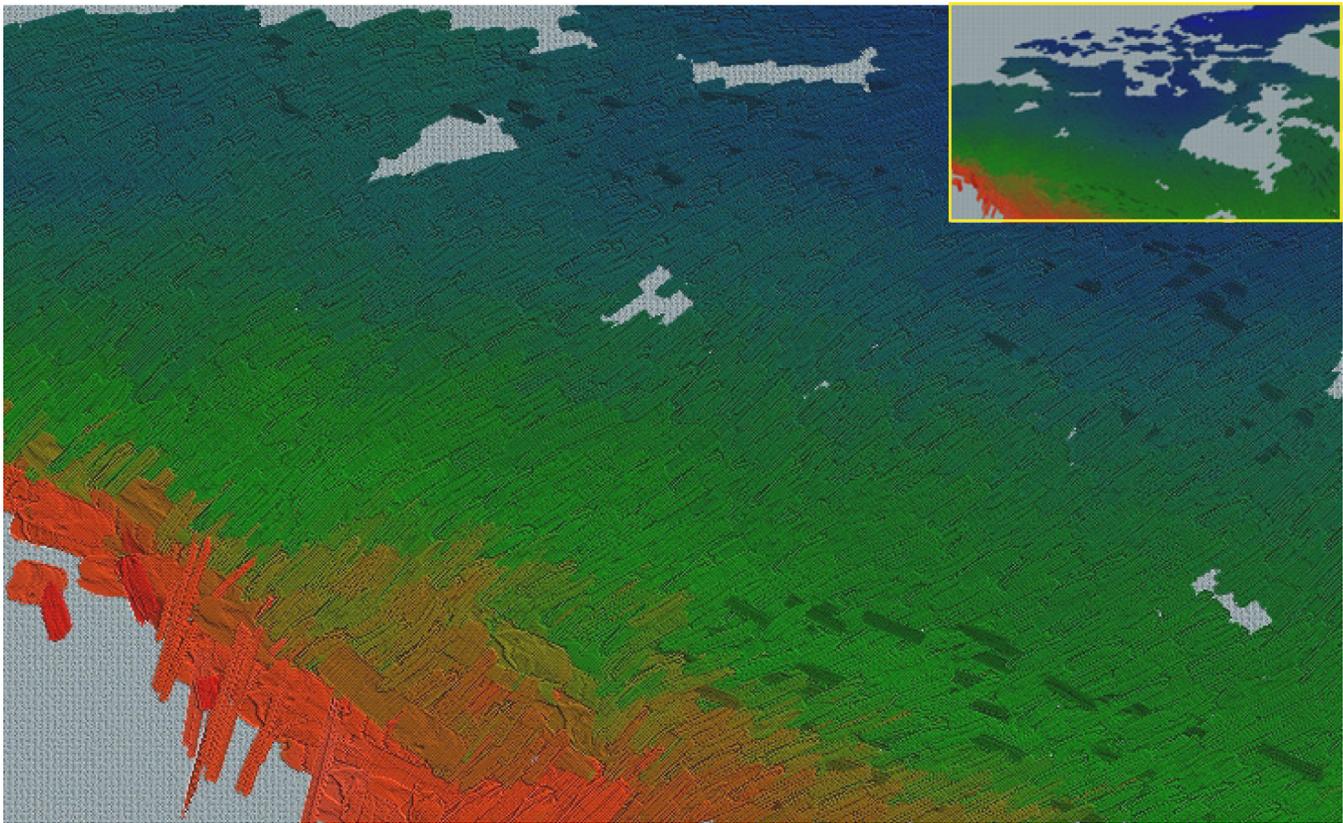
(a)

(b)

**Figure 13:** a) IC visualization of mean weather conditions for May over Africa, cloud cover → hue (blue to red for low to high), temperature → luminance (dark to light for low to high), pressure → size (small to large for low to high), diurnal range → orientation (flat to upright for low to high). b) ID visualization for October over South American, wind speed → hue, minimum temperature → luminance, radiation → size, precipitation → orientation.

