THE PERCEPTUAL OPTIMIZATION OF 2D FLOW VISUALIZATIONS USING HUMAN-IN-THE-LOOP LOCAL HILL CLIMBING

BY

PETER W. MITCHELL
BS, McGill University, 1985

THESIS

Submitted to the University of New Hampshire

In Partial Fulfillment of

The Requirements for the Degree of

Master of Science

In

Computer Science

December, 2007

This thesis has been examined and approved.

Thesis Director, Colin Ware,
Professor of Computer Science

R. Daniel Bergeron,
Professor of Computer Science

Alejo Hausner,
Assistant Professor of Computer Science

Date

DEDICATION

This thesis is dedicated to my wife, Denise, and to the rest of my family who have allowed and encouraged me to complete this work.

ACKNOWLEDGEMENTS

I hereby acknowledge and thank my thesis advisor and mentor, Colin Ware, and my other thesis committee members, Dan Bergeron and Alejo Hausner, for their expertise and guidance through this process. I thank the people who took the time to be subjects: Colin Ware, Stephen Schaeffer, Briana Sullivan, Dave Brown, Fritz Drury, Robert Brinkerhoff, and John Kelly. Thank you also to everyone else who offered technical, informational, and/or logistical support throughout this research, including David Laidlaw, Don House, John Murphy, Matt Quinn, Matt Plumlee, and Roland Arsenault. And finally I must acknowledge my wife and family, whose support made this possible.

This research was funded in part the National Science Foundation grant ITR 0324899.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

THE PERCEPTUAL OPTIMIZATION OF 2D FLOW VISUALIZATIONS USING HUMAN-IN-THE-LOOP LOCAL HILL CLIMBING

by

Peter W. Mitchell

University of New Hampshire, December, 2007

Flow visualization is the graphical representation of vector fields or fluids that enables an observer to visually perceive the forces or motions involved. The fields being displayed are typically dynamic and complex, with a vector direction and magnitude at every point in the field, and often with additional underlying data that is also of interest to the observer. Distilling this mass of data into a static, two-dimensional image that captures the essential patterns and features in a way that is intuitively understandable can be a daunting task.

Historically, there have been many different techniques and algorithms to generate visualizations of a flow field. These methods differ widely in implementation, but conceptually they involve the association of significant aspects of the data field (e.g., direction, velocity, temperature, vorticity) to certain visual parameters used in the graphic representation (e.g., size and orientation of lines or arrows, foreground and background color, density/sparsity of graphical elements). For example, the velocity of a field could be mapped to color, line width, line length, arrow head or glyph size, etc. There are many such potential parameter mappings within each technique, and many value ranges that can be used to constrain each parameter within a given mapping, resulting in a virtually limitless number of possible permutations for visually representing a flow field. So, how does one optimize the output? How can one determine which mappings and what values within each mapping produce the best results? Such optimization requires the ability to rapidly generate high-quality visualizations across a wide variety of parameter mappings and settings.

We address this need by providing a highly-configurable interactive software system that allows rapid, human-in-the-loop optimization of two-dimensional flow visualization. This software is then used in a study to generate quality visual solutions to a two-dimensional ocean current flow plus surface temperature over a variety of parameter mappings. The results of this study are used to identify relevant rules and patterns governing the efficacy of each combination of parameters, and to draw some general conclusions concerning 2D flow visualization parameter mapping and values.

INTRODUCTION

The goal of scientific visualization is to graphically display large and/or complex data sets such that viewers can accurately and intuitively perceive both quantitative and qualitative aspects of the underlying data. Creating effective visualizations depends on many factors, including the type and quantity of the data, the task that users of the data will perform, and other constraints such as the medium on which the visualization is displayed or the time allotted to generate it. One area of particular interest is *flow visualization*, the visualization of fluids in motion. Yet, while many papers have been written on methods for rendering flows, information on what constitutes effective flow visualization is still largely anecdotal, and many practical applications use simple but clearly suboptimal methods (figures 1 and 2).

Laidlaw et al. [LDM01] conducted a user study comparing several common methods for depicting two-dimensional flow in an attempt to bring a more scientific approach to identifying the advantages and disadvantages of each technique, though without allowing any variation of parameters *within* each technique that might potentially improve or degrade the visualization for a given task. Due to the sheer number of possible methodologies and the number of possible parameter variations within each methodology, it is virtually impossible to carry out any kind of exhaustive experiment covering all (or even most) combinations. However, by selecting a single flexible (parameterized) technique capable of producing a wide variety of flow visualization styles, we can conduct an experiment in which several of the underlying parameters are varied and evaluated.

A further factor complicating effective flow visualization (not addressed by Laidlaw et al.) is the additional display of other static data related to the flow (e.g., temperature, salinity, depth). One common method for displaying such data is via background color. Clearly, the requirement of perceiving relevant colors in the background affects the choices driving the flow field rendered over it.

This thesis describes a human-in-the-loop optimization technique whereby the parameters used to render a two-dimensional visualization of ocean surface currents against a background representing surface temperature can be modified and evaluated. The supporting software is able to interactively generate visualizations over a large variety of parameters and mappings. The goal is to evaluate the utility of the approach and supporting software via an evaluation study, and perhaps to derive some patterns or universal truths regarding effective flow visualization in two dimensions.

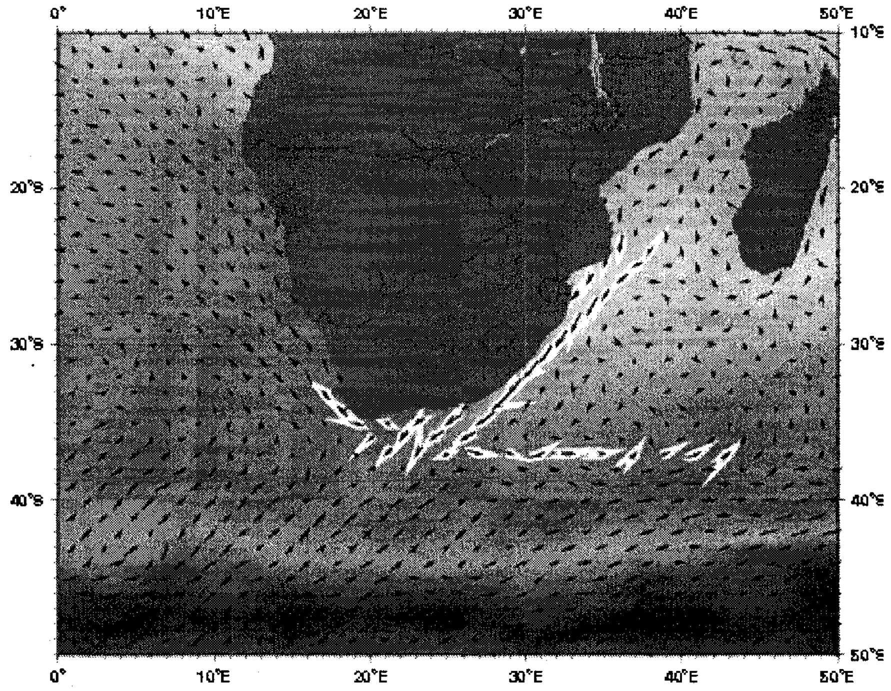


Figure 1: Image from an existing University of Miami web page shows surface currents in the Indian Ocean. No key is provided. [UMi07]

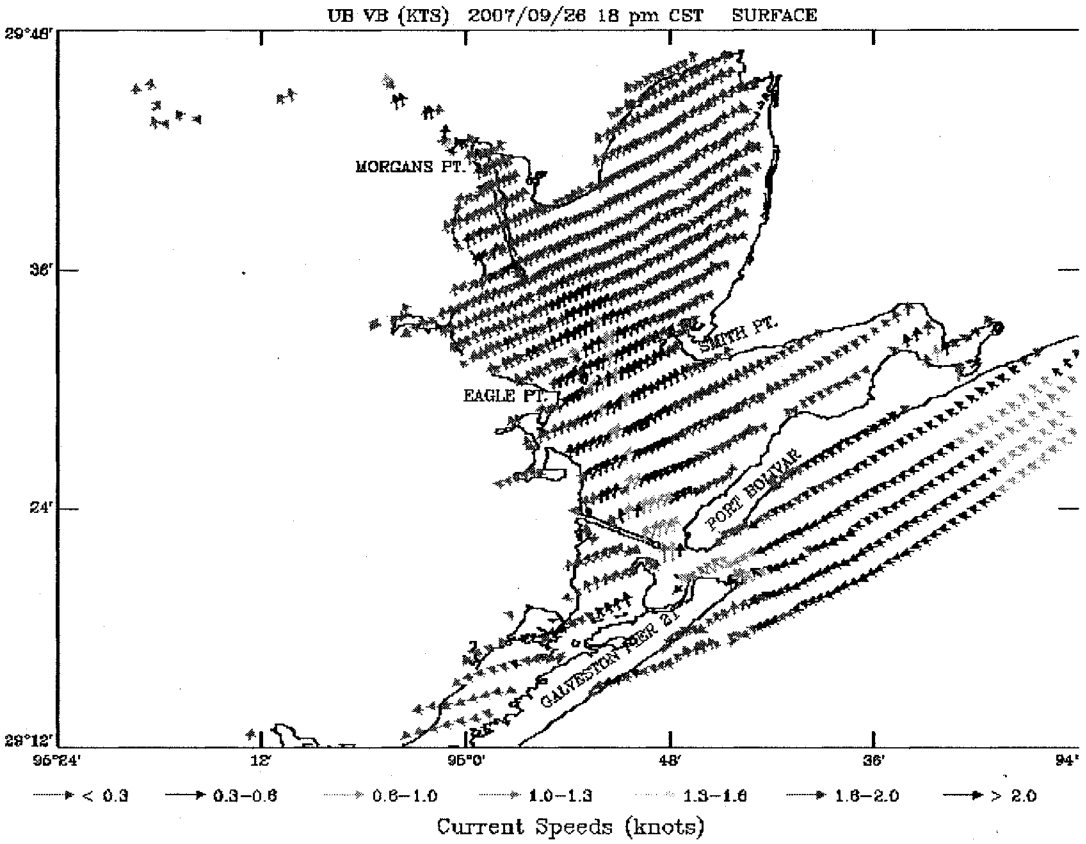


Figure 2: Image from an existing NOAA web page shows surface currents of Galveston Bay using arrows to indicate direction and arrow color to represent velocity. [NOA07]

CHAPTER 1

BACKGROUND

1.1 Flow Visualization

Flow visualization is a major focus in computer-aided visualization research, with applications spanning many different disciplines, including meteorology (wind and air currents), aerodynamics, oceanography (tides and currents), fluid dynamics, magnetic fields, and medicine (blood flow), to name but a few. Flow data may be formulaically derived, as in a representation of an ordinary differential equation (ODE), sampled from a computer-generated flow simulation, or empirically derived via measurements on an actual flow. Since effective visualization is intimately tied to both the nature of the flow and the task being performed, approaches and techniques are as varied as the information being harvested.

While the basic concept of flow visualization is simple – the representation of a vector field – it is both the nature of the field and the specific information to be conveyed to the observer that drives the myriad permutations of techniques. For example, a flow may be one-dimensional, two-dimensional (2D), three-dimensional (3D), or even *n*-dimensional. Flow fields may be steady, where the vector at each point in the field does not change over time, or time-variant, where the flow field is continually varying or cycling. Someone doing disaster modeling of coastal oil spills would be interested in advection paths, which are the paths a particle would travel if dropped into the flow at particular points. An aeronautical engineer studying the aerodynamics of a flow over a car or airplane wing would likely focus on the velocity, vorticity (curl), and turbulence in certain key areas. A meteorologist tracking the eye of a storm might be concerned with flow singularities, which are points of zero velocity like sources, sinks, or saddle points.

While the majority of real-world flows are three-dimensional and time-variant, these are often rendered as two-dimensional cross-sections or layers [PVH02]. This is natural because the human visual system is especially effective at 2D perception. The introduction of occlusion in a 3D model is a huge complication.

At a minimum, a flow field consists of a velocity and a direction at all points in a plane or volume (and at any given time, for a time-variant flow). A steady flow field (or a time-variant flow field at a specific point in time) can be represented by contour lines, everywhere tangential to the direction of flow. Such contour lines, however, are ambiguous; there is no clear indication of which way the field flows along the line. We consider such a contour line to have *orientation*, but not *direction*. While these terms are often used interchangeably (or distinguished conversely), we find it helpful to distinguish between them. The addition of arrow heads, luminance changes along the line, or the overlaying of glyphs are a few of the ways that direction can be indicated (see *Section 1.2.6 – Direction perception*). The purpose of drawing this distinction is that some methods of flow visualization, particularly among texture-based algorithms, depict orientation only.

In general, flow visualizations are used to convey one or more of the following characteristics [War04]:

- Critical points (locations of zero velocity, e.g., sources, sinks, saddle points)

- Advection trajectory (path of a particle dropped at a certain point)
- Areas of high and low velocity
- Areas of high and low vorticity (curl)
- Areas of high and low turbulence

In addition, users may need to identify specific values for the velocity or other static variables associated with the flow at a given point. For ocean currents, for example, users may be interested in surface temperature, salinity, or depth.

1.2 Perceptual issues in flow visualization

There are a number of perceptual issues that help to anticipate and justify certain results relating to flow visualization. While a comprehensive background is beyond the scope of this thesis, we touch on several key issues. For more in-depth discussion on these and other perceptual issues, refer to Ware's *Information Visualization: Perception for Design* [War04].

1.2.1 Color channels

The human retina contains two types of photoreceptors: rods, which are only active in low light levels, and cones, which are active in normal light. Cones are further divided into three types, each having peak sensitivity at different wavelengths: long (red), medium (green), and short (blue). Of the three, the blue cones are by far the sparsest and least sensitive.

Hering [Her20] proposed that color is actually perceived in three orthogonal color-pair channels: black-white, red-green, and yellow-blue. Modern research in color perception and physiology appear to confirm Hering's opponent process theory, which is the foundation of the majority of color theory to this day [War04]. The color channels are interpreted from the retinal cones by the brains as follows:

- Black-white (luminance): The sum of inputs from all three cone types, though blue plays virtually no role in luminance.
- Red-green: The difference in input between the red and green cones.
- Blue-yellow: The difference in input between the blue cones and the sum of the red and green cones.

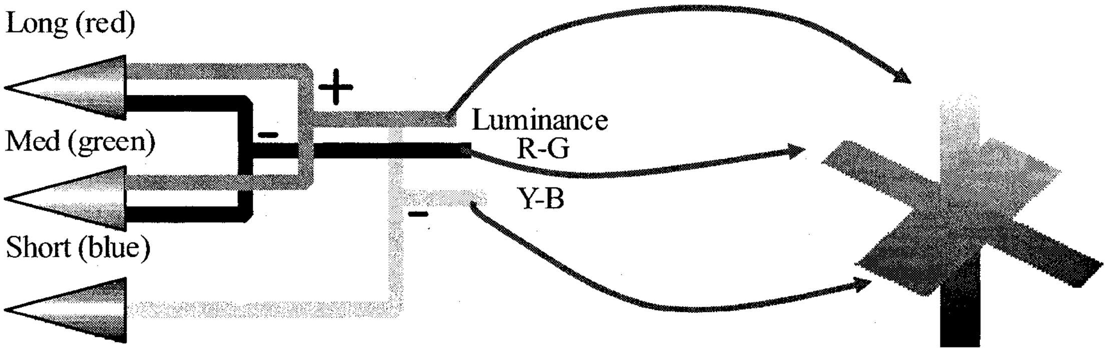


Figure 3: Color channels. [War04]

The luminance (black-white) channel is most effective for representing detail, so patterns will be most apparent when using colors of contrasting luminance. We would expect isoluminant colors, even in greatly varying hues, to be considerably less effective in representing flow patterns.

Color blindness generally affects the red-green channel as it is most commonly caused by a lack of long-wavelength (red) or medium-wavelength (green) receptors.

1.2.2 Categorizing attribute data

Data attributes can be generally grouped into three types – nominal data, ordinal data, and quantitative data. Nominal data names or categorizes an entity, ordinal data assigns an ordered value such that it is possible to determine whether one entity comes before or after another, and quantitative data is measurable such that the value of one entity can be compared to another in numeric terms (e.g., as a ratio).

For example, consider a map of the United States, with each state representing a single entity. Nominal data might include primary industry, majority religion, or political bent. Each of these represents a label or category. Ordinal data might include standard of living rank or order of acceptance into the union. Quantitative data might include population, average income, or number of electoral votes.

This categorization of data is relevant since certain visualization parameters are more conducive to represent certain types of data.

Nominal data is best represented by a visualization parameter that can be easily named. One possible choice is color or, more specifically, hue. This is especially effective if the number of categories is relatively small (i.e., six to ten), since there are limits to the number of hues we can uniquely identify.

For ordinal data, we need a visualization parameter that represents a sequential order. Saturation (of a particular hue) and luminance can be effectively used to represent ordinal data, as can changes in size. Hue can be used providing that the hue colors fall across an identifiable color channel (e.g., red to green). Note that in the absence of a key, "warmer" colors (e.g., reds, yellows) tend to be perceived as higher values while "cooler" colors (e.g., blues, greens) are perceived as lower values.

Quantitative data requires a visualization parameter that can be measured or estimated for a particular value. Size (length and width) are typically most effective for this type of data. If color is used, it must be accompanied by a key, since people cannot accurately determine whether one entity is, say, 25% brighter (brightness is not linear) or 50% greener.

1.2.3 Integral versus separable dimensions

When assigning visualization attributes to represent data values, it is important to realize that some combinations of these attributes are not perceived separately, but rather as an integral whole [War04]. Color is a prime example. If we are selecting visualization attributes for two parameters (say, temperature and salinity), it is ineffective to represent one by an amount of red saturation and another by an amount of green saturation. This is because the brain does not perceive combinations of red and green as separable entities – it perceives the resultant combination as a single integral color (e.g., yellow).

This phenomenon may affect the viability of using non-opaque colors on streaklets to represent a value such as velocity when color is also being used in the background to represent temperature.

1.2.4 Lightness and chromatic contrast

Even when using fully opaque colors in the foreground, the human optical system perceives colors and luminance relative to the local environment. This means that the choice of background color (or colors) may bias the way we perceive colors in the foreground. In figure 4, the two Xs are the same color, however against a red background we perceive its "blueness", while on a blue background we perceive its "redness".

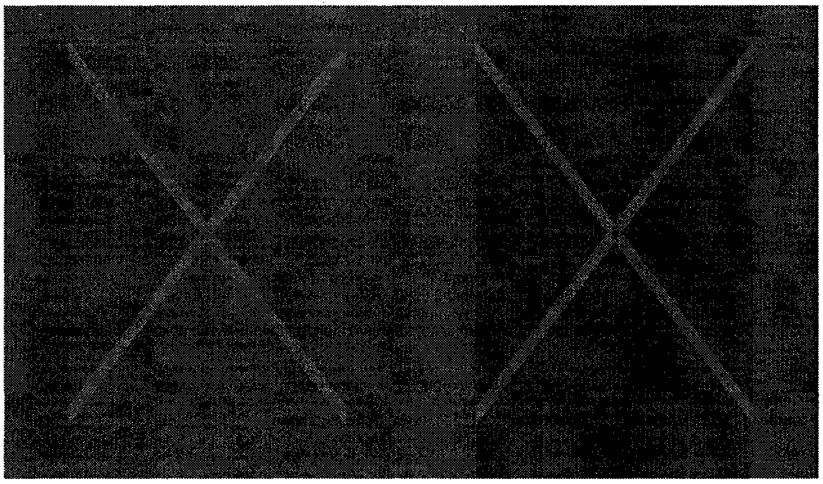


Figure 4: Chromatic contrast. Hue is perceived relative to background. [War04]

This local contrast effect applies to luminance as well. Perhaps one of the most stunning examples of this can be seen in the following figure from MIT professor Edward Adelson. The squares marked ‘A’ and ‘B’ on the chessboard are the exact same shades of gray, as can be seen by the superimposition of vertical bars of the same shade. However, our brains perceive each square’s shade relative to its immediate surrounding, so square ‘A’, being surrounded by much lighter squares, is perceived as dark and square ‘B’, amidst much darker squares, appears light.

