A NEW METHOD FOR PERCEPTUALLY OPTIMIZED VISUALIZATION OF TWO LAYERED FLOW FIELDS

BY

KARTHIKEYAN NATHAM
Bachelor of Computer Science and Engineering, University of Madras, India, 2002

THESIS

Submitted to the University of New Hampshire
In Partial Fulfillment of
The Requirements for the Degree of

Master of Science
In
Computer Science

September, 2005

This thesis has been examined and approved.

Thesis Director, Colin Ware,
Professor

Alejo Hausner,
Assistant Professor

James L. Weiner,
Associate Professor

Date

ACKNOWLEDGEMENTS

I would like to thank Colin Ware, my thesis advisor, for his support during the entire course of this thesis. I also thank all the members of the CCOM Data Visualization Research Lab for their mentoring and constant kindness in dealing with my queries. I thank my parents, brother and sister-in-law for their constant encouragement during my time at UNH. I also thank my thesis committee members, Alejo Hausner and James Weiner, for their time and criticisms during this process.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

A NEW METHOD FOR PERCEPTUALLY OPTIMIZED VISUALIZATION OF TWO LAYERED FLOW FIELDS

by

Karthikeyan Natham

University of New Hampshire, September, 2005

One of the most challenging problems in flow visualization is the ability to display one flow field over the other in such a way as to allow the viewer to perceive both as clearly as possible. A great many variations of attributes like color, shape, texture and orientation are required to define all the possible alternative visual representations of two layered flow fields. The problem is to find the optimal solutions for visualization of layered flows. This thesis develops and evaluates a new method for finding perceptually optimal or near optimal visualization of two layered flow fields given such a large parameter space. The method involves reducing the space first by identifying the important parameters of perception. Then an experimental study is designed and conducted that involves assessment of thousands of visual representation of the same two layered flow fields by human subjects. Henceforth, the experimental data obtained is analyzed to obtain principles for the display of two-layered flow fields.

This thesis also incorporates these optimal perceptual parameters in the development of an algorithm for time varying two-layered flow field visualization. The problem inherent in extending the algorithm for visualizing static two-layered flow fields to a time varying one is to be able to maintain the parameters like spacing as much as

possible between successive frames of the visualization. The best parametric combinations obtained from the perceptual study are used to produce effective time varying visualization of two-layered flow fields.

CHAPTER 1

INTRODUCTION

The power of computers allows us to collect, store and manipulate large volumes of data and transform them to images for humans to interpret and understand. The process of visualizing such raw data involves the mapping of data to images using some algorithmic technique. To create the most effective mappings various perceptual issues should be considered to allow us to grasp the information provided, as efficiently as possible. A good understanding of these perceptual factors of the human vision separates a visualization solution that is scientifically derived and a solution based on a trial and error methodology. The perceptual factors to be considered during the development of a visualization solution depend on the information to be displayed and on those aspects that are most relevant to the specific task being carried out by a data analyst. In the case of flow patterns the analyst might be interested in critical points (where the velocity is zero) or in advection paths (where a dropped particle would end up after a period of time).

The data produced by flow models is in the form of a field of vectors, which may be spaced uniformly or non-uniformly. The goal of the visualizations of flow data is then to map these vectors onto images, and in doing so represent the flow in an easily understandable manner so as to help in the analysis of various features of flow such as turbulence, vortices, advection paths and other structural forms. In many visualization problems it is desirable to display the flow field information in layers. For example, the ocean is layered with warm water on top and cold water in the bottom and the layers remain fairly separate except in a few places. The flow patterns in these layers are often different and visualizing such overlapping patterns presents a visualization challenge. Such visualizations of overlapping flow patterns could provide some useful insights into the interaction between different ocean components at different levels, and aid the researcher in understanding complex and significant ocean patterns that might affect the way organisms, plants and animals move from one layer to the other.

Adding multiple layers to the visualization increases the difficulty of the visualization problem considerably. It is necessary both to be able to clearly show flow patterns in the individual layers and for the analyst to be able to inter-relate features in the two layers. In the process of designing a visualization solution for two layered 2D flow fields there are a number of questions related to pattern perception that can be raised, which include: How can the two layers be displayed to allow least interference between them? How will the problem of occlusion influence the perception of the individual layers? If transparency is used on the top layer to minimize the occlusion of the bottom layer, what transparency level will make the two layers distinct yet clearly visible? In the visualization of two layered flow fields, what would be the optimal size of the visual element used to indicate the flow vector at a particular point in space that would allow maximal distinctness between the two layers? Answers to these questions are central to the design of a visualization solution for two layered 2D flow fields since pattern perception deals with the extraction of information structures from 2D space and also since most displays are two-dimensional.