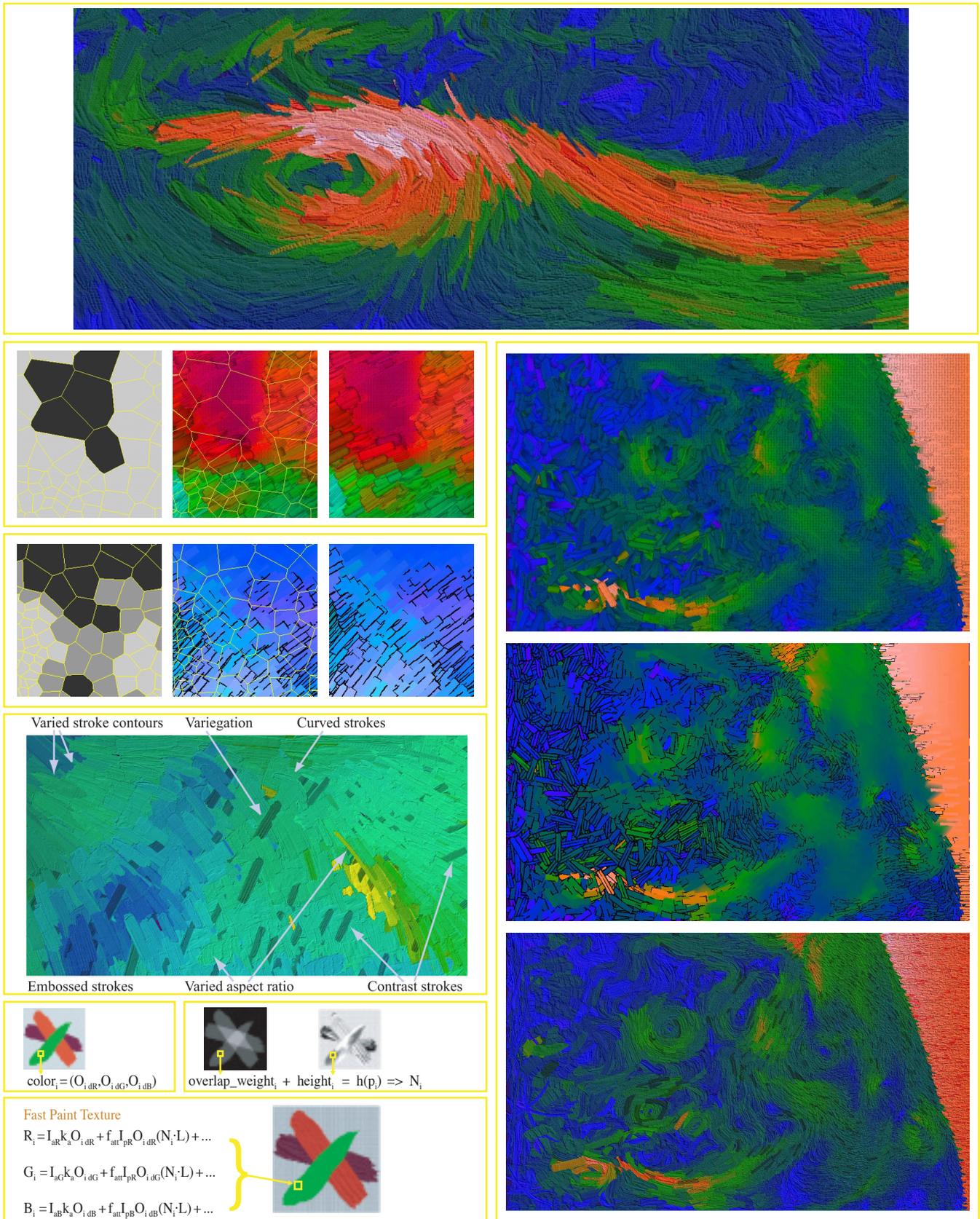
## References

- BALTISSEN, R., AND OSTERMANN, B. 1998. Are the dimensions underlying aesthetic and affective judgment the same? *Empirical Studies of the Arts* 16, 2, 97–113.
- BARBER, C., DOBKIN, D., AND HUHDANPAA, H. 1993. The quickhull algorithm for convex hull. Tech. Rep. TR GCG53, The Geometry Center, University of Minnesota, Minneapolis, MN.
- BARBER, C., DOBKIN, D., AND HUHDANPAA, H. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22, 4, 469–483.
- BERLYNE, D. 1971. *Aesthetics and Psychobiology*. Appleton-Century-Crofts, New York, NY.
- BIRKHOFF, G. 1932. *Aesthetic Measure*. Harvard University Press, Cambridge, MA.
- BROWN, KEVIN, Q. 1979. Voronoi diagrams from convex hulls. *Information Processing Letters* 9, 5, 223–228.
- DAVIS, R. 1936. An evaluation and test of birkhoff's aesthetic measure and formula. *Journal of General Psychology* 15, 231–240.
- GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, T. Whitted, Ed., 209–216.
- GARLAND, M., AND HECKBERT, P. S. 1998. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings Visualization '98*, 263–269.
- GARLAND, M. 1999. *The Design, Use, and Required Facilities of an Interactive Visual Computer Simulation Language to Explore Production Planning Problems*. Ph.D. thesis, Carnegie Mellon University, Pittsburg, PA.
- GRIGORYAN, G., AND RHEINGANS, P. 2002. Probabilistic surfaces: Point based primitives to show uncertainty. In *Proceedings Visualization 2002*, 147–153.
- HAEBERLI, P. 1990. Paint by numbers: Abstract image representations. *Computer Graphics (SIGGRAPH 90 Conference Proceedings)* 24, 4, 207–214.
- HEALEY, C. G., TATEOSIAN, L., ENNS, J. T., AND REMPLE, M. 2004. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Trans. Graph.* 23, 1, 64–96.
- HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH 98 Conference Proceedings*, M. Cohen, Ed., 453–460.
- HERTZMANN, A. 2002. Fast texture maps. In *Proceedings NPAR 2002 Symposium on Non-Photorealistic Animation and Rendering*, 91–96.
- HOLMAN, D., VERTEGAAL, R., SOHN, C., AND CHENG, D. 2004. Attentive display: paintings as attentive user interfaces. In *CHI '04: Extended abstracts of the 2004 conference on Human factors and computing systems*, ACM Press, 1127–1130.
- HSU, S. C., AND LEE, I. H. H. 1994. Drawing and animation using skeletal strokes. In *SIGGRAPH 94 Conference Proceedings*, A. Glassner, Ed., 109–118.
- KIRBY, R. M., MARMANIS, H., AND LAIDLAW, D. H. 1999. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings Visualization '99*, 333–340.
- LAIDLAW, D. H., AHRENS, E. T., KREMERS, D., AVALOS, M. J., JACOBS, R. E., AND READHEAD, C. 1998. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings Visualization '98*, 127–134.
- LEWIS, J.-P. 1984. Texture synthesis for digital painting. *Computer Graphics (SIGGRAPH 84 Proceedings)* 18, 3, 245–252.
- LITWINOWICZ, P. 1997. Processing images and video for an impressionist effect. In *SIGGRAPH 97 Conference Proceedings*,