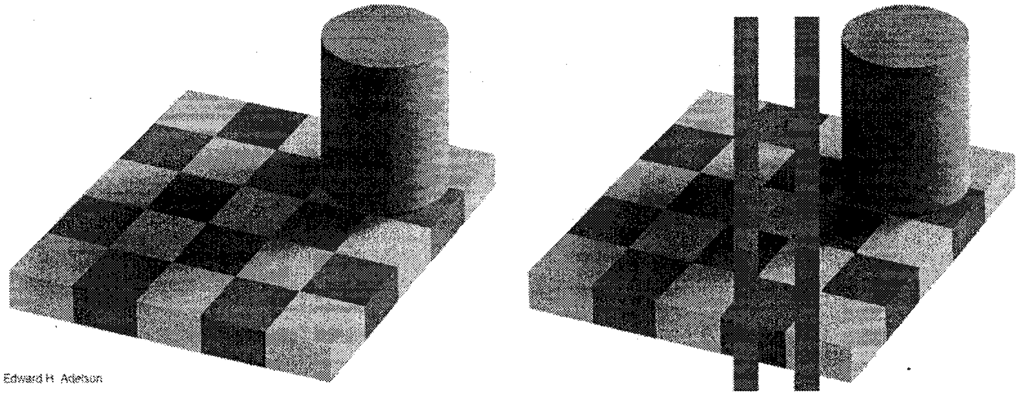


Figure 5: Luminance contrast. Shade is perceived relative to local area. [Ade95]

1.2.5 Contour perception

In the early twentieth century, a group of German psychologists developed a set of *Gestalt Laws* to explain pattern perception [Kof35]. One of the principles is that of *continuity*. In other words, a line that defines a smooth, continuous curve is more likely to be perceived as a single continuous entity. In figure 6, the pattern in (a) is perceived to be the combination of the two entities in (b), not the two entities in (c). Field et al. [FHH93] note that elongated elements situated along a continuous curve are similarly perceived. Ware [War04] applies this phenomenon to the perception of vector fields, suggesting that vector flows should be more easily perceived with arrows placed tail-to-head to form continuous contours. This theory is further supported by the work of Laidlaw et al., [LDM01] who note that visualizations that consist of integral curves perform better on all tasks than those using grid or random distribution patterns.

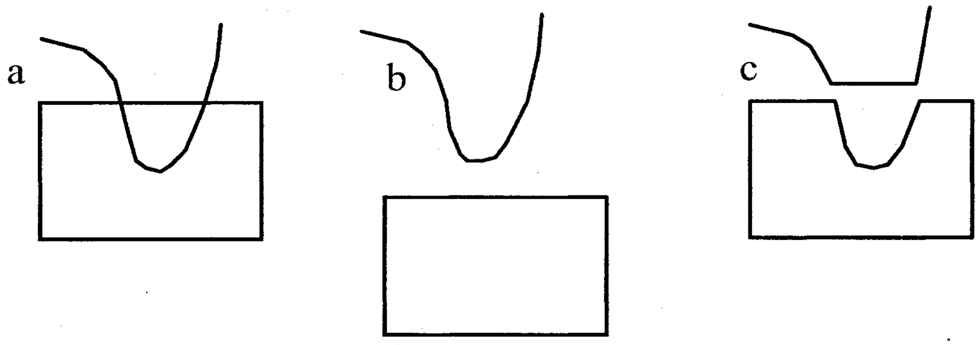


Figure 6: Gestalt Law: Continuity. The brain perceives 'a' as being comprised of the two elements in 'b', not 'c'.

1.2.6 Direction perception

As indicated earlier, contour lines everywhere tangential to the field have orientation, but not direction. In other words, there are two opposing directions for any given orientation. For example, a road may be oriented east/west, while the cars traveling on it will have a direction that is either east or west.

One method for indicating direction is the use of arrow heads (figure 7). While this is generally unambiguous, the use of arrow heads can cause problematic clutter, especially in dense flow visualizations where contour lines are very close together.

Fowler and Ware [FW89] apply Reeves' Particle System [Ree83] to flow fields. Scattering particles across the field and tracking each one as it moves through the field for a specified lifespan results in a particle trace they call a *stroke.* Attributes of a stroke (e.g., color, size) can be varied as the stroke ages and/or can be mapped to some static data value at each point. These attribute mappings can be effectively used to impart flow magnitude and direction. In figure 8, the strokes are clearly moving left to right.

Results of the Fowler and Ware study indicate that direction is most effectively indicated by the interaction of the stroke color and the background color, with the tail of the stroke blending into the background and the head of the stroke contrasting sharply. Variable width is a secondary indicator, with the wider end of the stroke generally perceived to be its head.

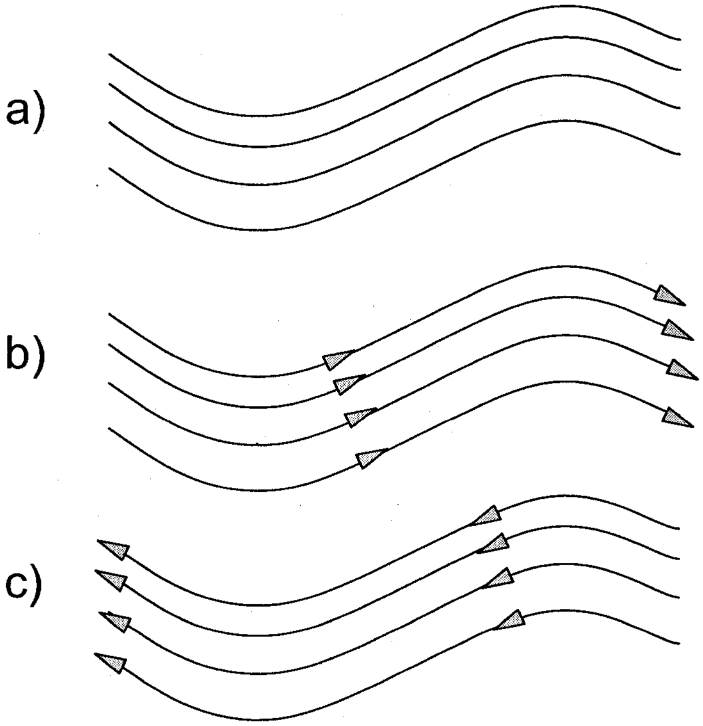


Figure 7: Use of arrow heads to disambiguate direction.

Figure 8: Fowler and Ware's "strokes" are generally perceived here to be moving from left to right. [FW89]

1.3 **Flow visualization techniques**

Visualization techniques can be broadly categorized as *direct*, *texture-based*, *integration-based*, or *feature-based*, representing dramatically different methodologies for rendering images of flow fields. We will briefly describe each of these methods in the following sections before delving into the specific integration-based techniques implemented for this research. Figure 9 shows the results of three different techniques: direct visualization using arrows, texture-based visualization using line integral convolution (LIC), and integrated visualization.

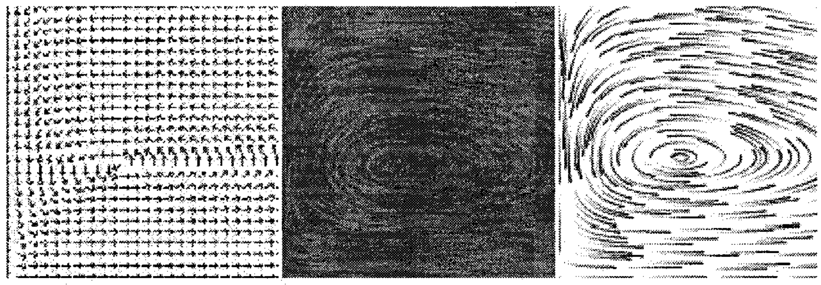


Figure 9: Flow visualization techniques, from left: arrow plot, line interval convolution, and integrated streaklets. [PVH02]

1.3.1 Direct flow visualization

Direct flow visualization maps the velocity vector at each given point directly to a graphical icon or color. This is the simplest mechanism, as there is a one-to-one correlation between the graphic representation at each point and its underlying vector value. A common example of this is the arrow plot (see figure 9), where small arrows are placed across a grid to show the direction of flow at each point. While this method accurately displays flow direction at the grid points, there are many drawbacks and issues.

One concern with the direct arrow plot is aliasing. Placing arrows across a grid or other regular pattern can result in visual artifacts that cause the user to perceive false patterns related to the regularity that are not in the flow. This effect can be addressed by introducing some form of randomization to the placement of the arrows [DW85]. While there are several methods that can be applied, one of the simplest is the "jittered grid", whereby the point at which each arrow is rendered is adjusted by a small random amount. However, even with jittering, the direct arrow plots are considerably inferior to other visualization techniques for identifying critical points and advection paths [LDM01], do not inherently represent vector magnitude, and are not visually intuitive.

1.3.2 Dense, texture-based flow visualization

Texture-based flow visualizations provide dense images of a flow field through the convolution of a texture across the flow's vector field. Introduced by van Wijk [van91], "Spot Noise" was the first of these methods, whereby small spots are randomly distributed over a flow field and "smeared" based on the local vector values. Cabral and Leedom [CL93] then presented "Line Integral Convolution" (LIC), claiming it to be a more generally applicable method than Spot Noise, for which results often depend on the relative size of the rendered spots compared to the flow vector at each point. In LIC, a texture, typically white noise, is convolved across the flow field through the application of a one-dimensional filter based on a curved streamline segment oriented to the local vector values at each point.

LIC has become the de facto standard for dense, texture-based flow visualizations, and many extensions and improvements have been published, notably the Fast LIC algorithm of Stalling and Hege [SH95], which reduces the computation time by an order of magnitude, and the introduction of Oriented Line Integral Convolution (OLIC) by Wegenkittl et al [WGP97]. OLIC addresses one of the principal drawbacks of LIC – that visualizations depict the orientation of flow but not direction – by using a ramplike convolution kernel to show direction and velocity, similar to Fowler and Ware's strokes (see *Section 1.2.6 – Direction perception*). (*Note: Wegenkittl et al.'s use of the terms "orientation" and "direction" are opposite to the definitions used in this paper.*)

1.3.3 Feature-based flow visualization

Originally developed by Helman and Hesselink [HH91], feature-based flow visualization relies on the preprocessing of data to abstract key topological features, such as sources, sinks, or saddle points. This allows subsequent rendering to focus on what are assumed to be the areas of interest. Feature-based flow visualization is especially useful for very large, time-variant flow fields where the sheer amount of data makes direct or integrated approaches uninterpretable. While the abstraction can be time-consuming, once the preprocessing is complete and the critical features have been isolated, visualizations can be rendered without referring back to the original data [PVH02]. Topological features can be portrayed using simple icons or glyphs [PPv95], reducing a large, complex field to a barebones representation of the important qualitative features (see figure 10).

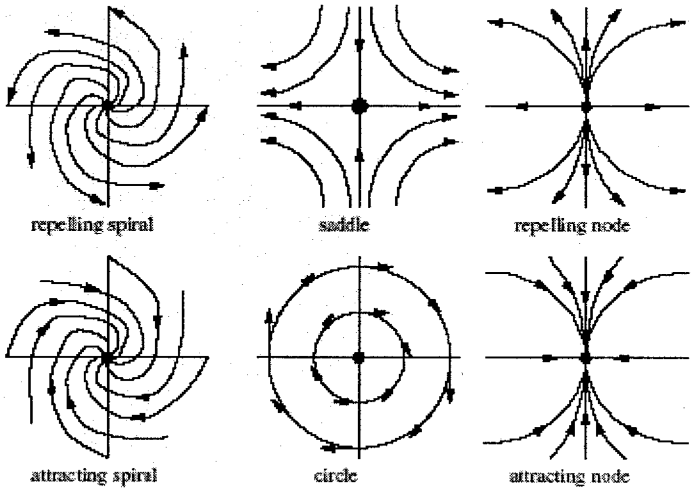


Figure 10: Sample critical points – from [SHK97]

1.4 Integration-based flow visualization

Integration involves the approximation of a curve represented by the flow field using a series of small line segments. Common curves that are integrated include streamlines, pathlines, or streaklines. Streamlines are curves that are always tangent to the velocity vector of the flow. Pathlines trace the path that a point would follow if dropped into the flow. Streaklines are the summation of all points that have flowed through a given point in the past. While these differences are significant in time-variant flow fields, for steady fields, which do not vary over time, they are the same.

1.4.1 Integrating streamlines

The integration of a streamline in a non-time-variant flow starts with the selection of a seed point followed by the application of a vector value across a small time increment. The simplest method of integration is Euler's method, whereby each subsequent point is derived based on the vector value at the previous point, as follows (V_i = vector value at point P_i):

$$\mathbf{P_{n+1} = P_n + (V_n \cdot \Delta t)}$$

The tradeoff to this computational simplicity is lack of accuracy, especially for nearly circular curves (Euler's method will render a circle as a spiral). This error can be reduced dramatically by implementing a second-order Runge-Kutta method, which calculates the vector value at P_n and P_{n+1} (as with Euler), but then averages the two vectors together and reapplies the mean to the original point, as follows:

$$\mathbf{P_{temp} = P_n + (V_n \cdot \Delta t)}$$

$$\mathbf{P_{n+1} = P_n + ((V_n + V_{temp}) / 2) \cdot \Delta t)}$$

1.4.2 Streaklets and glyphs

One of the advantages of using an integrated streamline approach to flow visualization is the ability to overlay glyphs along each generated streamline. A glyph is a graphical symbol that is used to describe multivariate data [War04]. Each attribute of a glyph can be mapped to an underlying data value in the flow (velocity, temperature, salinity, etc.).

One such glyph, introduced above (see *Section 1.2.6 – Direction perception*) is Fowler and Ware's concept of the *stroke*. We use the stroke concept; however rather than rendering strokes from randomly scattered start points, we desire the additional benefit of continuous contours (see *Section 1.2.5 – Contour perception*). Therefore, we render them head-to-tail along integrated streamlines as a sort of glyph. We call these stroke glyphs *streaklets*. Streaklet attributes include length, width, color, and opacity. As with strokes, these attributes can be varied as the streaklet ages, or can be mapped to any data value at each point in the field. Streaklets can also include a circle or arrow head as an additional direction indicator.

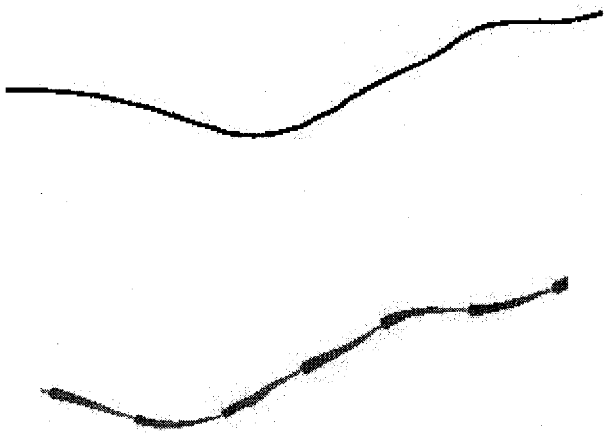


Figure 11: A streamline contour (top), and a streamline with streaklets laid head-to-tail along it (bottom).

1.5 Evenly-spaced streamlines

One goal of effective streamline flow visualization of a 2D non-time-variant field is often to control, and more specifically to homogenize, the density of the streamlines. Turk and Banks [TB96] proposed an image-guided method of placing evenly-spaced streamlines whereby a visualization is iteratively modified and rated using an energy function until a desired threshold of density spacing is obtained. While able to produce very high quality results, the iterative change and test process is quite time consuming and thus not readily applicable where rendering time is critical.

Jobard and Lefer [JL97] followed with a single-pass algorithm to generate evenly-spaced streamlines of an arbitrary density. While often yielding results slightly inferior to those produced by Turk and Banks, the drastically reduced rendering time (seconds as opposed to minutes) allows the control of streamline density in an interactive environment. Furthermore, the single-pass algorithm can be extended to create streamlines that vary in density based on an underlying static value.

Since the Jobard-Lefer algorithm forms the basis of the coding done for this paper, it is described in more detail below.

1.5.1 The Jobard-Lefer algorithm [JL97]

Jobard and Lefer proposed and developed a high-performance method of generating evenly spaced streamlines at *any* user-defined density. This approach can be used to create flow visualizations that simulate dense representations (e.g., LIC) as well as the sparse, hand-drawn style specifically addressed by Turk and Banks [TB96].

The algorithm involves generating a single streamline, with subsequent streamlines being deliberately placed relative to those that already exist such that the separating distance approaches the user-specified value (d_{sep}) for the desired density. This is done by ensuring that all possible lines are generated relative to a single streamline before moving on to the next. Each streamline integrates both forward and backward until it becomes too close to an existing line, approaches zero velocity (i.e., hits a "source" or "sink"), or iterates beyond the viewable area. Equally-spaced *control points* are placed along each streamline at a distance slightly less than the separation value; these points represent the streamline and are used to increase efficiency, instead of testing against every point in the line. New streamlines are generated from an existing streamline by attempting to place a new seed point to the left and right of every control point on the existing streamline. The algorithm can be described as follows:

```
Generate a streamline with equally spaced control points and put in queue;
For each streamline in queue:
    For each control point on the streamline:
```

```
Determine two candidate points ($d_{sep}$ away on each side of streamline)
For each candidate point:
    If candidate point is valid seed ($d_{sep}$ away from *all* streamlines)
        Generate a new streamline at candidate point and put in queue
```

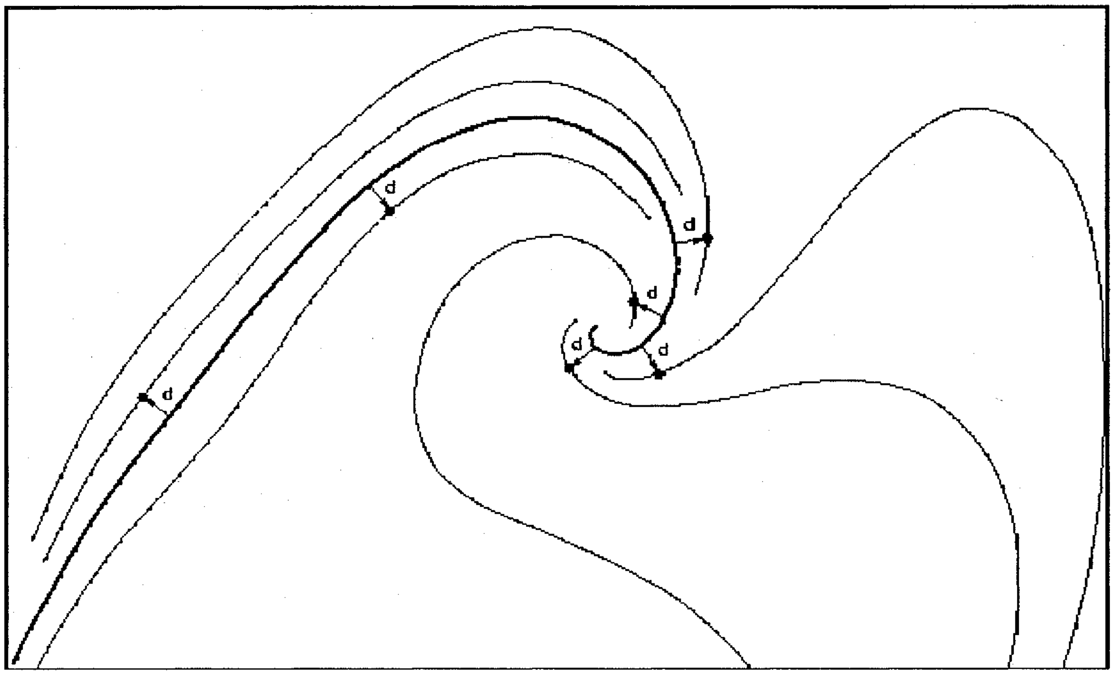


Figure 12: "Streamlines are derived from the first (thick) one by choosing seed points (circles) at a distance $d=d_{sep}$ from it." [JL97]

Since performance is a key goal, it is important to reduce the calculation required to determine the candidacy of each potential seed point (i.e., we do not want to determine the distance between it and every point of every existing streamline). Jobard and Lefer accomplish this in two ways:

1) Store regularly spaced control points along generated streamlines. By ensuring that these points are closer together than d_{sep}, we can make a

reasonable approximation of acceptability by testing against these control points.

2) By overlaying a Cartesian grid of cell size d_{sep} across the flow and storing each streamline's control points in the appropriate cell, a candidate point need only compare itself against control points in the same cell and the eight adjacent cells.

In order to produce good visual results, it is necessary to allow some tolerance against the constraining d_{sep} value when integrating a streamline, such that streamlines can approach each other by some percentage of the d_{sep} value. This tolerance value is called d_{test}. If d_{test} is too close to d_{sep}, streamlines tend to be choppy, as even a slight convergence from the newly integrating streamline toward its spawning line causes streamline generation to stop in that direction. Jobard and Lefer suggest that a d_{test} value of 0.5 is ideal (i.e., during streamline generation, a streamline can come to within $d_{sep}/2$ of another).

1.6 Application of techniques to thesis goals

The goal of this thesis, as stated above, is to develop a highly configurable, human-in-the-loop software program capable of generating and optimizing a large number of visualizations interactively.

We use an integrated streamline approach, specifically an adaptation of the Jobard-Lefer technique, for generating evenly-spaced streamlines representing flow contours with head-to-tail continuity for optimal perception. This technique is parameter driven, allowing direct control of streamline spacing, and thus is capable of rendering a variety of styles from dense textures (similar to Line Integral Convolution) to sparser hand-drawn styles. It is fast, thus conducive to interactive manipulation of parameters, which we allow via the use of slider controls, and it is also adaptable, which we leverage to create *parameterized* streamline spacing (yet another possible data mapping). This adaptation is described in detail in later sections. The use of highly-configurable streaklets in conjunction with such a flexible streamline-generation technique creates an immense number of possible parameter-to-data mappings with a virtually unlimited number of parameter combinations within each mapping.