This thesis describes a new method for obtaining perceptually optimized representation of two-layered flow fields. The remainder of this chapter provides an introduction to the various areas that are relevant to this thesis. It begins with a discussion of perceptual issues: color, pre-attentive processing, spatial channels, continuity, contour perception and transparency. This is followed by a discussion of the common methods that are used in flow visualization. Chapter 2 describes a method for rapidly producing streamline-based flow representations with different spatial characteristics, and an experiment designed to yield a perceptually near optimal solution to the two-layered flow visualization problem. Chapter 3 describes an extension to the algorithm described in Chapter 2 designed to allow the results of the study to be applied to the visualization of time varying two-layered flow fields. Chapter 4 concludes the thesis.

1.1 Perceptual issues in layered flow visualization

1.1.1 Perception of color:

Visual perception can be defined as the ability to understand patterns of light. The human visual system is sensitive to light only between the wavelengths 400 and 700nm. The photoreceptors in the retina absorb the incident photons of these wavelengths and transmit electrical signals to the brain. The phenomenon of color vision is attributed to the presence of three distinct color receptors, called cones, in our retinas that are active at normal light levels. There are also other photoreceptors, called rods, which are sensitive at low light levels and can be ignored because of their over-stimulation in all but the dimmest light levels. Cones have three types of wavelength sensitivity functions that peak at different wavelengths red, green and blue, whereas rods have only one wavelength sensitivity function. The cone wavelength sensitivity function illustrates that response to blue is small compared to red and green which suggests that blue does not carry lots of information. This can be attributed to the presence of very few short wavelength sensitive cones in the retina which are not very sensitive.

Hering (1920) proposed a theory that suggested the presence of six elementary colors which can be perceptually arranged as opposite pairs along three axes: black-white, red-green and yellow-blue. More recent research in neurology has confirmed this theory: From the retina a group of nerves, collectively called the optic nerve, carries the electrical signals from the photoreceptors of the retina to the brain. Before reaching the brain these signals are converted to the following three color channels:

Black/White (R + G): This is responsible for the perception of luminance contrast and is based on input from all the cones

Red/Green (R – G): This gives red or green color perception and is based on the difference between the long and middle wavelength cone signals

Yellow/Blue (Y – B): This gives yellow or blue color perception and is based on the difference between the short wavelength cones and the sum of the other two.

The spatial and temporal properties of the color channels are different. The Black /White channel has considerably greater capacity to resolve detail than the other two. The Black/White channel is also more capable of displaying continuous contours. The Yellow/Blue channel is the worst in displaying information (Ware, 2004). Nevertheless the presence of the different channels suggests a method for making different layers that are visually distinct.

1.1.2 Pre-attentive processing:

Pre-attentive processing can be defined as the phenomenon of the human visual system to quickly grasp the information displayed even before conscious attention. A simple example is the visual popping out of a curved line displayed with a set of straight lines. Information that is perfectly pre-attentive takes approximately the same amount of time to be perceived irrespective of the number of other distracting elements in the display. A number of experiments have been conducted to verify the pre-attentive capability of different elements used in data visualization like orientation, size, line width etc. Although these elements are found to be pre-attentive, their distinctness is found to be a matter of degree i.e. the more different the elements are, the more completely pre-attentive they will be.

Studies conducted to understand the neurological perspective of pre-attentive processing, like the one by Triesman and Gormican (1988), suggest that pre-attentive processing happens in the early stages of visual processing mostly in the primary visual cortex. The human visual system processes information in stages. Information about the orientation, texture, color and motion features are first processed in parallel and then serial processing takes place to capture the subtle details of the environment like 2D patterns, contours and regions. An architecture of the primary visual areas in the cortex proposed by Livingstone and Hubel (1988) suggests that the neurons in this region are tuned up to specifically detect particular pieces of information. These neurons, often termed as visual analyzers or channels, are sensitive to different spatial and temporal patterns. For example there are neurons in the visual areas that are differentially tuned to perceive orientation, width, length, texture, color etc. The receptive fields of many of the billions of neurons in the visual cortex can be represented by using a mathematical model called a Gabor function. This suggests that primary dimensions for differentiating low level patterns are the three dimensions of texture perception by the human vision: orientation, size (1/frequency) and contrast (Ware and Knight, 1995). The gabor model can also be used to predict visual interference between interleaved or overlapping patterns. Similarly oriented flow patterns are likely to visually interfere but it may be possible to use size and contrast to differentiate them.

1.1.3 Spatial channels:

An important concept in modern perceptual theory is that the visual system carries out something like a local fourier analysis of the visual image because of the gabor-like properties of receptive fields of neurons in the visual cortex. Information is separated according to "spatial frequency channels". Channels that are completely independent and do not interfere with other channels are said to be orthogonal. Wilson and Bergen (1983) found that the visual response could be modeled by the existence of three or four different spatial channels at each orientation, although these were not perfectly orthogonal. The model suggests that to clearly differentiate two patterns they should have size (spatial frequency) components that differ by approximately a factor of 3. In two layered 2D flow visualization, layers of information are displayed one over the other. It may be possible to separate them by varying them according to their spatial frequency components.