**Figure 14:** VC visualization for January for southwest Canada (with overview in corner), temperature  $\rightarrow$  color, pressure  $\rightarrow$  size, wet day frequency  $\rightarrow$  orientation, wind speed  $\rightarrow$  contrast, precipitation  $\rightarrow$  aspect ratio.

- T. Whitted, Ed., 407–414.
- LU, A., MORRIS, C. J., EBERT, D. S., RHEINGANS, P., AND HANSEN, C. 2002. Non-photorealistic volume rendering using stippling techniques. In *Proceedings Visualization 2002*, 211–218.
- MEIER, B. J. 1996. Painterly rendering for animation. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., 477–484.
- PHAM, B. 1991. Expressive brush strokes. *Computer Vision, Graphics and Image Processing* 53, 1, 1–6.
- RAMACHANDRAN, V. 2000. The science of art: How the brain responds to beauty. In *Understanding wisdom: Sources, science, and society*, W. S. Brown, Ed. Templeton Foundation Press., Philadelphia, PA.
- RHEINGANS, P., AND EBERT, D. 2001. Volume illustration: Non-photorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics* 7, 3, 253–264.
- SALISBURY, M., ANDERSON, S. E., BARZEL, R., AND SALESIN, D. H. 1994. Interactive pen-and-ink illustrations. In *SIGGRAPH 94 Conference Proceedings*, A. S. Glassner, Ed., 101–108.
- SALISBURY, M., WONG, M. T., HUGHES, J. F., AND SALESIN, D. H. 1997. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH 97 Conference Proceedings*, T. Whitted, Ed., 401–406.
- SECOND, A. 2002. Weighted voronoi stippling. In *Proceedings NPAR 2002 Symposium on Non-Photorealistic Animation and Rendering*, 37–43.
- SHIRAIISHI, M., AND YAMAGUCHI, Y. 2000. An algorithm for automatic painterly rendering based on local source image approximation. In *Proceedings NPAR 2000 Symposium on Non-Photorealistic Animation and Rendering*, 53–58.
- STOMPPEL, A., LUM, E., AND MA, K.-L. 2002. Feature-enhanced visualization of multidimensional, multivariate volume data using non-photorealistic rendering techniques. In *10th Pacific Conference on Computer Graphics and Applications*, 1–8.
- STRASSMANN, S. 1986. Hairy brushes. *Computer Graphics (SIGGRAPH 86 Proceedings)* 20, 4, 185–194.
- TATEOSIAN, L. G. 2002. *Nonphotorealistic visualization of multidimensional datasets*. Master’s thesis, North Carolina State University.
- WALTER, J. D., AND HEALEY, C. G. 2001. Attribute preserving dataset simplification. In *VIS ’01: Proceedings of the conference on Visualization ’01*, IEEE Computer Society, 113–120.
- WILSON, B., AND MA, K.-L. 2004. Rendering complexity in computer-generated pen-and-ink illustrations. In *NPAR ’04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, ACM Press, 129–137.
- WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *SIGGRAPH 94 Conference Proceedings*, A. Glassner, Ed., 91–100.
- WINKENBACH, G., AND SALESIN, D. H. 1996. Rendering parametric surfaces in pen-and-ink. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., 469–476.
- WOODING, D. 2002. Eye movements of large populations: Ii. deriving regions of interest, coverage, and similarity using fixation maps. *Behavior Research Methods, Instruments, and Computers* 34, 4, 518–528.



**Figure 15:** Top, supernova closeup (Fig. 1); left column, top to bottom: IC process (Fig 4), ID process (Fig 6), VC components (Fig. 7), texture procedure (Fig. 9); right column, top to bottom: IC supernova visualization, ID supernova visualization, VC supernova visualization (Figs. 10, 11)