CHAPTER 2

HUMAN-IN-THE-LOOP OPTIMIZATION

2.1 Local hill climbing

The difficulty of obtaining high-quality visualizations is a result of the virtually endless combinations of parameter mappings and values. Streaklet color could be used to indicate speed, direction, or temperature; streaklet width could be used to indicate the same, or even a combination of direction and speed; background color could indicate current velocity or temperature; etc. Even within a single mapping, the combinations are staggering. Consider, for example, using streaklet color to represent the velocity of the current. In the simplest case, two colors must be selected, one to represent the minimum velocity (or low end of the scale) and one to represent the maximum velocity (or high end of the scale), with the actual color calculated as a linear combination of the two extremes, based on actual velocity at a given point. It is simply not possible to generate and evaluate an exhaustive set of all possibilities.

Local hill climbing is a methodology for identifying good solutions, based on the premise that the quality of a solution is a continuous value across its parameter space. In other words, we expect solutions to be locally surrounded by other solutions of similar quality. If we imagine the parameter space to be two-dimensional grid (it is actually of much higher dimensionality that this, but this model serves to explain the concept in a manner that we can visualize), with the quality of the solution being a third dimension, height, then we could view the solution space as a contour map, with the highest hills representing the best solutions and the lowest valleys representing the poorest ones. In order to locate hill peaks, one can randomly evaluate solutions across the parameter space to locate points of relatively high "altitude", and then focus in on those locations by iteratively varying each parameter slightly to determine whether the change results in a higher value, until the local peak has been identified.

2.2 Human-in-the-loop optimization

As noted above, the local hill climbing process requires the iterative generation of solutions and the subsequent evaluation of those solutions to feed the next generation. Each iteration cycle could take hours, or even days, depending on the logistics of the methodology being used.

We provide an interactive software interface to allow configurable parameter mappings, random parameter generation within a selected mapping, and human-in-the-loop control of these parameter values using slider controls. This ability to rapidly change values and receive immediate feedback allows local hill climbing to be implemented interactively, allowing good solutions to be generated in a time-efficient manner that is just not possible using existing methods.

This software is then used to run an evaluation study to validate the process. In theory, this should result in combinations of mappings and parameters that produce a visually-intuitive rendering of a flow that provides clear details of the underlying multivariate data.

CHAPTER 3

SYSTEM SOFTWARE

3.1 System overview

The purpose of the software is to provide a flexible system to control the mappings between flow model data and its visual representation via a set of streaklets, and to allow human-in-the-loop interaction to vary the individual parameter settings in an attempt to optimize the visualization.

The software is written for the Windows® operating system using Microsoft Visual C++®, using OpenGL®, glut, glui and netCDF libraries.

3.2 Parameter mapping

The basis of this research is the ability to map visualization parameters to flow data parameters in a variety of combinations. Visualization parameters include streaklet opacity, color, width, and length; streamline density; background color, and the absence or existence and size of a streaklet head. Flow data parameters include direction, velocity, and surface temperature. Other data, such as surface salinity, could also be included in flow data parameters, though they are not evaluated in this research.

Once a visualization parameter is mapped to a data parameter, the user interface will allow the user to set the range (minimum and maximum values) of each visualization parameter. The application of these minimum and maximum settings are dependent on the data mapping, and will be explained the appropriate subsections of *Section 3.4 – Data parameters*.

3.3 Visualization parameters

Visualization parameters directly control the attributes of the streaklets that are generated for the flow field. Most visualization parameters have corresponding *minimum* and *maximum* values. These are mapped to the extremes of the data values to which they are mapped.

3.3.1 Streaklet opacity

The opacity of the streaklet is a percentage value from 0 to 1, with 0 indicating completely transparent and 1 indicating totally opaque. Recall that opacity is a good visual indicator for ordinal representation, but not for quantitative measurements (*Section 1.2.2 – Categorizing attribute data*). This means opacity is particularly well-suited to show direction. As seen in figure 13, the use of opacity makes the streaklets appear to fade in from the background in the direction of motion.

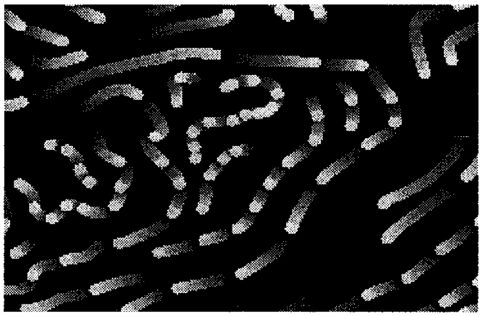


Figure 13: Streaklet opacity, used here to indicate direction.

3.3.2 Streaklet color and background color

Colors are controlled via the hue, saturation, and value (HSV) system. Hue is a value from 0 to 360 representing the entire hue spectrum. Saturation is a percentage value from 0 to 1, with 0 indicating no saturation (white) and 1 indicating full saturation (the selected hue). Value represents luminance and is also a percentage from 0 to 1, with 0 indicating no luminance (black) and 1 indicating full luminance (the selected hue/saturation value).

The HSV system was selected over the more common red, green, blue (RGB) system, as it is more intuitive for a user who is trying to generate a particular color. See *Section 3.6 – User interface* for more on color selection.

Recall that colors can be effective for nominal data if the data set is small, ordinal data if the colors fall across a recognizable color spectrum, or quantitative data if accompanied by a legend. Therefore, color could provide good visual cues for any of the data parameters. In figure 14, the left image shows the use of color to indicate direction, while the right image shows the use of color to indicate velocity.

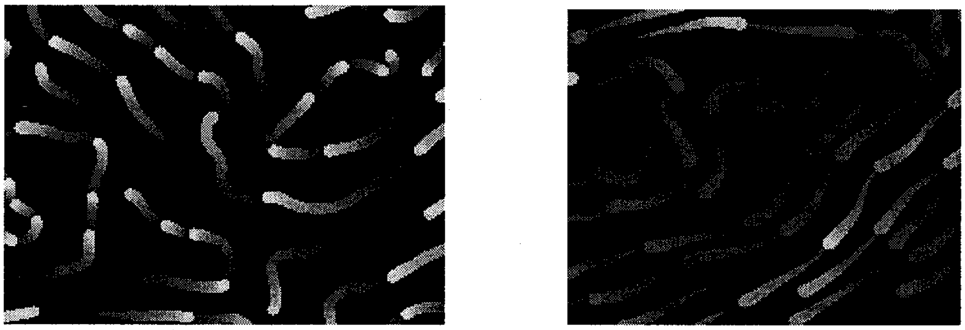
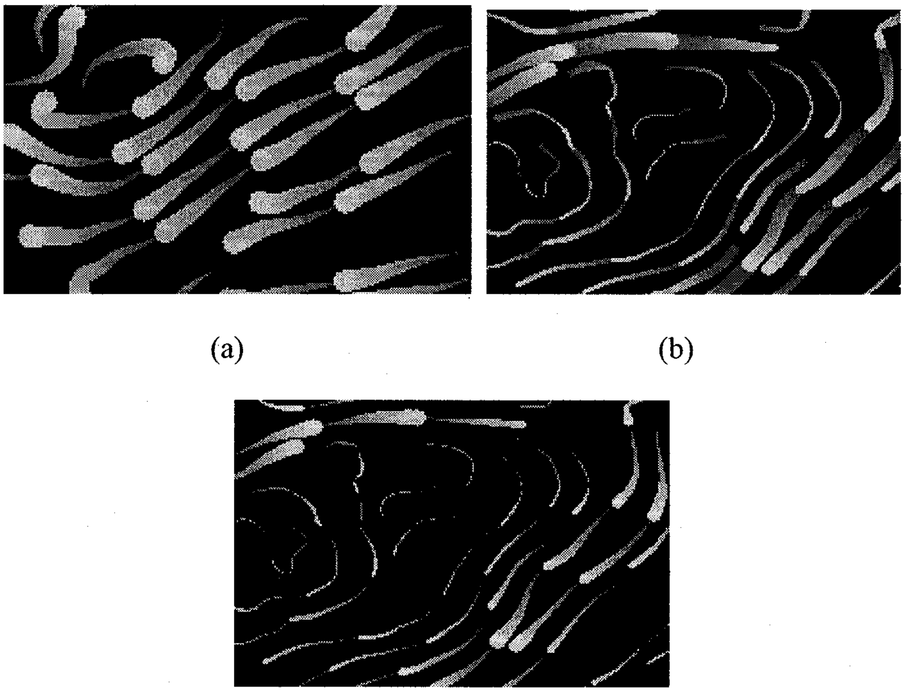


Figure 14: Streaklet color mapped to direction (left) and velocity (right).

3.3.3 Streaklet width and length

Streaklet width is measured in pixels. It can be mapped to velocity, to surface temperature, to direction, or to a hybrid direction/velocity combination. In the hybrid case, the maximum width at any point is controlled by the velocity at that point, but the width also cycles from 0 to that value as the streaklet progresses from tail to head. Figure 15 shows three different streaklet width mappings, direction (a), velocity (b), and the direction/velocity hybrid (c).



(a)

(b)

(c)

Figure 15: Streaklet width mapped to: a) direction; b) velocity; c) direction/velocity hybrid.

Streaklet length can either be constant, in which case all streaklets are the same length, or can vary based on velocity, in which case all streaklets have the same number of points (i.e. in areas of higher velocity, a streaklet's points will be more spread apart as a direct result of the iteration process). For constant length streaklets, the length is measured in pixels, while in the variable case, the actual lengths will be dependent on the flow data itself and a velocity multiplier (see *Section 3.6.2 – Control panel*). Figure 16 shows variable length (left) and constant length (right) streaklets.

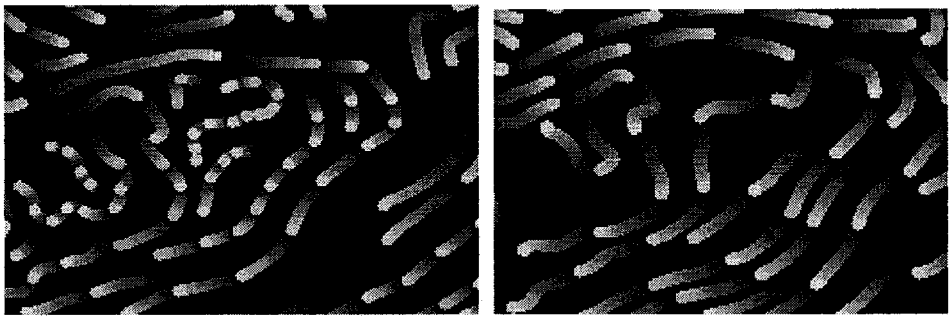


Figure 16: Streaklet length: variable (left) and fixed (right).

Recall that both length and width tend to be effective in representing quantitative data, and will likely be good indicators of velocity.

3.3.4 Streamline density

Streamline density controls how close streamlines are to each other. Higher density streamlines will be capable of showing more detail in the flow, but may obscure information represented by the background.

Streamline density can be set to vary based on velocity such that they may be closer together on average in areas of higher velocity and further apart in areas of lower velocity (or vice versa). This variability is enabled by a modification to the Jobard-Lefer algorithm for evenly-spaced streamlines, described in *Section 3.7.2 – Variable streamline density*.

Streamline density is *not* viable for representing quantitative data, and is used as a secondary indicator of velocity and for aesthetic purposes. Figure 17 shows dense and sparse streamline densities. Note how high density streamlines show more flow detail, while low density streamlines obscure less of the background colors.

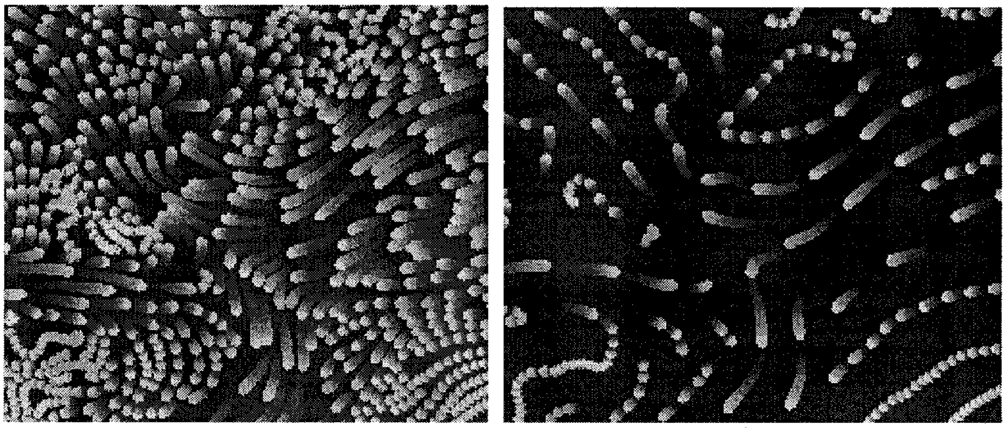


Figure 17: Streamline density

3.3.5 Streaklet circle head

The ability to add a circle to the head of a streaklet is *not* mapped to any data parameters, and is always available. It is provided for aesthetic purposes and to accentuate streaklet direction in certain cases. The circle head setting is relative to the maximum streaklet width, so wider streaklet heads will have proportionally larger circle diameters. Circle heads can be removed entirely by setting the size (diameter) to zero. Circle heads of varying sizes can be seen in figure 18.

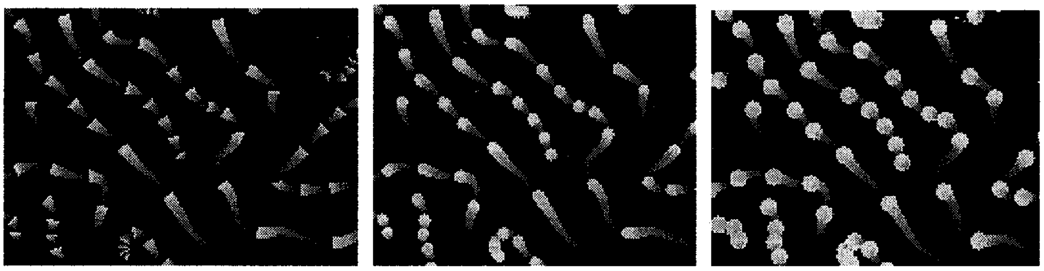


Figure 18: Circle heads, from non-existent (left) to large (right).

3.4 Data parameters

Data parameters represent the underlying information in the flow field. It is by effectively mapping these parameters to the visualization parameters above that we provide a visual representation of the flow. The data parameters considered for this study are orientation, direction, velocity (speed), and surface temperature.

3.4.1 Direction and orientation

As discussed in *Section 1.2.6 – Direction perception*, the orientation of the vector currents are represented by contour lines, however, disambiguating the direction of the contour lines is an important criterion for effective flow visualization. As noted, color and luminance changes, especially when perceived as the tail fading in from the background, and width, narrower to wider as we move from tail to head, have been shown to be excellent direction cues.

When mapping a visualization parameter to direction, the *minimum* value for that parameter indicates the value at the *tail* of the streaklet and the *maximum* value indicates the value at the *head* of the streaklet.

3.4.2 Velocity

One of the key aspects that a flow visualization must portray is the velocity of the flow at given points across the field. Viewers should be able to quickly discern areas of high versus low velocity and possibly estimate, with the aid of a key, the actual velocity at any given point. Since velocity is quantitative data, streaklet width, length, or color would be potentially effective cues.

When mapping a visualization parameter to velocity, the *minimum* value for that parameter indicates the value *where velocity is at its lowest* and the *maximum* value indicates the value *where velocity is at its highest.*

3.4.3 Surface temperature

Similar to velocity, surface temperature is a quantitative data field with similar requirements for discerning high versus low areas or specific temperatures at any point, and we would expect width, length, or color to be effective cues.

When mapping a visualization parameter to temperature, the *minimum* value for that parameter indicates the value *where temperature is at its lowest* and the *maximum* value indicates the value *where temperature is at its highest.*

3.5 Parameter mappings

Since the mapping of every possible visual parameter to every flow data parameter results in an unwieldy number of permutations, we need to make some educated decisions in an effort to reduce the combinations to a manageable number. As noted above, certain visualization parameters are known to be more effective at representing certain categories of data.

We start by locking background color to temperature. Since direction and velocity are intimately tied to the orientation of the streaklets, it makes little sense to use background color for either of these. Also, using background color to represent temperature is a common and intuitive treatment, especially displayed as *isobars* of temperature ranges rather than as continuously varying color.

We similarly lock opacity to direction. Fowler and Ware [FW89] found that the strongest direction cue was for a stroke to fade in from the background. Clearly, mapping opacity to direction provides this capability regardless of the other mappings, while mapping opacity to velocity would tend not to provide good quantitative value.

While any visualization parameter could theoretically be mapped to "constant", the only one for which we will specifically provide that mapping is streaklet length. This is primarily because the nature of the Jobard-Lefer algorithm requires different handling of constant streaklet lengths as opposed to streaklet lengths varying based on velocity. There is only one length setting, which controls either the actual length (in pixels) or the number of points in each streaklet, for constant length or variable length, respectively. Other visualization parameters can be made constant simply by setting the minimum and maximum values the same. This results in the following possible data mappings:

Background color: surface temperature (1 option)

Streaklet color: direction or velocity (2 options)

Streaklet opacity: direction (1 option)

Streaklet length: velocity or constant (2 options)

Streaklet width: direction, velocity, direction/velocity combination (3 options)

Separation: velocity (1 option)

The above set results in twelve possible mapping combinations, as follows:

	Color	**Opacity**	**Length**	**Width**	**Separation**	**Background Color**
1	Direction	Direction	Velocity	Direction	Velocity	Temperature
2	Direction	Direction	Velocity	Velocity	Velocity	Temperature
3	Direction	Direction	Velocity	Dir & Vel	Velocity	Temperature
*4	*Direction*	*Direction*	*Constant*	*Direction*	*Velocity*	*Temperature*
5	Direction	Direction	Constant	Velocity	Velocity	Temperature
6	Direction	Direction	Constant	Dir & Vel	Velocity	Temperature
7	Velocity	Direction	Velocity	Direction	Velocity	Temperature
8	Velocity	Direction	Velocity	Velocity	Velocity	Temperature
9	Velocity	Direction	Velocity	Dir & Vel	Velocity	Temperature
10	Velocity	Direction	Constant	Direction	Velocity	Temperature
11	Velocity	Direction	Constant	Velocity	Velocity	Temperature
12	Velocity	Direction	Constant	Dir & Vel	Velocity	Temperature

Table 1: Initial parameter mappings set

Finally, we remove mapping #4, as its only velocity cue is *separation*, which is not viable as a primary indicator. This leaves us with eleven mappings.

3.6 User interface

The following sections describe the user interface controls that allow the direct manipulation of parameters and mappings.

3.6.1 Sliders and interactive parameter adjustment

Slider controls allow the user to interactively control the visualization parameters for a given flow field through simple click-and-drag mouse actions.

Each visualization cue mapping is controlled by a set of radio buttons (in the control panel, see *Section 3.6.2 – Control panel*) and two slider controls (or sets of controls, in the case of color). The radio buttons allow the user to select which data value is to be mapped to the visualization cue. The sliders control the minimum (S_{min}) and maximum (S_{max}) values for each visualization parameter. These values are mapped to a data value at a given point by normalizing the mapped data value (V) at that point, such that V_{min} (the minimum value of V across the entire flow) maps to 0 and V_{max} (the maximum value of V across the entire flow) maps to 1. Call this normalized value V_{norm}. The desired visualization parameter value at that point is then:

$$S_{min} + (V_{norm} \cdot (S_{max} - S_{min}))$$

As an example, consider streaklet width mapped to velocity, where the minimum width slider (S_{min}) is set to 5 pixels, the maximum width slider (S_{max}) is set to 25 pixels, and the velocity ranges from 0 to 10 knots. For a point where velocity is, say, 8 knots ($V_{norm} = 0.8$), the streaklet width would be: $5 + (0.8 \cdot (25 - 5)) = 21$ pixels.

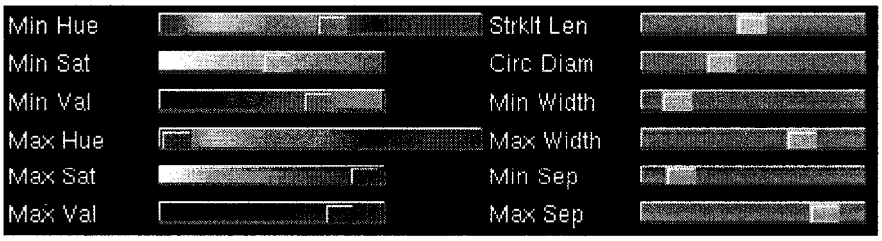
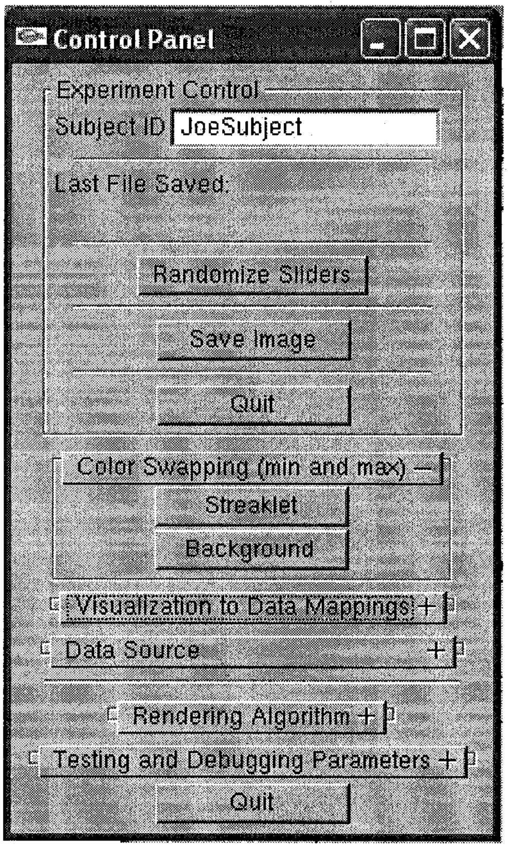


Figure 19: Slider controls

3.6.2 Control panel

The control panel is a separate window, coded using the glui library, that provides the tools needed to administer the experiment trials and save the data. It also provides an interface to fine tune the Jobard-Lefer rendering algorithm, to set the flow field data source, and to control debugging.



Subject ID JoeSubject

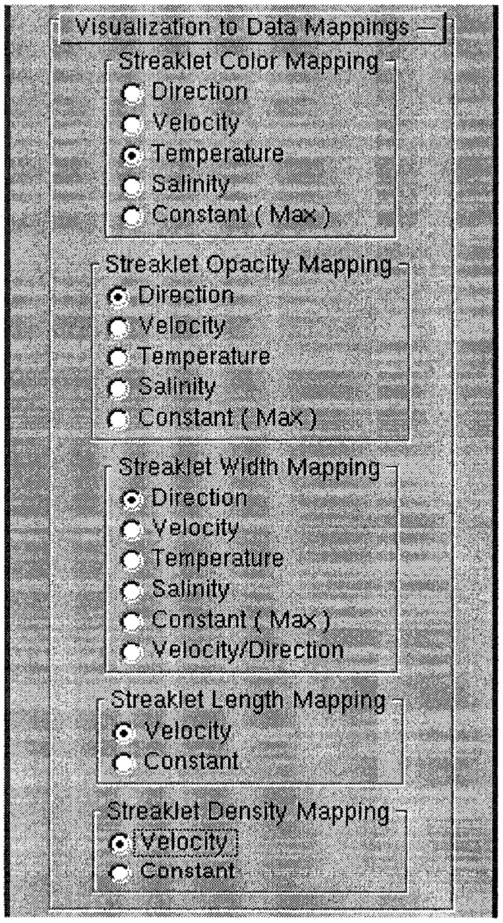
Figure 20: Control panel

The experimental control section allows the trial administrator to enter an ID name for each subject, to be used in file naming of the results data. It also includes a

“Randomize sliders” button to randomly set each slider value, and a “Save image” button to preserve the resulting image and settings for an optimized visualization.

Two buttons were also added to allow the user to interchange minimum and maximum color settings, since manipulating six color sliders to accomplish this manually was cumbersome.

The “Visualization to Data Mappings” section of the control panel allows the trial administrator to set each of the appropriate mappings during the course of a trial.



Visualization to Data Mappings

Streaklet Color Mapping
○ Direction
○ Velocity
● Temperature
○ Salinity
○ Constant (Max)

Streaklet Opacity Mapping
● Direction
○ Velocity
○ Temperature
○ Salinity
○ Constant (Max)

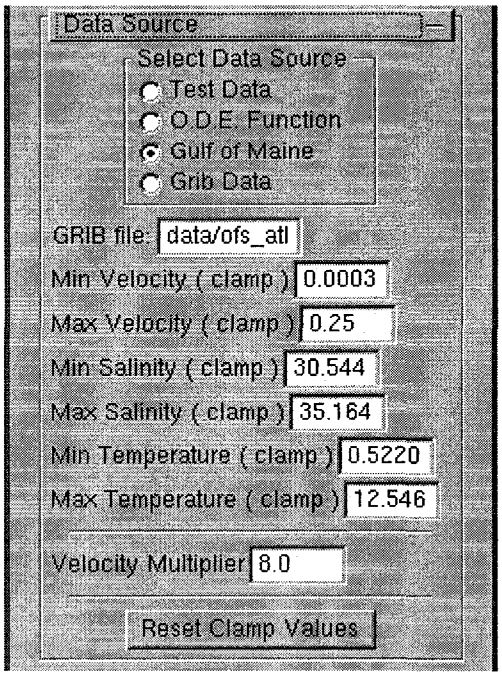
Streaklet Width Mapping
● Direction
○ Velocity
○ Temperature
○ Salinity
○ Constant (Max)
○ Velocity/Direction

Streaklet Length Mapping
● Velocity
○ Constant

Streaklet Density Mapping
● Velocity
○ Constant

Figure 21: Control panel - visualization mappings

The "Data Source" section of the control panel is where the data source of the flow field is specified. It also displays the minimum and maximum values of each data element in the loaded field, and allows for the override (clamping) of "fly-away" values that can skew the resulting visualization. The code is designed such that the data input classes can be easily extended to support different flow data formats.



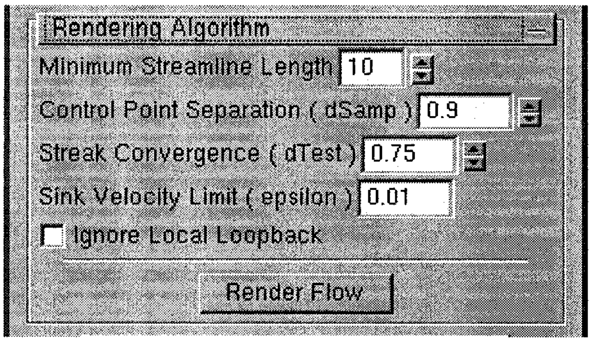
Select Data Source
○ Test Data
○ O.D.E. Function
● Gulf of Maine
○ Grib Data

GRIB file: data/ofs_atl
Min Velocity (clamp) 0.0003
Max Velocity (clamp) 0.25
Min Salinity (clamp) 30.544
Max Salinity (clamp) 35.164
Min Temperature (clamp) 0.5220
Max Temperature (clamp) 12.546

Velocity Multiplier 8.0

Figure 22: Control panel - data source

The "Rendering Algorithm" section of the control panel allows dynamic adjustment of the Jobard-Lefer rendering algorithm itself. The separation (d_{sep}) and convergence (d_{test}) values are discussed in *Section 1.5.1 – The Jobard-Lefer algorithm*. The sink velocity value provides a termination criteria during the iteration process as velocity approaches zero.



Rendering Algorithm

Minimum Streamline Length 10

Control Point Separation (dSamp) 0.9

Streak Convergence (dTest) 0.75

Sink Velocity Limit (epsilon) 0.01

□ Ignore Local Loopback

Render Flow

Figure 23: Control panel - rendering algorithm

3.7 Algorithm notes

3.7.1 Auxiliary changes to the Jobard-Lefer algorithm

A few issues arise during the implementation of the Jobard-Lefer algorithm for evenly-spaced streamlines. The first issue concerns the placing of sample control points along the iterated streamline. Jobard and Lefer state that "to make this approximation [testing against control points] acceptable, the control points on a streamline must be evenly spaced and the distance between them must be less than d_{sep}" [JL97]. If these control points are too far apart (i.e., d_{sep} is large), it is possible to miss certain areas of the flow when attempting to place new seed points due to control points spanning a significant bend in the flow. If we add new control points unnecessarily (i.e., for small values of d_{sep} where control points are already close together), we take a high performance penalty due to the number of comparisons required at each step of the streamline iteration. As a compromise, we set a graduated control point spacing *based on d_{sep}*. For larger spacing (lower density) renderings we set control points at 0.4 * d_{sep}; for lower spacing (high density) renderings we set control points at 0.9 * d_{sep}; in between we set at 0.6 * d_{sep}. This balance seems to generate good visualizations in reasonable time frames.

This solution, however, introduces another problem. When we place control points closer together, we complicate the determination of whether a potential point is valid or not. Specifically, we now may have several points along a streamline that are within the d_{sep} threshold of each other. Tracking which streamline each control point belongs to helps but does not completely solve the problem, since there are two situations when we *do* want control points on the current streamline to cause iteration to cease:

1) When a streamline hooks back on itself in its "local neighborhood". We define the local neighborhood of a potential point as all control points *sequentially back along the streamline* that are within d_{sep} of the point being tested. We call this "local loopback". In left-hand diagram in figure 24, point 2 should not interfere with the placement of point 4 since the control points are getting further apart along the streamline (distance(4:2) > distance(4:3). In the right-hand diagram, point 2 should disallow the placement of point 4, since it is closer to point 4 than point 3 is.
2) When a streamline loops or spirals and encounters itself again outside of its local neighborhood. We call this "global loopback". In figure 25, the green points should not interfere with the placement of the white point since these local neighborhood control points are getting further apart along the streamline. The red points, however, should

disallow the placement of the white point since we are looping back on ourselves outside the local neighborhood.

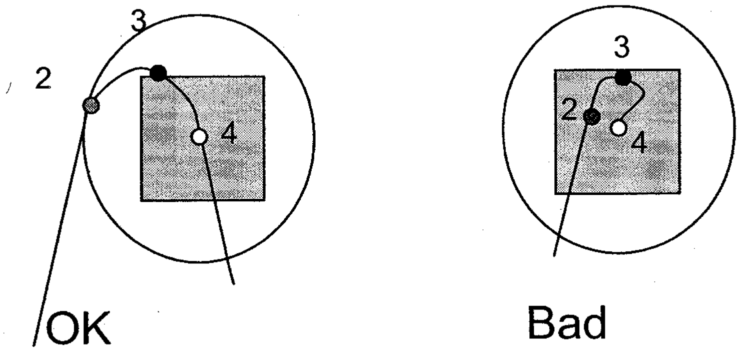


Figure 24: Local loopback

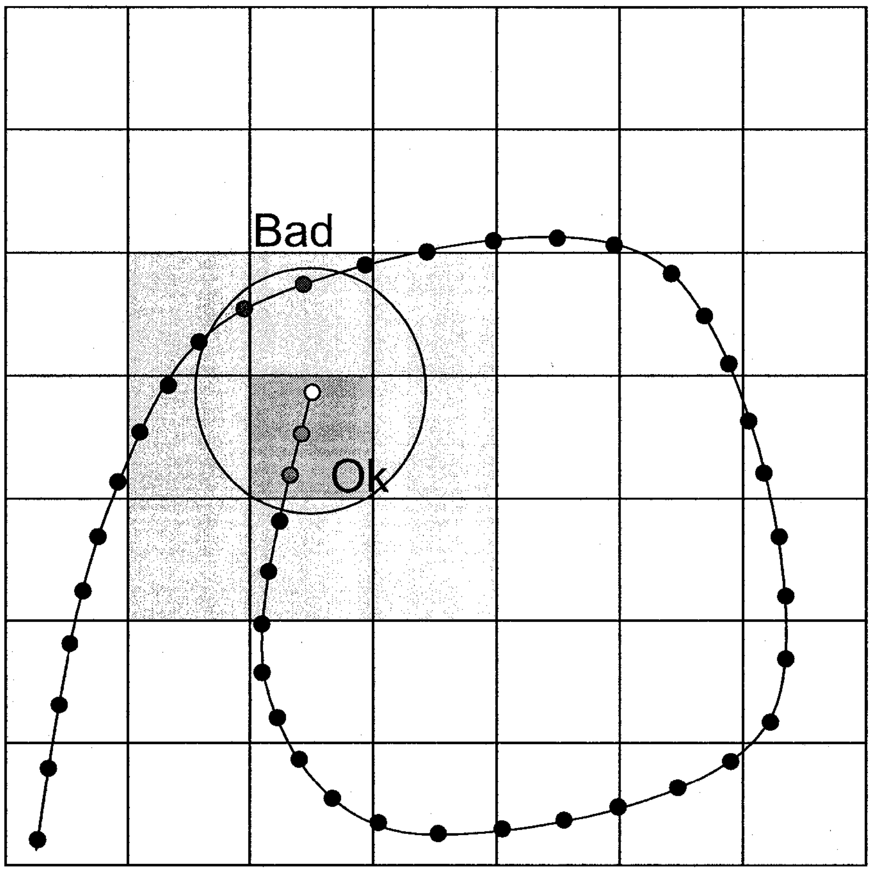


Figure 25: Global loopback

Both local and global loopback can be handled simultaneously by assigning sequential IDs to control points along each streamline. When testing a point, the control points in the local neighborhood are first tested in order (moving away from the target point). If any control point within the neighborhood is closer to the target than the previous one, then we have a local loopback condition and the validity test of the target point fails. If there is no local loopback, then we store the control point ID of the last point tested (i.e. the furthest point away that is still within the local neighborhood). Now, the global test can occur as normal, testing against all control points in the neighboring grid cells. If we find a point that is too close but belongs to the same streamline, we can now easily check its control point ID to determine if it is part of the local neighborhood (in which case it's okay, since we already checked for that condition); if it is not, then we have a global loopback condition and the validity test must fail.

3.7.2 Variable streamline density

We extend the Jobard-Lefer algorithm to allow the streamline density to vary based on local data values in the flow field. In this way, streamline spacing (density) can be used as another potential visualization cue, which can then be integrated into an interactive interface that allows users to control data-to-visualization-cue mappings and values along with streamline color and opacity, streamline width, streaklet or glyph length, and background color. This enhancement of the algorithm requires changes in the handling of several elements, as follows:

1) *Control point placement along the streamline.* As mentioned above, Jobard and Lefer specify that control points be evenly spaced along the streamline and at a distance smaller than the desired spacing (d_{sep}). In order to satisfy both requirements, control point separation is based on the *minimum separation* value specified.

2) *Overlay grid.* Jobard and Lefer specify that the overlay grid contain cells that are *exactly* the separation value (d_{sep}) apart. This limits the testing of control points to searching the current grid cell and eight surrounding cells. Rather than adding the complication of a multi-resolution grid, we set the grid based on the *minimum separation* value. It is then a simple calculation to determine how many grid cells

are required to encompass the grid cell neighborhood (see "Iteration control, below). Although this requires the testing of more than eight grid cells where separation is wider, wider separation also implies lower density so that many of the grid cells in that neighborhood will be sparse and the number of points tested will be similar.

3) *Seed point placement.* Potential seed points are placed based on the *local* separation value at each control point on an existing streamline.

4) *Separation error tolerance.* The Jobard-Lefer algorithm defines a tolerance, controlling how close a streamline can come to another streamline during generation, as a percentage of d_{sep}. We continue to allow this to be specified as a single percentage, applying it during iteration based on the local separation value at the point being tested.

5) *Iteration control.* Each separation test is based on the static value (e.g. velocity) at the target point. For the local loopback test, the local neighborhood is determined by the local separation value of the point being tested. The global test occurs as normal using the overlay grid (see 2 above), except that instead of a neighborhood of eight cells (the current cell plus or minus one cell), we need to calculate the number of cells (plus or minus "delta" cells) as follows:

delta = ceil ((local separation) / (grid cell size));

3.7.3 Artifact control and partial streaklets

Two major aesthetic issues arise when rendering streaklets along a streamline. The first is the appearance of undesirable artifacts due to streaklets on adjacent streamlines running side-by-side in a synchronous pattern. The second concerns the fact that the entire length of a streamline does not typically equate to an integer value of whole streaklets, resulting in a partial streaklet on most streamlines. These issues are controlled as follows:

Partial streaklets are disallowed. Since streaklet length is one of our mapping parameters, is it important that each streaklet length represent a true value. One disadvantage of eliminating partial streaklets is that there is more chance of visual gaps in portions of the resulting flow, especially for settings where streaklet lengths are long.

Streamlines below a configurable threshold length are discarded in the streamline generation algorithm. This prevents the creation of streamlines that are less than a full streaklet in length.

For each streamline, the number of whole streaklets that will fit along its length is calculated, along with a remainder value (i.e., the length that will not be covered by whole streaklets). The starting point for rendering streaklets along each streamline is then a randomly-generated fraction of the remainder value. This effectively removes the artifacts that occur when streaklets are always rendered from the beginning of the streaklet.

CHAPTER 4

EVALUATION STUDY

4.1 Evaluation study goals

The goals of the evaluation study are to utilize and evaluate the developed parameter-control software in a human-in-the-loop optimization process for a moderately complex visualization design task; to produce a variety of optimized visual display solutions for showing ocean flow model output; and to analyze these solutions for common patterns in an attempt to determine some generalized truths regarding flow visualization in two-dimensions.

4.2 Configuration

The evaluation study was performed on a 20" (~51 cm) monitor running a resolution of 1280x1024 at 96 dpi. The actual flow field displayed in a square of 800 x 800 pixels, or approximately 21 x 21 cm.

4.3 Task

This evaluation study consists of creating two-dimensional flow visualization solutions using local hill climbing. Each prospective solution represents the same ocean flow dataset, for which we select the common representation of ocean currents combined with surface temperature, specifically empirical data of the average (de-tided) currents in the Gulf of Maine for the month of February, 2004. This type of visualization is often used in oceanography and meteorology, and the data is readily-available.

All flow visualization solutions consist of streaklets of potentially varying length, width, color, opacity, and separation (density) over a varicolored background. While the background colors always represent sea surface temperature, the streaklet parameters are used in various combinations to represent flow direction and velocity. We selected eleven of the streaklet-parameter-to-data mappings most likely to produce high-quality solutions (see *Section 3.5 – Parameter mappings*); each subject generates two solutions within each mapping. The eleven mappings are as follows:

	Color	**Opacity**	**Length**	**Width**	**Separation**	**Background Color**
1	Direction	Direction	Velocity	Direction	Velocity	Temperature
2	Direction	Direction	Velocity	Velocity	Velocity	Temperature
3	Direction	Direction	Velocity	Dir & Vel	Velocity	Temperature
4	Direction	Direction	Constant	Velocity	Velocity	Temperature
5	Direction	Direction	Constant	Dir & Vel	Velocity	Temperature
6	Velocity	Direction	Velocity	Direction	Velocity	Temperature
7	Velocity	Direction	Velocity	Velocity	Velocity	Temperature
8	Velocity	Direction	Velocity	Dir & Vel	Velocity	Temperature
9	Velocity	Direction	Constant	Direction	Velocity	Temperature
10	Velocity	Direction	Constant	Velocity	Velocity	Temperature
11	Velocity	Direction	Constant	Dir & Vel	Velocity	Temperature

Table 2: Summary of parameter mappings

To generate a solution, subjects select a starting visualization by repeatedly randomizing all visualization parameters until they find a reasonable starting point. They then modify the visualization parameters (width, length, colors, etc.) via interactive sliders to optimize the solution as much as possible within the confines of the experiment. Generated solutions are saved to disk for later evaluation. This process is done twice for each of the above mappings, for a total of twenty-two solutions per subject.

We then evaluate all solutions, grading them on a scale of 1 to 5. While it is theoretically possible to construct an objective test, whereby users determine velocity and temperature at given points and results are based on error rates, that is beyond the scope of this thesis. For our purposes, we will be content with subjective evaluations.

4.3.1 Introduction phase

Each subject is given an overview of the research and the task that will be performed. The relationship of mappings (data values to visualization parameters) is explained, and each slider is explained and demonstrated. The subject is given the criteria for a "good solution", one in which:

- It is easy to determine areas of fast and slow velocity.
- It is easy to determine areas of warm and cold surface temperature.
- Actual velocity and surface temperature can be estimated using the key.
- Sufficient detail exists to ascertain the flow pattern.

Several randomizations are shown, and subject is allowed to adjust the sliders to get a feel for how the visualization can be affected.

4.3.2 Solution generation phase

For each of the eleven mappings, in a predetermined pseudo-random order, the mapping is explained, especially in relation to the previous mapping just completed, so the subject is clear on what the parameters are.

For each of two trials within each mapping, the subject randomizes the slider parameters until a reasonable starting point is found. The subject is encouraged to change any "obvious" sliders to make immediate improvements. The subject then makes at least one pass through each slider control, adjusting as necessary in an attempt to improve the image. When the subject is satisfied, the solution is saved.

4.3.3 Rating solutions

Solutions are rated by the author and his advisor, an expert in flow visualization, on a scale of 1 (worst) to 5 (best). Solutions are rated in two separate passes, first in the pseudo-random order in which they were created and then grouped by mapping. The scores for each pass are then averaged together for a final score. In addition to the overall final score for each solution, separate scores for streaklets and background are given to allow for separate analysis on each.

4.4 Subjects

There are a total of eight subjects participating in the evaluation. Two are from the field of oceanography at the University of New Hampshire (UNH), two are professional designers from the Rhode Island School of Design (RISD), one is my thesis advisor, a UNH professor specializing in both oceanography and flow visualization, and three, including myself, are current or recent UNH graduate students from the Visualization Laboratory.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Overall results

An Analysis of Variance (ANOVA) indicates a significant main effect for subjects [$F(7,88) = 5.14$; $p < 0.001$], a significant effect for mappings [$F(10,88) = 2.95$; $p < 0.05$], and an interaction between the two [$F(70,88) = 1.92$; $p < 0.005$]. However, a Tukey "Honestly Significant Difference" (HSD) test on mappings shows three overlapping groups (see Table 3). This indicates that, statistically, 9 of the 11 mappings are equally good. This is not surprising given the large number of interactions relative to the sample size. While a larger domain, including more trials and more ratings, would likely provide a more statistically sound set of results, we will discuss the existing results from a perspective of which mappings produced the most solutions at or above different ratings levels.

	N	Subset		
MAPPING		1	2	3
1	16	2.4063		
9	16	2.5313	2.5313	
6	16	2.6188	2.6188	2.6188
4	16	2.6500	2.6500	2.6500
7	16	2.7781	2.7781	2.7781
10	16		2.7906	2.7906
5	16		2.8281	2.8281
2	16			2.9312
8	16			2.9469
11	16			2.9625
3	16			2.9688
Sig.		.054	.256	.089

Table 3: A Tukey HSD on mappings shows three overlapping groups.

A second ANOVA shows that there are significant differences between the different groups of subjects [F(2,143) = 7.41; $p < 0.005$], with designers producing the best results and meteorologists producing the worst.

5.2 Evaluating the mappings

As noted above, there is little statistical distinction among the mappings; most are capable of producing both good and bad solutions. Figure 26 summarizes the results of the mapping evaluations, provided below in Table 4, sorted by highest average rating. The graph indicates the minimum, maximum, and average rating for each mapping

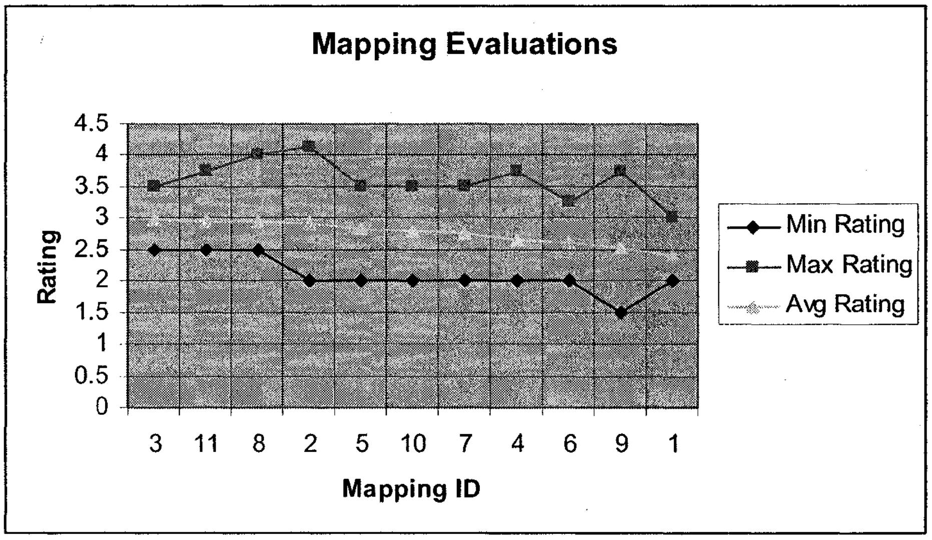


Figure 26: Mapping evaluations graph, sorted by average rating

Table 4 also includes the percentage of mappings that are:

- 2.75 or better (potential solutions – average-plus)
- 3.0 or better (good solutions)
- 3.5 or better (very good solutions)
- 4.0 or better (excellent solutions)

Notable values have been highlighted in yellow.

Mapping	Min Rating	Max Rating	Avg Rating	2.75+	3.0+	3.5+	4.0+	Total Solutions
3	2.5	3.50	2.97	81.3%	68.8%	12.5%	0.0%	16
11	2.5	3.75	2.96	81.3%	68.8%	6.3%	0.0%	16
8	2.5	4.00	2.95	62.5%	56.3%	12.5%	6.3%	16
2	2.0	4.15	2.93	68.8%	37.5%	18.8%	6.3%	16
5	2.0	3.50	2.83	62.5%	43.8%	12.5%	0.0%	16
10	2.0	3.50	2.79	62.5%	50.0%	6.3%	0.0%	16
7	2.0	3.50	2.78	68.8%	31.3%	6.3%	0.0%	16
4	2.0	3.75	2.65	37.5%	31.3%	6.3%	0.0%	16
6	2.0	3.25	2.62	31.3%	25.0%	0.0%	0.0%	16
9	1.5	3.75	2.53	31.3%	25.0%	6.3%	0.0%	16
1	2.0	3.00	2.41	6.3%	6.3%	0.0%	0.0%	16
Overall	**1.5**	**4.15**	**2.76**	**54.0%**	**40.3%**	**8.0%**	**1.1%**	**176**

Table 4: Mapping evaluations summary

For reference, the mapping parameter table (table 2) is repeated here, sorted as above by average rating.

	Color	Opacity	Length	Width	Separation	Background
3	Direction	Direction	Velocity	Dir & Vel	Velocity	Temperature
11	Velocity	Direction	Constant	Dir & Vel	Velocity	Temperature
8	Velocity	Direction	Velocity	Dir & Vel	Velocity	Temperature
2	Direction	Direction	Velocity	Velocity	Velocity	Temperature
5	Direction	Direction	Constant	Dir & Vel	Velocity	Temperature
10	Velocity	Direction	Constant	Velocity	Velocity	Temperature
7	Velocity	Direction	Velocity	Velocity	Velocity	Temperature
4	Direction	Direction	Constant	Velocity	Velocity	Temperature
6	Velocity	Direction	Velocity	Direction	Velocity	Temperature
9	Velocity	Direction	Constant	Direction	Velocity	Temperature
1	Direction	Direction	Velocity	Direction	Velocity	Temperature

Table 5: Summary of mappings (sorted by average rating).

Overall, only 54% of the generated solutions are rated as potential solutions, 40% are considered at least good, 8% are very good or better, and only two solutions (1.1%) are rated excellent.

Note also that the mappings that produce the highest average rating are not the same that produce the best solutions. Considering the limited size of this study, the first four mappings (3, 11, 8, and 2, marked in green) are very close in average rating. We consider these the first tier mappings, generating the best overall solutions. The second tier mappings (5, 10, and 7, marked in blue) also produce above average mappings, but are not at the same level as the first four. The third tier mappings (4, 6, 9, and 1, marked in red) produced generally poor results.

The most immediately salient result is with respect to the width parameter, with the mapping to direction producing the worst results on average, while the combination of Direction and Velocity produces four of the top five results on average. These mappings are further analyzed in the following sections, followed by a more detailed analysis of each mapping parameter.

5.3 First tier mappings

5.3.1 Mapping #3

	Color	Opacity	Length	Width	Separation	Background
3	Direction	Direction	Velocity	Dir & Vel	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
3	2.5	3.5	2.97	81.3%	68.8%	12.5%	0.0%	16

Table 6: Mapping #3 – excerpts from tables 4 and 5

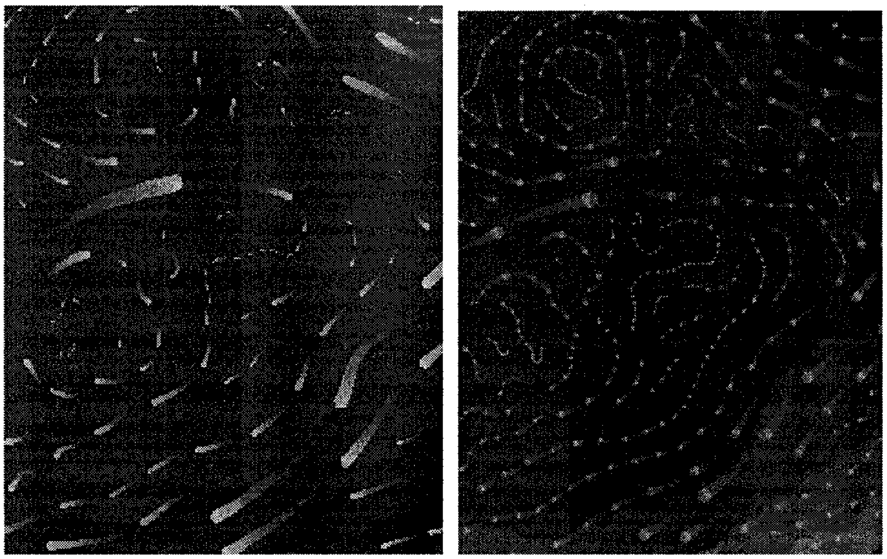


Figure 27: Excerpts from two top solutions for mapping #3. Both solutions had ratings of 3.5.

Mapping #3 has the highest overall average rating, with over 80% of the solutions rating average or better, and over two-thirds of the solutions at least "good" (3.0). The combination of length representing velocity with width representing a combination of direction and velocity produce a noticeable sweep of flow in areas of high velocity, while still representing an effective amount of detail in the lower velocity areas. Note, however, that the best solutions do not come from this mapping, and that only two solutions are rated "very good" (3.5+).

5.3.2 Mapping #11

	Color	Opacity	Length	Width	Separation	Background
11	Velocity	Direction	Constant	Dir & Vel	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
11	2.5	3.75	2.96	81.3%	68.8%	6.3%	0.0%	16

Table 7: Mapping #11 – excerpts from tables 4 and 5

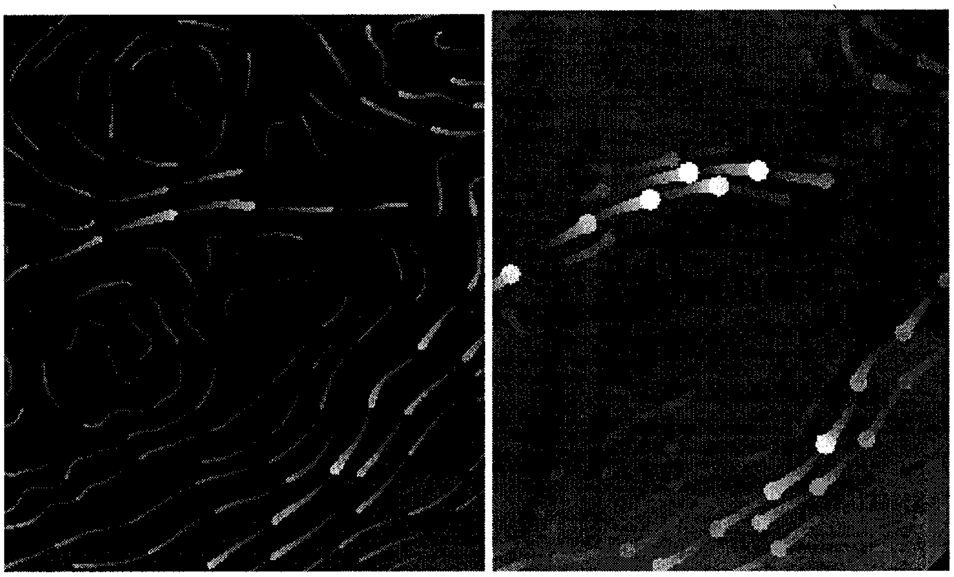


Figure 28: Excerpts from two top solutions for mapping #11. The first has a rating of 3.75, while the second has a rating of 3.25

Mapping #11 also produces a high percentage (81.3%) of good solutions, though with only one solution rated very good (3.5+). The combination of size and color to represent velocity effectively draws the eye to areas of higher velocity, though the constant length streaklets tend to lose detail in the lower velocity areas.

5.3.3 Mapping #8

	Color	Opacity	Length	Width	Separation	Background
8	Velocity	Direction	Velocity	Dir & Vel	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
8	2.5	4	2.95	62.5%	56.3%	12.5%	6.3%	16

Table 8: Mapping #8 – excerpts from tables 4 and 5

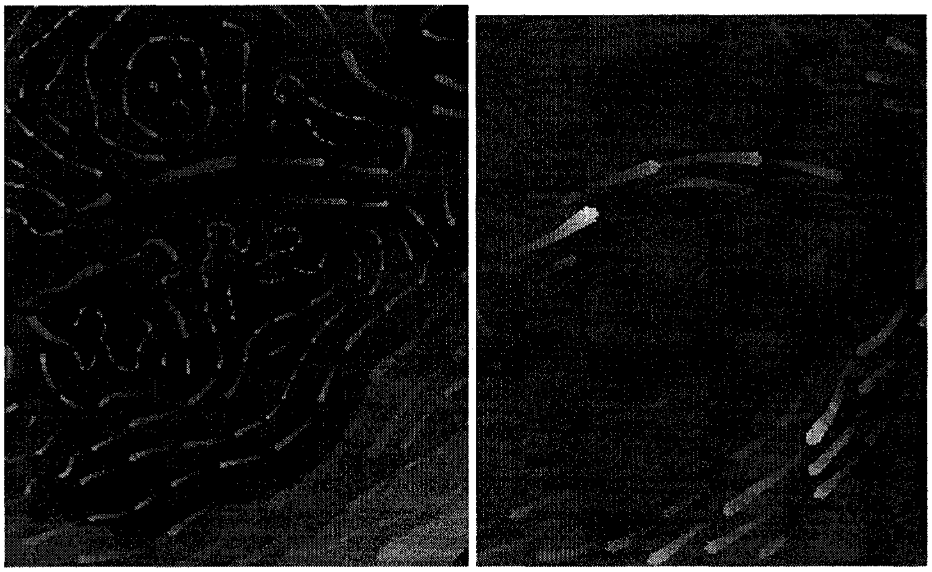


Figure 29: Excerpts from two top solutions for mapping #8. The first has a rating of 4.0, while the second has a rating of 3.75.

Mapping #8 has an average score only slightly less than the previous two mappings, above. However, it produced more very good and excellent solutions. The combination of color and length representing velocity, and width mapped to both velocity and direction, provides a strong visual flow with good detail. There is a slight drawback in that the ability to determine velocity at a particular point is diminished somewhat, since there is no longer a straight mapping to width. The only direct mapping to the velocity at a certain point is color, which, while relatively effective overall, has the potential to be misperceived due to its relationship to the background color at that point.

5.3.4 Mapping #2

	Color	Opacity	Length	Width	Separation	Background
2	Direction	Direction	Velocity	Velocity	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
2	2	4.15	2.93	68.8%	37.5%	18.8%	6.3%	16

Table 9: Mapping #2 – excerpts from tables 4 and 5

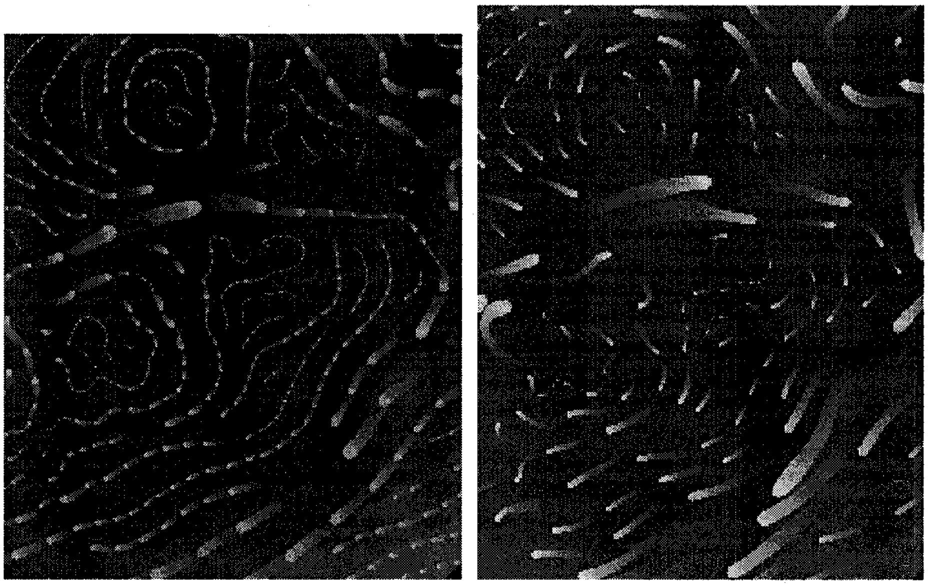


Figure 30: Excerpts from two top solutions for mapping #2. The first has a rating of 4.15, which was the highest rated solution of any mapping; the second has a rating of 3.75.

While mapping #2 had only the fourth best average rating, it was only slightly lower than the top three mappings and it had the highest percentage of very good or better solutions. Length and width appear to be an effective combination for representing velocity, as the wide, bold strokes draw attention to the areas where the current is fastest, while the detail in the slower areas is not lost. Color and opacity also combine well for indicating direction and, while it might seem that opacity would do as well alone, that mapping combination (#7) produced inferior results.

5.4 Second tier mappings

5.4.1 Mapping #5

	Color	Opacity	Length	Width	Separation	Background
5	Direction	Direction	Constant	Dir & Vel	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
5	2	3.5	2.83	62.5%	43.8%	12.5%	0.0%	16

Table 10: Mapping #5 – excerpts from tables 4 and 5

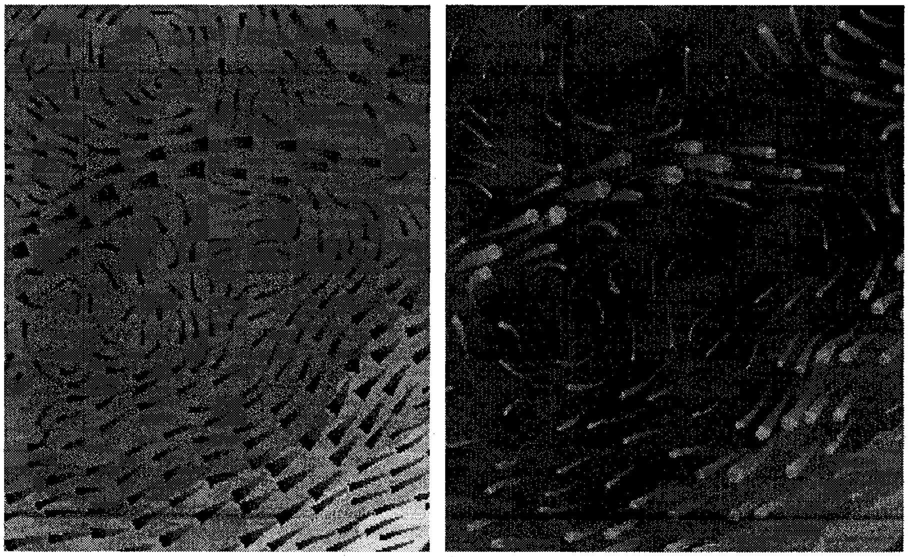


Figure 31: Excerpts from two top solutions for mapping #5. Both have ratings of 3.5.

This mapping suffers from an overabundance of direction cues, as width is the only velocity cue and it is representing direction as well. Since streaklets are constant length, they need to be relatively short and stubby to generate a adequate indication of high velocity areas. It is virtually impossible in this mapping to determine the velocity at a particular point in the flow.

5.4.2 Mapping #10

	Color	Opacity	Length	Width	Separation	Background
10	Velocity	Direction	Constant	Velocity	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
10	2	3.5	2.79	62.5%	50.0%	6.3%	0.0%	16

Table 11: Mapping #10 – excerpts from tables 4 and 5

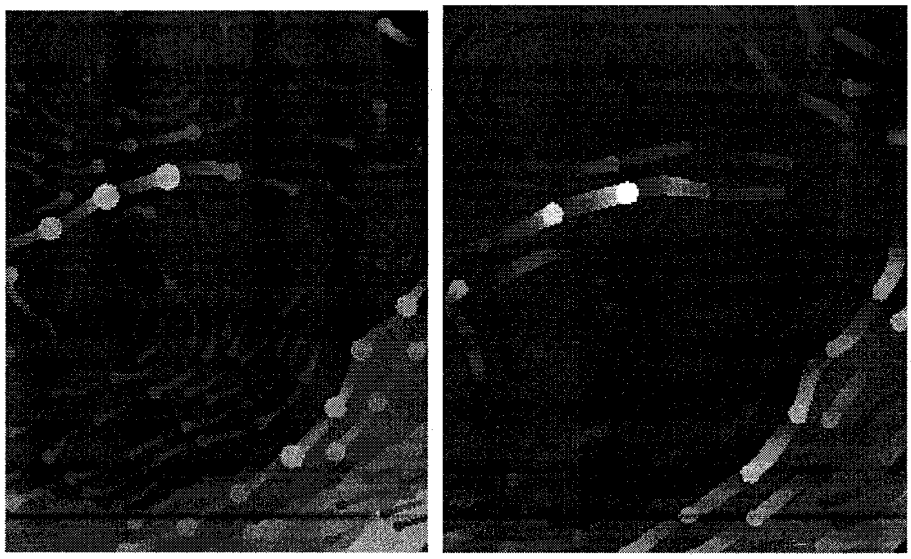


Figure 32: Excerpts from two top solutions for mapping #10. The first has a rating of 3.5, while the second has a rating of 3.25.

This mapping has the opposite issue of the previous one (#5). Almost all of the mappings represent velocity, with only opacity left to indicate direction. The resulting solutions did not evoke a strong sense of flow. Also, while it is usually effective to maximize the difference between the minimum and maximum widths, for greater separation of values, setting the minimum width a bit wider for this mapping enhances the perception of opacity change along the streaklets in areas of low velocity.

5.4.3 Mapping #7

	Color	Opacity	Length	Width	Separation	Background
7	Velocity	Direction	Velocity	Velocity	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
7	2	3.5	2.78	68.8%	31.3%	6.3%	0.0%	16

Table 12: Mapping #7 – excerpts from tables 4 and 5

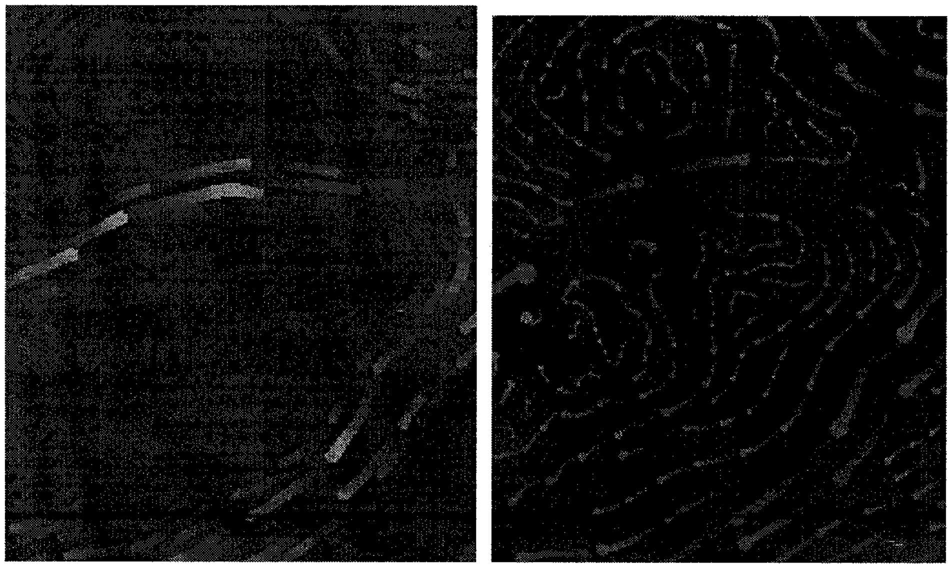


Figure 33: Excerpts from two top solutions for mapping #7. The first has a rating of 3.5, while the second has a rating of 3.25.

This mapping is identical to the previous one (#10), except that streaklet length varies based on velocity, instead of remaining a constant length. In general, variable length mappings are superior to constant length mappings (see *Section 5.6.2 – Streaklet length*). In this case, the results are very similar, with average scores almost identical.

We expected this to be a top combination, with opacity sufficient to indicate direction, width providing a solid velocity measure at each given point, and the combination of width, length, and color providing very strong visual cues to areas of high and low velocity. This expectation is not supported by the experiment, as it appears that opacity alone is not as strong a directional cue as when combined with width or color. The areas of high and low velocity are certainly apparent, but the feeling of flow is less pronounced that in the higher-rated mappings.

5.5 Third tier mappings

5.5.1 Mapping #4

	Color	**Opacity**	**Length**	**Width**	**Separation**	**Background**
4	Direction	Direction	Constant	Velocity	Velocity	Temperature

Mapping	**Min**	**Max**	**Avg**	**2.75+**	**3.0+**	**3.5+**	**4.0+**	**Total**
4	2	3.75	2.65	37.5%	31.3%	6.3%	0.0%	16

Table 13: Mapping #4 – excerpts from tables 4 and 5

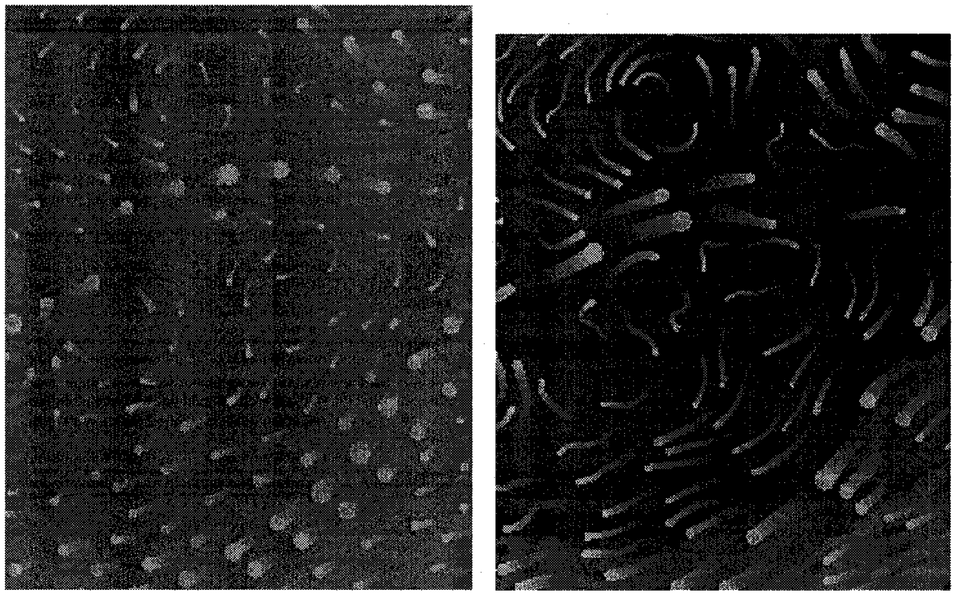


Figure 34: Excerpts from two top solutions for mapping #4. The first has a rating of 3.75, while the second has a rating of 3.25.

In this mapping, width is the only velocity cue, color and opacity are mapped to direction, and length is constant. Note how much worse the solutions are with constant length streaklets, compared to mapping #2, which is the same except with streaklets varying based on velocity. Here, only 37.5% of solutions are rated as average or better. Shorter length streaklets tend to accentuate the sense of flow, which can be seen above in the only "very good" (3.75) solution for this mapping.

5.5.2 Mapping #6

	Color	Opacity	Length	Width	Separation	Background
6	Velocity	Direction	Velocity	Direction	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
6	2	3.25	2.62	31.3%	25.0%	0.0%	0.0%	16

Table 14: Mapping #6 – excerpts from tables 4 and 5

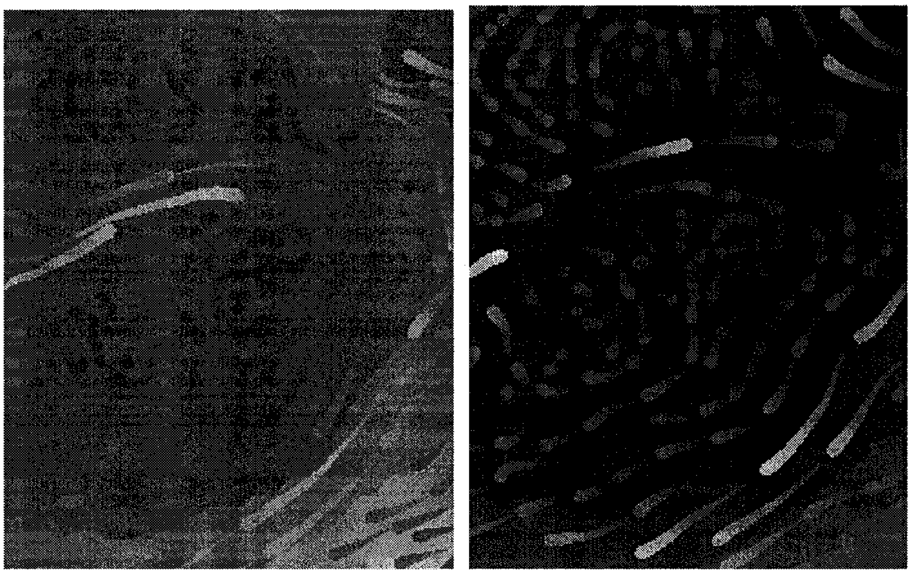


Figure 35: Excerpts from two top solutions for mapping #6. Both have ratings of 3.25

This mapping is the first of the three mappings of width to direction, which together represent the worst of all mappings tested. Length and color alone do not provide nearly as strong a visual cue to velocity, as evidenced by the less than one-third of the solutions rating average or better.

5.5.3 Mapping #9

	Color	Opacity	Length	Width	Separation	Background
9	Velocity	Direction	Constant	Direction	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4.0+	Total
9	1.5	3.75	2.53	31.3%	25.0%	6.3%	0.0%	16

Table 15: Mapping #9 – excerpts from tables 4 and 5

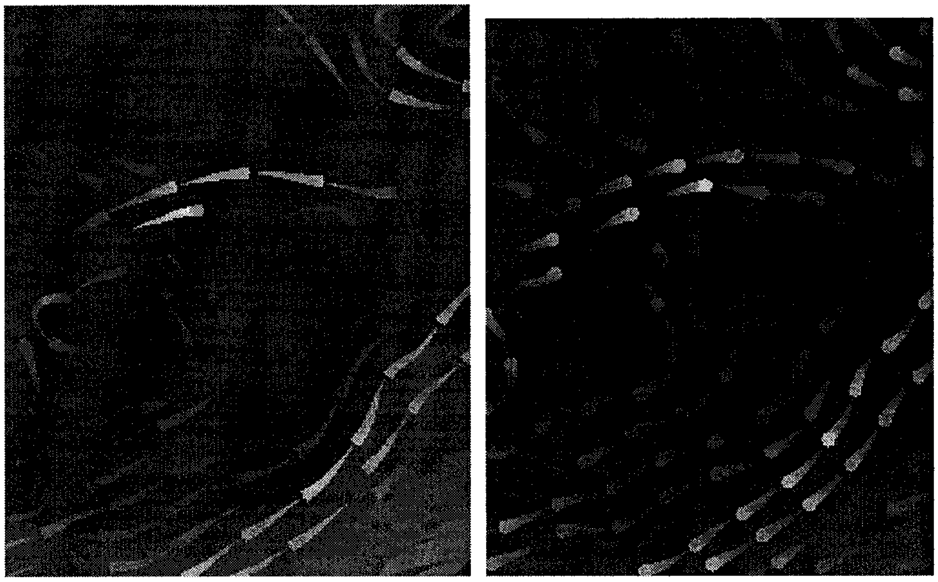


Figure 36: Excerpts from two top solutions for mapping #9. The first has a rating of 3.75, while the second has rating of 3.25.

This is essentially the same as the previous mapping (#6), except with constant streaklet lengths, losing yet another velocity cue. The dependence on color alone to represent velocity makes generating good solutions difficult (one subject referred to this particular mapping as "crippling"). We would expect that people would intuitively be more likely to associate warmer colors (reds, yellows, high-luminance values) with faster velocities and cooler colors (greens, blues, low-luminance values) with slower velocities. Additionally, we would expect higher velocities to have a higher contrast against the background, making them more prominent. These two factors together would indicate that the better solutions will likely have dark (low-luminance) background colors, which we can see in the above examples.

5.5.4 Mapping #1

	Color	Opacity	Length	Width	Separation	Background
1	Direction	Direction	Velocity	Direction	Velocity	Temperature

Mapping	Min	Max	Avg	2.75+	3.0+	3.5+	4+	Total
1	2	3	2.41	6.3%	6.3%	0.0%	0	16

Table 16: Mapping #1 – excerpts from tables 4 and 5

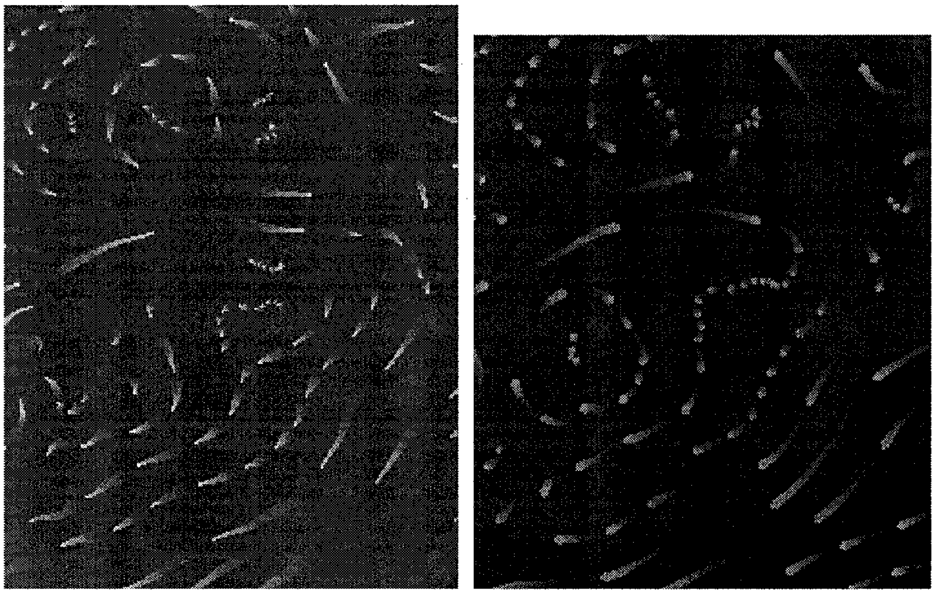


Figure 37: Excerpts from two example solutions for mapping #1. The first was the only solution in this mapping that rated average or better (3.0). The second is typical of the many below average solutions (2.5).

In this last mapping, length is the only velocity cue. Color, opacity, and width are mapped to direction. This is by far the worst mapping in our study, with only one solution (6.3%) rated as average or better. This single good solution, seen in figure 37, above, is also the only solution to map width in a reverse direction (narrower at the head, wider at the tail), generating a sort of "wake" appearance behind moving particles.

While length as a velocity cue is sufficient to indicate areas of high and low velocity, it is less intuitive than mappings for which velocity is also represented by color and/or width. Also, it is impossible to determine velocity at a particular point, since the velocity may vary significantly over the length of a streaklet. Also, having a maximum width set too large tends to draw attention to areas of low velocity because of the apparent density of the streaklets there.

5.6 Analysis of streaklet parameters

Table 17 is a summary of averages and standard deviations for each length, width, and separation value, in centimeters. Values are further summarized across three ratings levels (2.5 and greater, 3.0 and greater, and 3.5 and greater) in an effort to see whether the values vary differently for better solutions; and across each length and width parameter mapping, since these values may have different significance in each mapping scenario. A similar table for each individual mapping follows (table 18).

	Min Length	Max Length	Min Width	Max Width	Width Diff.	Min Sep.	Max Sep.
Overall							
Avg	0.59	3.79	0.04	0.22	0.18	0.41	0.51
Stdev	0.71	2.60	0.03	0.05	0.06	0.18	0.20
2.5+							
Avg	0.56	3.79	0.03	0.22	0.18	0.41	0.50
Stdev	0.69	2.57	0.03	0.05	0.06	0.18	0.19
3.0+							
Avg	0.52	3.35	0.03	0.22	0.19	0.38	0.45
Stdev	0.58	2.53	0.03	0.05	0.05	0.16	0.16
3.5+							
Avg	0.37	3.19	0.03	0.22	0.20	0.31	0.42
Stdev	0.46	2.62	0.02	0.04	0.05	0.10	0.19
Width: Combo							
Avg	0.63	3.48	0.03	0.23	0.20	0.41	0.49
Stdev	0.72	2.44	0.02	0.05	0.05	0.18	0.19
Width: Dir							
Avg	0.42	4.81	0.03	0.18	0.15	0.44	0.48
Stdev	0.65	2.81	0.05	0.06	0.06	0.18	0.20
Width: Vel							
Avg	0.67	3.34	0.04	0.23	0.19	0.38	0.56
Stdev	0.73	2.41	0.03	0.03	0.05	0.17	0.20
Length: Const							
Avg	1.30	1.30	0.03	0.22	0.20	0.39	0.53
Stdev	0.44	0.44	0.03	0.05	0.06	0.19	0.19
Length: Vel							
Avg	0.00	5.87	0.04	0.21	0.17	0.42	0.50
Stdev	0.00	1.62	0.04	0.05	0.06	0.17	0.20

Table 17: Average values, in cm, across different parameter breakdowns

	Min Length	Max Length	Min Width	Max Width	Width Diff.	Min Sep.	Max Sep.
Mapping 1							
Avg	0.00	6.84	0.05	0.16	0.14	0.50	0.45
Stdev	0.00	1.26	0.07	0.07	0.07	0.18	0.19
Mapping 2							
Avg	0.00	5.33	0.04	0.24	0.19	0.42	0.52
Stdev	0.00	1.99	0.04	0.03	0.06	0.19	0.20
Mapping 3							
Avg	0.00	5.51	0.02	0.24	0.22	0.45	0.52
Stdev	0.00	1.39	0.02	0.03	0.04	0.16	0.18
Mapping 4							
Avg	1.34	1.34	0.03	0.25	0.23	0.41	0.55
Stdev	0.40	0.40	0.02	0.01	0.03	0.21	0.20
Mapping 5							
Avg	1.37	1.37	0.02	0.24	0.21	0.38	0.51
Stdev	0.46	0.46	0.02	0.04	0.05	0.21	0.22
Mapping 6							
Avg	0.00	6.33	0.03	0.20	0.17	0.40	0.48
Stdev	0.00	1.65	0.03	0.05	0.05	0.17	0.23
Mapping 7							
Avg	0.00	5.35	0.05	0.20	0.15	0.36	0.53
Stdev	0.00	1.74	0.02	0.03	0.04	0.12	0.22
Mapping 8							
Avg	0.00	5.88	0.05	0.21	0.16	0.42	0.50
Stdev	0.00	1.25	0.03	0.04	0.05	0.16	0.18
Mapping 9							
Avg	1.26	1.26	0.02	0.17	0.15	0.42	0.52
Stdev	0.46	0.46	0.03	0.04	0.06	0.17	0.16
Mapping 10							
Avg	1.35	1.35	0.04	0.24	0.19	0.34	0.63
Stdev	0.41	0.41	0.03	0.03	0.05	0.13	0.17
Mapping 11							
Avg	1.16	1.16	0.03	0.22	0.19	0.38	0.43
Stdev	0.47	0.47	0.02	0.06	0.06	0.21	0.18

Table 18: Average values, in cm, across different parameter mappings

5.6.1 Streaklet width

Width	Min Rating	Max Rating	Avg Rating	Pct 2.75+	Pct 3.0+	Pct 3.5+	Pct 4.0+	Total Solutions
Direction	1.5	3.75	2.52	22.9%	18.8%	2.1%	0.0%	48
Velocity	2	4.15	2.79	59.4%	37.5%	9.4%	1.6%	64
Combo	2	4	2.93	71.9%	59.4%	10.9%	1.6%	64

Table 19: Comparison of width mapping ratings

As we noted in *Section 5.2 – Evaluating the mappings*, the solutions that have width mapped to a combination of velocity and direction resulted in the highest average ratings, while the solutions that have width mapped to direction generated the lowest average ratings. This is further demonstrated by the following graph (figure 38).

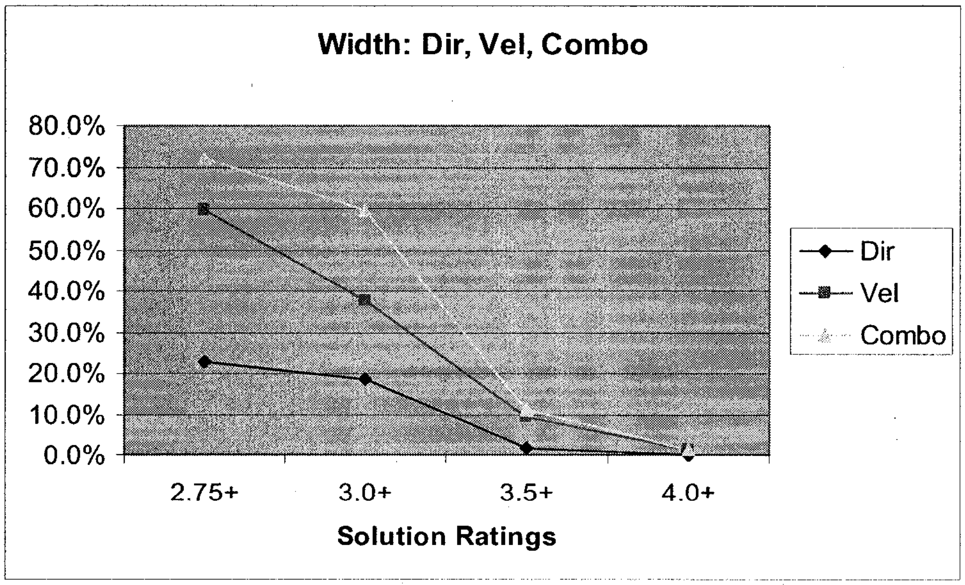


Figure 38: Comparative ratings for each width mapping

We expected solutions with higher disparity between minimum and maximum width to tend to be higher rated, but as can been seen in the following scatter plot (figure 39), this is not a strong relationship. Width value settings had a range of 0 to 10 pixels (0 to 0.27 cm), with an average width disparity of 0.18 cm and a standard deviation of 0.06 cm (see table 17). While these values are slightly larger for higher ratings (0.20 cm for 3.5+ ratings, 0.19 cm for 3.0+ ratings, 0.18 cm for 2.5+ ratings, all with similar standard deviations) the difference is slight.

Note that the cases where width is used to denote velocity either alone (0.19 cm width difference) or in combination with direction (0.20 cm width difference) have a larger width disparity than when width is used to denote direction alone (0.15 cm width difference). This is not unexpected, since velocity is a quantitative value. The binary direction value does not require as large a disparity to portray its meaning. Similarly, in cases where length is used as a velocity cue, there is less width disparity (0.17 cm) on average than when length is constant (0.20 cm), presumably because width needs to be a stronger visual velocity cue in the absence of a length cue. This hypothesis is further supported by the fact that the two mappings for which width is the sole velocity cue (no color or length mappings to velocity) are mapping four (0.23 cm average, 0.03 std dev) and mapping five (0.21 cm average, 0.05 std dev), both of which have among the highest width disparity of any of the mappings (table 18).

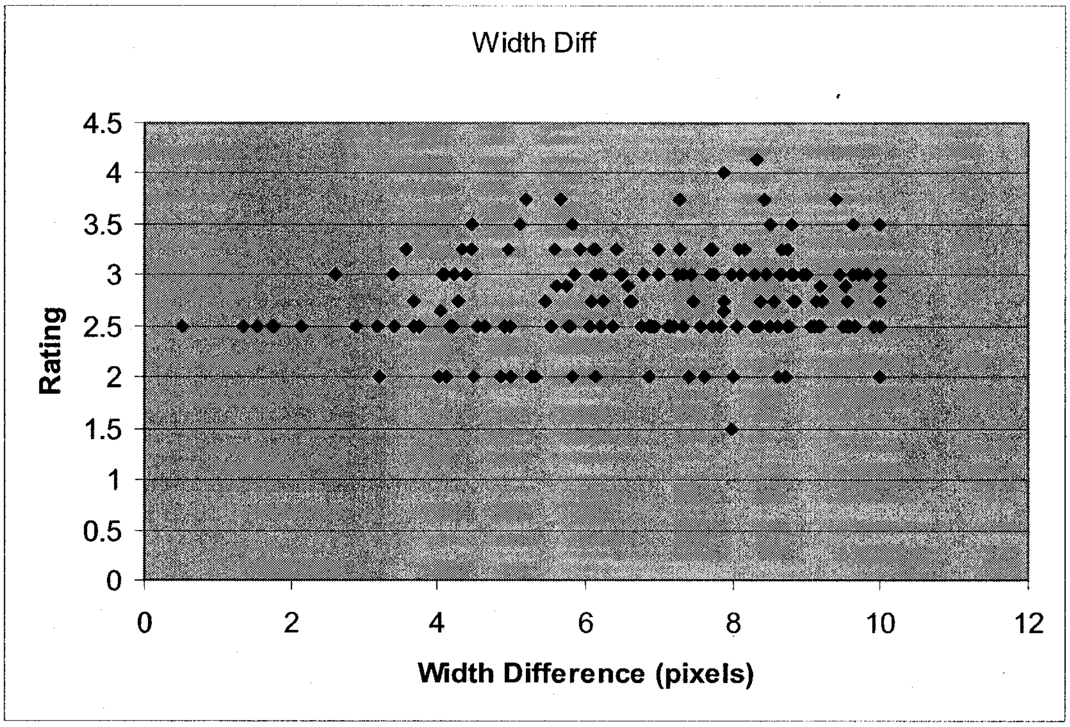


Figure 39: Ratings by width disparity

5.6.2 Streaklet length

Length	Min Rating	Max Rating	Avg Rating	Pct 2.75+	Pct 3.0+	Pct 3.5+	Pct 4.0+	Total Solutions
Constant	1.5	3.75	2.75	55.0%	43.8%	7.5%	0.0%	80
Velocity	2	4.15	2.78	62.5%	43.8%	10.0%	2.5%	80

Table 20: Comparison of constant- length vs. variable-length streaklets

Variable-length streaklets aid in the perception of velocity, downplay very slow currents, and tend to provide more detail in slower areas, but it may be difficult to determine direction in very slow areas, if the streaklets are too short. This may or may not be an issue, since direction of very slow currents is less likely to be relevant.

More than one subject expressed dislike for constant lengths, citing the difficulty of making areas of high velocity stand out. At first, the results do not seem to indicate a preference for variable length solutions. However, they were being skewed by the inclusion of the lowest-rated mapping #1 in the data for variable lengths. For this particular mapping, length is the *only* velocity cue; there is no corresponding mapping with constant length, as this would mean no velocity cue at all. Comparing the remaining ten mappings, we can definitely see a preference for varying the length of streaklets by velocity.

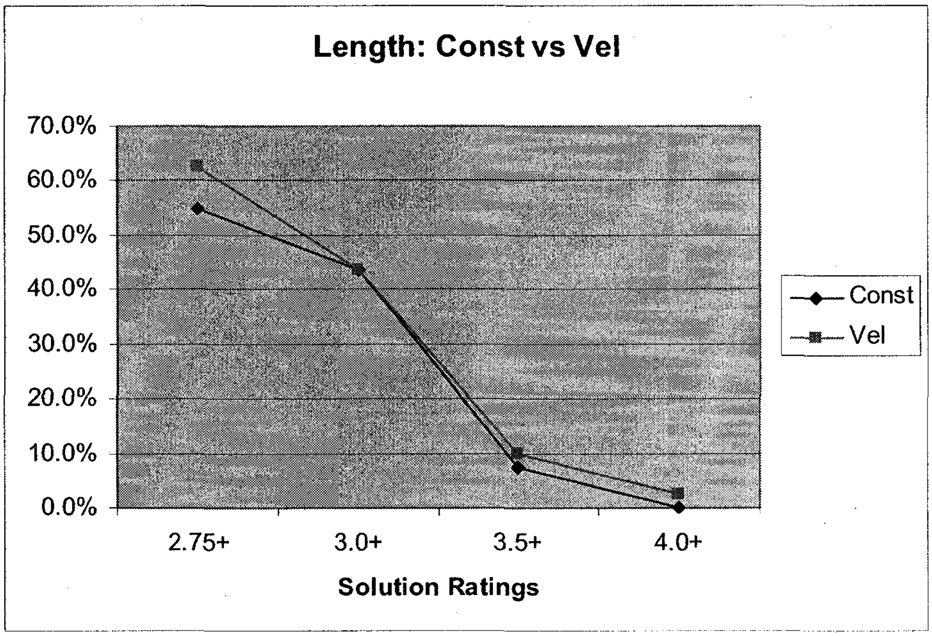


Figure 40: Percentage of solutions that are rated average (2.75) or better

Constant length solutions generally used relatively short streaklet-length settings, 1.30 cm average with a std dev of 0.44 cm, compared to variable length solutions whose maximum length averaged 5.87 cm with a std dev of 1.62 cm. In fact, the higher rated of the constant-length solutions tended to have shorter lengths, as represented by the dashed rectangle region in the scatter plot of figure 41 (note that 1 cm is approximately 38 pixels). Also, solution ratings tend to degrade with increasing streaklet length, as indicated by the trend line of the same figure.

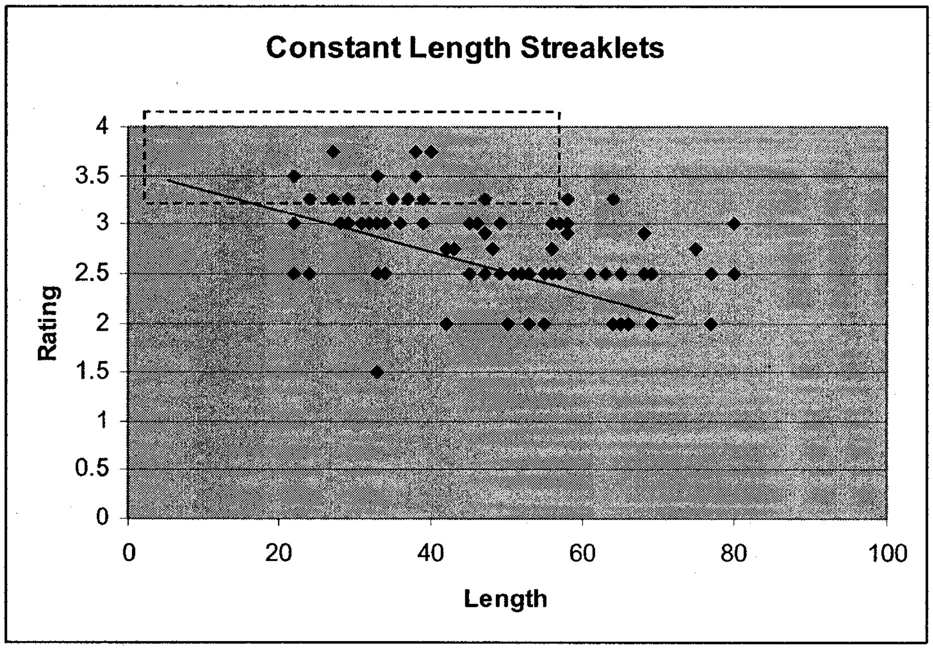


Figure 41: Scatter plot of constant length solutions (length in pixels).

Conversely, for variable length solutions, length does not appear to be a factor in the rating quality (figure 42). While more solutions tend to fall in the mid to high range of streaklet length, there is no demonstrable rating advantage to these solutions over those of lesser lengths.

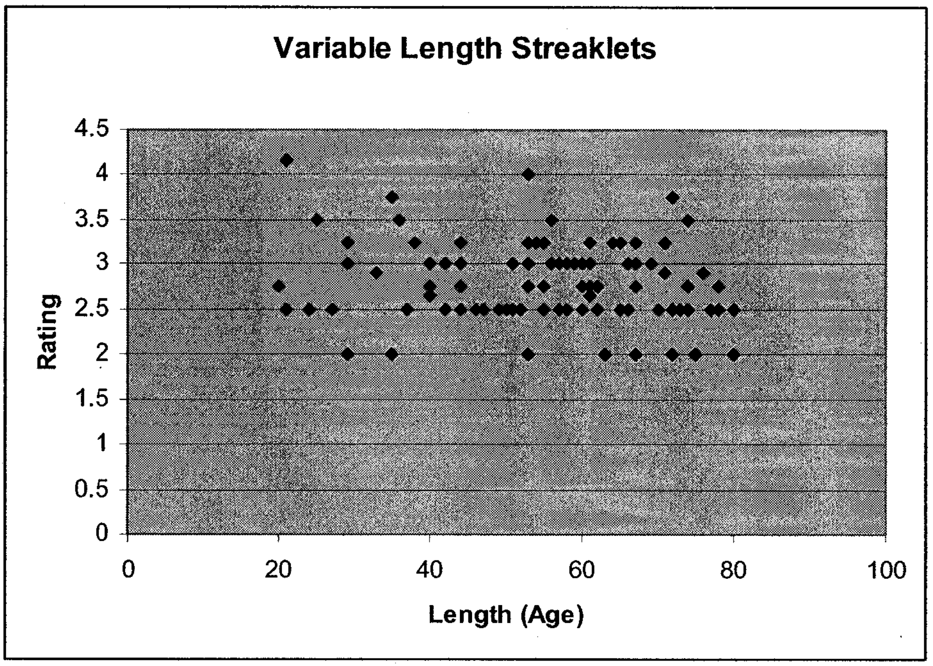


Figure 42: Scatter plot of variable length solutions

We conclude that varying streaklet length based on velocity is superior to constant streaklet length, but that length by itself does not draw attention to velocity differences as strongly as it does when combined with width and/or color. Also, it is difficult to identify velocity at a particular *point* along a streaklet since the streaklet length represents, in essence, an *average* velocity over its length.

5.6.3 Streaklet separation (density)

An analysis of streaklet separation does not provide much definitive insight into whether and how streaklet density affects the quality of visualizations. Roughly 69% of the solutions select values such that lower density streaklets (higher separation value) mapped to higher velocity areas, while 31% choose the opposite mapping. Each group mirrors the overall average ratings and standard deviations of the evaluation study (i.e., no significant rating difference one way or the other).

The average separation value is approximately 0.5 cm which, as noted in *Section 1.5.1 – The Jobard-Lefer algorithm*, is subject to a tolerance value in order to produce smooth continuous streamlines.

5.6.4 Color considerations

Color choices vary widely in this relatively small sample set, and significant patterns have not been detected. Color is also a very subjective parameter in that certain color combinations that are appealing to one may be distasteful to another. The fact that the flow visualization was of ocean currents certainly predisposed several subjects to select background colors in hues of blue. An effective analysis of color will require a larger sample set, more objective rating criteria, and a larger panel of judges. There are, however, some definite preferences explicitly noted by some of the subjects, as well as some considerations based on color perception in general.

Since color is perceived relative to background color, using color as a quantitative foreground against a variable-color background can be problematic. It is most effective when combined with another physical attribute (e.g., streaklet length or width) and when choosing colors with a wide disparity of luminance values (see examples from Mapping #11, figure 28). Using color for direction is not an issue, since the color is qualitative and perceived locally, relative to the rest of the streaklet.

Multiple subjects note that bright-colored backgrounds can cause strobing effects and is more distracting to the flow. Also, bright colors affect the way the foreground colors are perceived (see *Section 1.2.4, Lightness and chromatic contrast*). This suggests that low-luminance and grayscale backgrounds should produce better results, especially when mapping foreground color to a quantitative field (e.g., velocity). Also, if background colors vary widely in hue and/or luminance, it becomes more difficult to find foreground colors that will work effectively across the entire flow field.

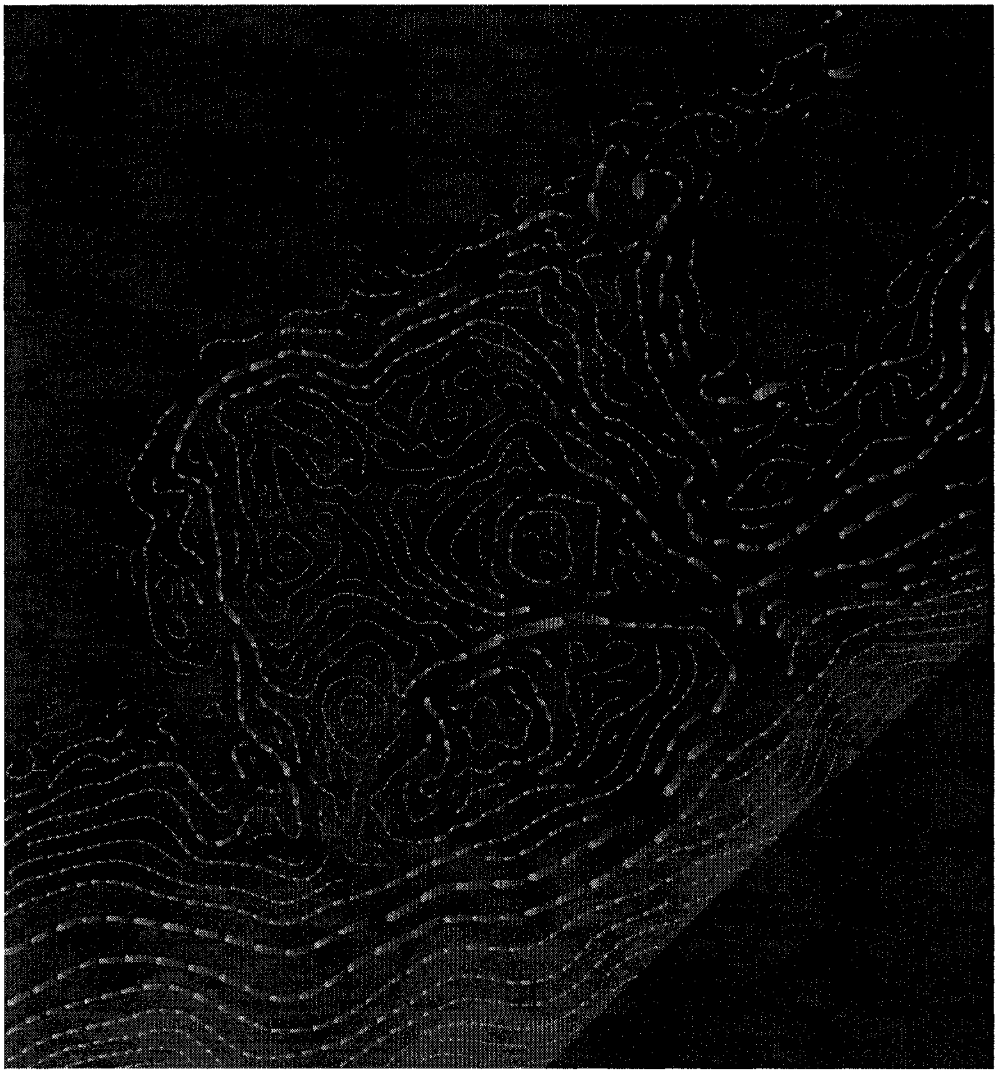


Figure 43: Highest rated solution, from mapping #2.

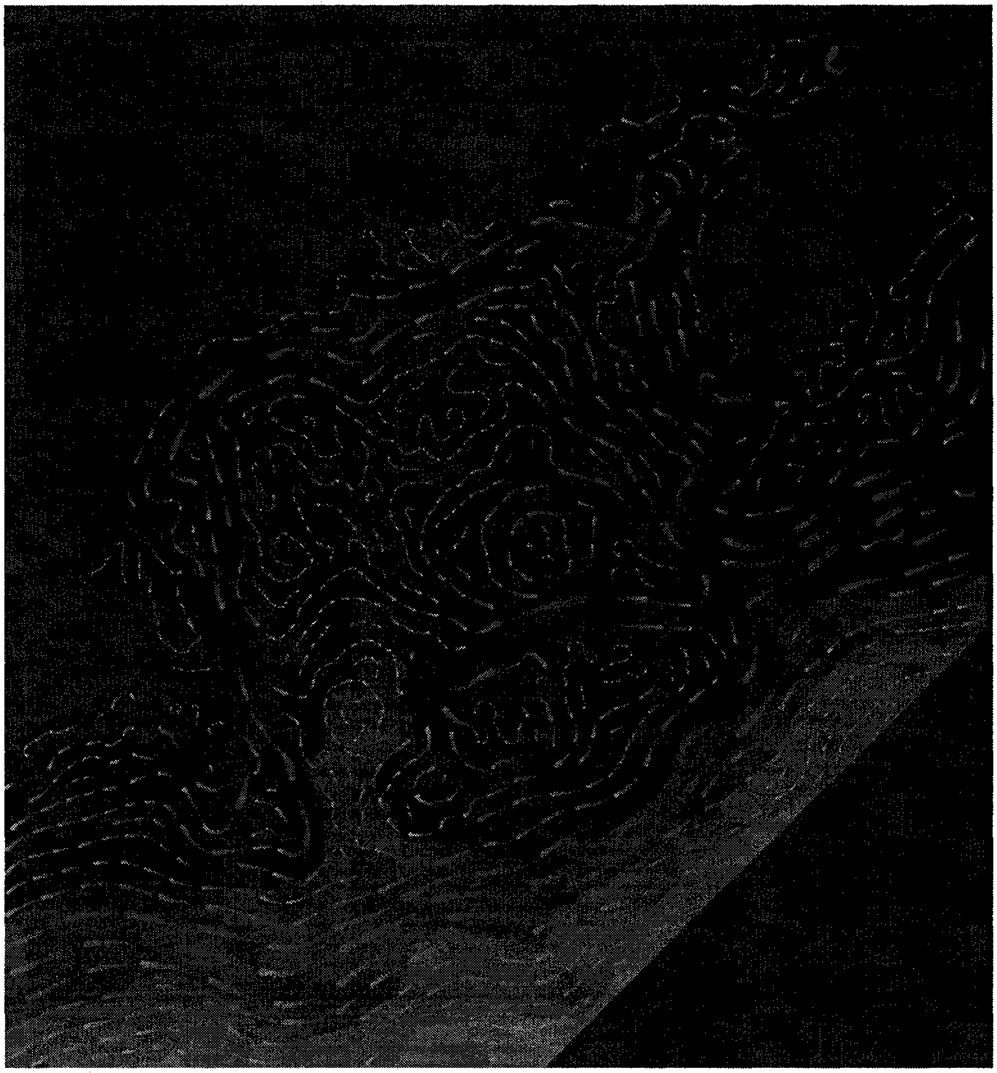


Figure 44: Second highest rated solution, from mapping #8.

CHAPTER 6

CONCLUSIONS

6.1 System software successfully supports human-in-the-loop methodology

The system software to allow human-in-the-loop interactive adjustment of visualization parameters using slider controls proves to be an effective tool to investigate flow visualization in two dimensions. Visualizations were typically rendered in under two seconds, allowing almost immediate feedback to changes made using the slider controls.

The algorithm to generate the streamlines is based on the Jobard-Lefer method, with a significant modification to support variable density control. Many of the algorithm's control parameters are adjustable in the program's control window, allowing fine-tuning of the streamline generation process based on the target data set or personal preferences. There are also several enhancements to handle flow idiosyncrasies such as global and local loopback.

The overlay of streaklets on each streamline requires special handling to ensure that only whole streaklets are rendered, since streaklet length is a significant quantitative parameter. Logic is also included to jitter the starting point of the head-to-tail streaklets on each streamline to reduce artifacts resulting from adjacent streaklets being in phase with each other.

The software's control window allows it to be run in several modes suitable for demonstrating the streamline generation algorithm, and is designed to be easily extended to support different input data formats.

The goal of leveraging the system software to conduct an evaluation study was accomplished. Eight subjects used the software in a controlled environment to generate twenty-two visualizations each, two for each of eleven parameter mapping combinations. Subjects who had never used or seen the tool were able to manipulate flow visualization parameters with relative ease within the first five to ten minutes of use, and a wide variety of visualizations were produced at an average rate of approximately one solution every two to three minutes. Other similar studies can be done with little or no modification to the existing system.

6.2 Human-in-the-loop local hill climbing is an effective methodology

While the software is successful in facilitating an evaluation study, it may seem less clear that the supported human-in-the-loop methodology is an effective method for producing quality flow visualization solutions. After all, while solutions were able to be generated interactively in relatively short time, just over the half (54%) of the solutions generated by the study are rated as "acceptable" or better, and very few are rated "very good" (8%) or "excellent" (1%). A number of factors may contribute to these results, including the small number (8) of subjects, and the smaller number (2) of evaluators, and the fact that almost all subjects had little to no exposure to the system or methodology prior to the trial.

More to the point, however, is that the people rating the solutions were doing so relative to the expected results. Compared to the visualizations currently being used today (see Appendix for examples), most of the solutions generated were far superior. In fact, a production version of the software is under development for use in NOAA's "NowCoast" website for disseminating ocean and weather data (figure 45).

It is likely that, with well-trained designers (who, as a group, generated the highest quality solutions), a much higher rate of quality solutions could be attained. And the speed with which solutions can be generated makes the human-in-the-loop, local hill-climbing method much preferable to long, drawn-out iterative approaches.

6.3 Some qualities of good solutions appear evident

While a larger evaluation study and more objective rating criteria would likely produce more definitive results, there are a few results worth noting. The best mapping for streaklet width appears to be to a combination of velocity and direction, with velocity alone being next best and direction alone being worst by a significant margin. Maximizing the disparity between minimum and maximum width tends to produce better results, and variable-length streaklets are preferred to constant-length streaklets, though constant-length streaklets do benefit somewhat from being relatively short.

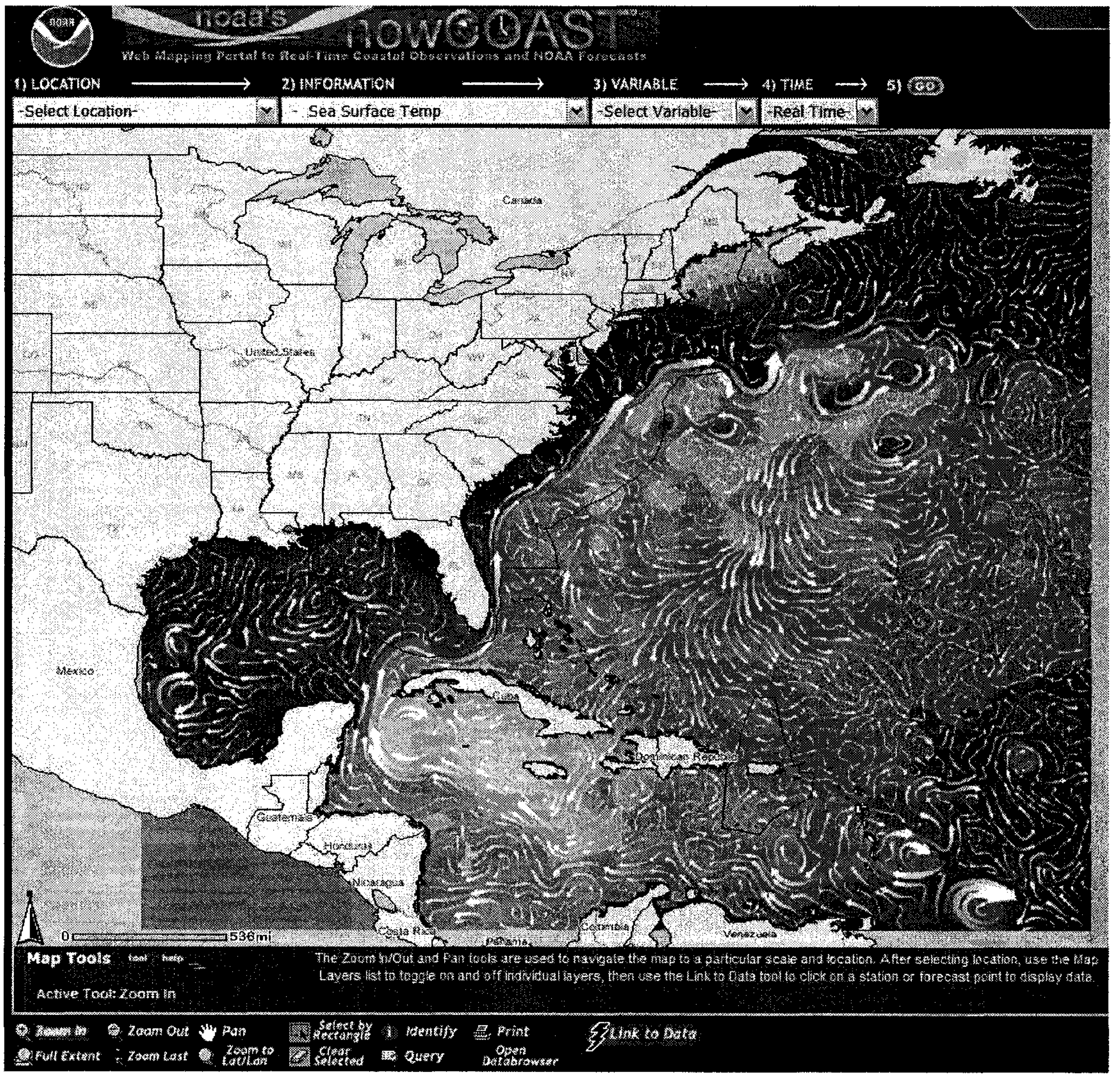
Streaklet spacing varied considerably, and did not appear to have a direct effect on the quality of the visualization.

6.4 Further studies are warranted

Further evaluation studies are warranted to generate a larger database of solutions and ratings, in an effort to generate more statistically-relevant results. Suggested tactics include:

- Utilize designers exclusively as subjects, since existing results indicate they create the best overall solution ratings.

- Perhaps pare down the number of mappings, focusing on the better mappings, to reduce the statistical uncertainty involved in multi-way comparisons.
- Use different data sources to test results against a variety of flow fields.
- Employ a wider variety of people to rate solutions, including people not involved in generating solutions.



noaa's nowCOAST

Web Mapping Portal to Real-Time Coastal Observations and NOAA Forecasts

1) LOCATION: -Select Location-

2) INFORMATION: - Sea Surface Temp

3) VARIABLE: -Select Variable-

4) TIME: -Real Time-

5) GO

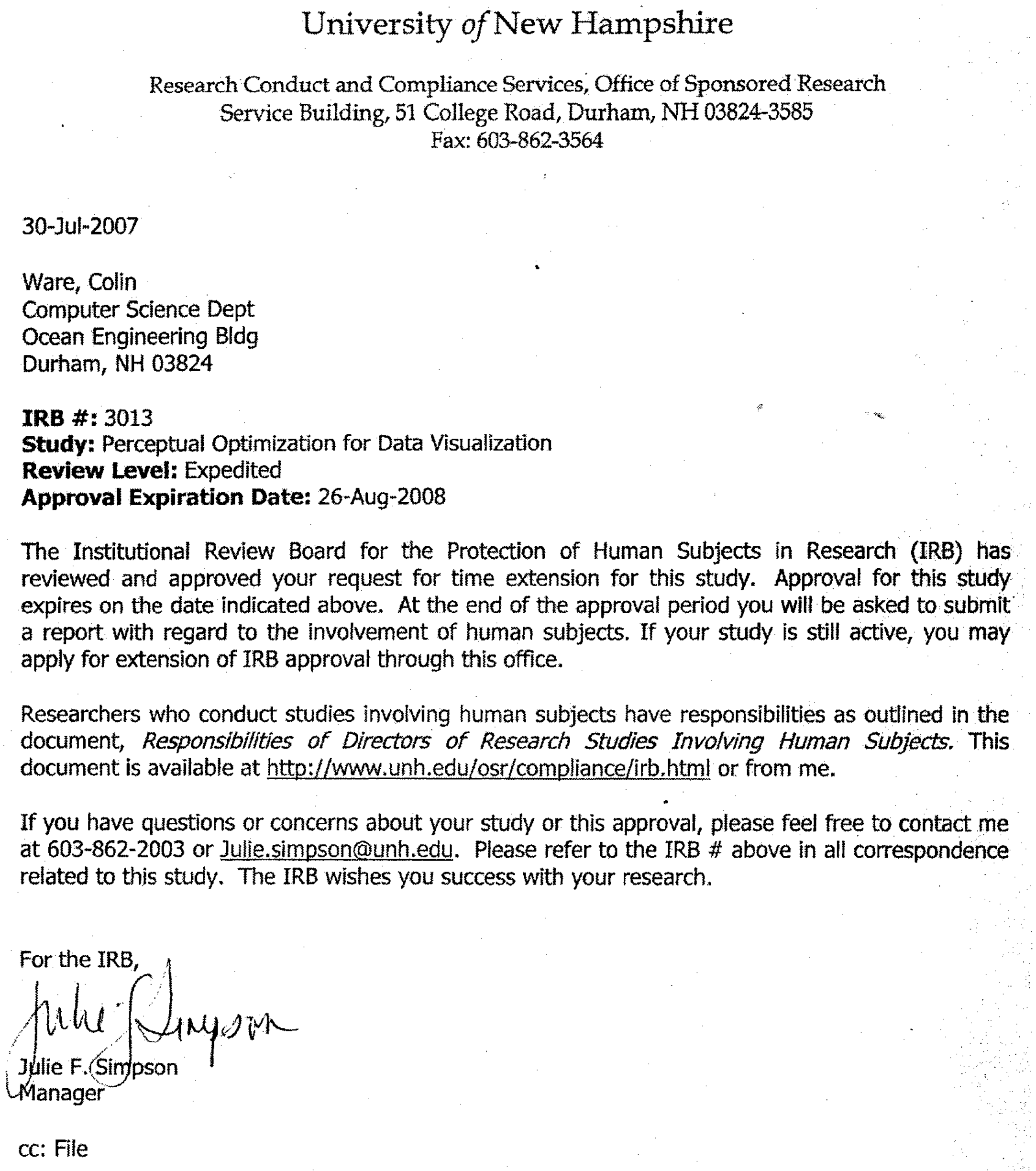
Active Tool: Zoom In

The Zoom In/Out and Pan tools are used to navigate the map to a particular scale and location. After selecting location, use the Map Layers list to toggle on and off individual layers, then use the Link to Data tool to click on a station or forecast point to display data.

Figure 45: Excerpt of a prototype solution for NOAA's "NowCoast" website.

APPENDIX

IRB APPROVAL LETTER



University *of* New Hampshire

Research Conduct and Compliance Services, Office of Sponsored Research
Service Building, 51 College Road, Durham, NH 03824-3585
Fax: 603-862-3564

30-Jul-2007

Ware, Colin
Computer Science Dept
Ocean Engineering Bldg
Durham, NH 03824

IRB #: 3013
Study: Perceptual Optimization for Data Visualization
Review Level: Expedited
Approval Expiration Date: 26-Aug-2008

The Institutional Review Board for the Protection of Human Subjects in Research (IRB) has reviewed and approved your request for time extension for this study. Approval for this study expires on the date indicated above. At the end of the approval period you will be asked to submit a report with regard to the involvement of human subjects. If your study is still active, you may apply for extension of IRB approval through this office.

Researchers who conduct studies involving human subjects have responsibilities as outlined in the document, *Responsibilities of Directors of Research Studies Involving Human Subjects.* This document is available at http://www.unh.edu/osr/compliance/irb.html or from me.

If you have questions or concerns about your study or this approval, please feel free to contact me at 603-862-2003 or Julie.simpson@unh.edu. Please refer to the IRB # above in all correspondence related to this study. The IRB wishes you success with your research.

For the IRB,

Julie Simpson

Julie F. Simpson
Manager

cc: File

LIST OF REFERENCES

[Ade95] Adelson, E. 1995. “Checker Shadow Illusion”. <http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html>. (May 3, 2006).

[BP96] Boring, E., and A. Pang. 1996. “Directional flow visualization of vector fields”. Proceedings of IEEE Visualization ’96. Pages 389–392.

[CL93] Cabral, B. and L. Leedom. 1993. “Imaging vector fields using line integral convolution”. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH ’93. Pages 263-272.

[CM92] Crawfis, R. and N. Max. 1992. “Direct volume visualization of three-dimensional vector fields”. *Proceedings of the 1992 Workshop on Volume Visualization.* VVS ’92. Pages 55-60.

[dv95] de Leeuw, W. C. and J. Van Wijk. 1995. “Enhanced spot noise for vector field visualization”. *Proceedings of the 6th Conference on Visualization ’95* IEEE Visualization. Pages 233-239.

[DW85] Dippé, M. A. and E. H. Wold. 1985. “Antialiasing through stochastic sampling”. *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH ’85. Pages 69-78.

[Dov95] Dovey, D. 1995. “Vector plots for irregular grids”. *Proceedings of the 6th Conference on Visualization ’95.* IEEE Visualization. Pages 248-253.

[FHH93] Field, D., A. Hayes, and R. Hess. 1993. “Contour integration by the human visual system: evidence for a local “association” field”. *Vision Research 33(2).* Pages 173-193.

[FW89] Fowler, D. and C. Ware. 1989. "Strokes for Representing Univariate Vector Field Maps". *Graphics Interface ’89.* Pages 249-253.

[HLD02a] Hauser, H., R. Laramee and H. Doleisch. 2002. “State-of-the-art report 2002 in flow visualization”. *TR-VRVis-2002-003*, Technical Report, VRVis Research Center, Vienna, Austria.

[HLD02b] Hauser, H., R. Laramee, H. Doleisch, F. Post, and B. Vrolijk. 2002. "The state of the art in flow visualization, part 1: direct, texture-based, and geometric techniques", *TR-VRVis-2002-046* , Technical Report, VRVis Research Center, Vienna, Austria.

[HH91] Helman, J., and L. Hesselink. 1991. "Visualizing vector field topology in fluid flows", *IEEE Computer Graphics and Applications*, vol. 11, no. 3. Pages 36-46.

[Her20] Hering, E. 1920. *Grundzuge der Lehr vom Lichtsinn.* Springer-Verlag, Berlin. (*Outlines of a Theory of Light Sense.* Translated by L.M. Hurvich and D. Jameson. Harvard University Press, Cambridge, MA 1964.)

[HZ00] Hertzmann, A. and D. Zorin. 2000. "Illustrating smooth surfaces". *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH '00. Pages 517-526.

[Hop96] Hoppe, H. "Progressive meshes". 1996. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH '96. Pages 99-108.

[JL97] Jobard, B., and W. Lefer. 1997. "Creating evenly-spaced streamlines of arbitrary density". *Eurographics Workshop on Visualization in Scientific Computing.* Pages 43-55.

[JL00] Jobard, B. and W. Lefer. 2000. "Unsteady flow visualization by animating evenly-spaced streamlines", *Computer Graphics Forum 19(3)* (*Proceedings of Eurographics 2000*).

[JL01] Jobard, B. and W. Lefer. 2001 "Multiresolution flow visualization". *WSCG'01, Plzen, Czech Republic.*

[KML99] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. 1999. "Visualizing multivalued data from 2D incompressible flows using concepts from painting". *Proceedings of IEEE Visualization '99.* Pages 333–340.

[KB96] Kiu, M. and D. Banks. 1996. "Multi-frequency noise for LIC". *Proceedings of the 7th Conference on Visualization '96.* Pages 121-126.

[Kof35] Koffka, K. 1935. *Principles of Gestalt Psychology.* Harcourt-Brace, New York.

[LDM01] Laidlaw, D. H., J. Davidson, T. Miller, M. da Silva, R. Kirby, W. Warren, and M. Tarr. 2001. "Quantitative comparative evaluation of 2D vector field visualization methods". *Proceedings of the Conference on Visualization '01.* VISUALIZATION. IEEE Computer Society. Pages 143-150.

[LHD04] Laramee, R., H. Hauser, H. Doleisch, B. Vrolijk, F. Post and D. Weiskopf. 2004. "The state of the art in flow visualization: dense and texture-based techniques". *Computer Graphics Forum*, Vol. 23, No. 2. Pages 203-221.

[Mac86] Mackinlay, J. 1986. "Automating the design of graphical presentations of relational information". *ACM Trans. Graph.* 5, 2 (Apr. 1986). Pages 110-141.

[MHH98] Mao, X., Y. Hatanaka, H. Higashida, and A. Imamiya. 1998. "Image-guided streamline placement on curvilinear grid surfaces". *Proceedings of IEEE Visualization '98.* Pages 135–142.

[MTH03] Mattausch, O., T. Theußl, H. Hauser and E. Gröller. 2003. "Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines". *Proceedings of the 19th Spring Conference on Computer Graphics.* Pages 213-222.

[MAD05] Mebarki, A., P. Alliez and O. Devillers. 2005. "Farthest point seeding for efficient placement of streamlines". *Proceedings of IEEE Visualization '05.*

[NOA07] National Oceanic and Atmospheric Administration (NOAA) 2007. "Experimental Galveston Bay/Houston Ship Channel Nowcasting/Forecasting System". <http://chartmaker.ncd.noaa.gov/csdl/op/hgops/images/galv/convec/convec.uv.1.0.png>. (September 26, 2007).

[PVH02] Post, F., B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. 2002. "Feature extraction and visualization of flow fields". *Eurographics 2002, State of the Art Reports.* Pages 69-100.

[PVH03] Post, F., B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. 2003. "The state of the art in flow visualization: feature extraction and tracking". *Computer Graphics Forum*, Vol. 22, No. 4. Pages 775-792.

[PPv95] Post, F. H., F. J. Post, T. van Walsum, and D. Silver. 1995. "Iconic techniques for feature visualization". *Proceedings of the 6th Conference on Visualization '95.* IEEE Visualization. Pages 288-295.

[Ree83] Reeves, W. 1983. "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects". *Computer Graphics* Vol. 17, No. 3. Pages 359-376.

[SHK97] Scheuermann, G., H. Hagen, H. Krger, M. Menzel, and A. Rockwood. 1997. "Visualization of higher order singularities in vector fields". *Proceedings of IEEE Visualization '97.* Pages 67-74.

[SH95] Stalling, D. and H-C Hege. 1995. "Fast and resolution independent line integral convolution". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH '95. Pages 249-256.

[Tv99] Telea, A. and J. van Wijk. 1999. “Simplified representation of vector fields”. *Proceedings of the Conference on Visualization '99: Celebrating Ten Years.* IEEE Visualization. Pages 35-42.

[TB96] Turk, G., and D. Banks. 1996. “Image-guided streamline placement”. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH ’96. Pages 453-460.

[UMi07] University of Miami Rosenstiel School of Marine & Atmospheric Science, Cooperative Institute for Marine and Atmospheric Studies (CIMAS), and National Oceanographic Partnership Program (NOPP) 2007. “Surface currents in the Indian Ocean”. <http://oceancurrents.rsmas.miami.edu/indian/img_mgsva/agulhas-yyy.gif>. (September 26, 2007).

[van91] van Wijk, J. 1991. "Spot noise -- texture synthesis for data visualization". *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH ’91. Pages 309-318.

[van02] van Wijk, J. 2002. “Image based flow visualization”. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH ’02. Pages 745-754.

[VKP99] Verma, V., D. Kao and A. Pang. 1999. “PLIC: bridging the gap between streamlines and LIC”. *Proceedings of the Conference on Visualization ’99: Celebrating Ten Years*. IEEE Visualization. Pages 341-348.

[VKP00] Verma, V., D. Kao and A. Pang. 2000. "A flow-guided streamline seeding strategy". *Proceedings of the Conference on Visualization ’00.* IEEE Visualization. Pages 163-170.

[War04] Ware, C.. 2004. *Information Visualization: Perception for Design*, Second Edition. Morgan-Kaufmann, San Francisco, CA.

[WG97] Wegenkittl, R. and E. Gröller. 1997. “Fast oriented line integral convolution for vector field visualization via the Internet”. *Proceedings of the 8th Conference on Visualization ’97.* IEEE Visualization. Pages 309-316.

[WGP97] Wegenkittl, R., E. Gröller, and W. Purgathofer. 1997. “Animating Flow-fields: Rendering of Oriented Line Integral Convolution”. Computer Animation ’97 Proceedings, IEEE Computer Society. Pages 15–21.