1.1.4 <u>Continuity:</u>

In the 1930s the gestalt psychologists developed a set of "gestalt laws" of pattern perception (Koffka, 1935). One among them is the principle of continuity, which states that humans more easily perceive smooth continuous contours rather than contours that abruptly change in direction. This clearly explains why a cross is perceived as two straight lines bisecting each other rather than as two right angles placed across each other. The principle of good continuity suggests that flow patterns may be better represented by using long continuous contours instead of the grids of short line segments or arrows that are commonly used.

1.1.5 <u>Contour perception:</u>

A contour can be defined as a continuous perceived boundary, between different regions in space. Contours can be defined by lines, color differences, binocular disparity, motions pattern discontinuities or texture discontinuities. Contours that have good continuity are more readily perceived. Flow visualization solutions are all about allowing the viewer to perceive the flow pattern by means of a set of contours. A set of experiments conducted by Field et al (1993) refines the Gestalt principle of good continuity. In their experiment, a set of 256 Gabor patches (which can be thought of as small visual elements with particular orientation and size, generated using a Gabor function) were randomly oriented and subjects were asked to detect the presence of contours among those patches. Results suggested that patches that were oriented to form a continuous path were more easily detected.

The phenomenon of contour perception has direct application to the visualization of 2D vector fields. One common technique for showing 2D vector fields, is to place arrows that are oriented in the direction of the vector field in a regular grid. Each arrow is a short, straight contour with an arrowhead attached. Ware (2004) suggests that vectors that are displaced to form a smooth continuous contour should be better than short line segments in showing a flow pattern. Various researchers have developed algorithms for producing long continuous and evenly spaced contours (Turks and Banks 1996). A drawback with this method is the 180 degree ambiguity in the sense of direction: for a given contour there can be two directions of flow. Fowler and Ware (1989) experimentally verified a method for resolving this ambiguity in flow direction, wherein the vector is drawn as strokes that are tapered in one end and blunt in the other. It was shown that the direction of the field was unambiguously taken to be towards the blunt end. However, this could not work with long contours. They also developed a new method for generating vector fields that represent the direction of the field unambiguously using gray scale color changes along the length of the strokes. Strokes that started with gray scale values close to the background and then increasingly contrasted with the background unambiguously showed vector direction.

1.1.6 Perception of transparency in producing overlapping layers of data:

In the visualization of two layered 2D flow data it may be desirable to display the foreground contours in a transparent form to allow the background layer to be seen as well. Transparency in computer graphics is made possible by specifying a blending value, called alpha, along with the red, green and blue color components. When a foreground pixel is drawn with transparency enabled, the final pixel value is calculated as: Final pixel value = current pixel value * (1 – alpha) + (new value * alpha). So basically the pixel drawn retains a fraction of the existing pixel's color component, which means that an alpha value of 1 is fully opaque and an alpha value of 0 is fully transparent. A good visualization of two layered flow data must use transparency in a careful manner, by choosing those degrees of transparency (alpha values) that add little to the problem of perceptual interference.

1.2 Previous evaluation studies:

There are a variety of techniques for visualizing flow data. These techniques range from representing data simply in the form of oriented arrows to dense texture based algorithms (Laidlaw et al, 2001). The motivation for all these techniques for visualizing 2D flow fields is to represent the various features of flow as clearly as possible. Often times, algorithms are proposed to solve one particular problem in visualizing flow data while ignoring the importance of the other features. For example, line integral convolution (Cabral and Leedom, 1993) is a texture based dense integration algorithm capable of representing the orientation of flow but fails to represent the flow direction (it could be in one of two directions). A study to quantitatively evaluate the performance of six methods for visualizing 2D flow was conducted by Laidlaw et al (2001). The methods used were: regular grid of arrows, jittered grid of arrows, line integral convolution, image guided streamline (in this case the arrows were curved and placed on continuous contours), streamlines with a regular grid of starting points and a method that used elongated triangles. The evaluations were carried for 3 different tasks: judging the location of critical points, (which can be defined as those points at which the magnitude of the vector field vanishes); detection of the type of these critical points (attracting focus, repelling focus, attracting node, repelling node and saddle points); and for a particle dropped in the center of a circle, determination of its advected position on the circle's circumference.

Laidlaw et al's results clearly illustrated the pros and cons of the six visualizations in representing the user tasks. Image guided streamlines were found to be efficient and accurate in representing the directional information compared to the other methods. Locating the critical points was found to be more accurate with line integral convolution. Streamlines seeded on a regular grid were helpful in locating the types of critical points more than the other methods. Such quantitative analysis of the flow visualization methods was revolutionary in evaluating the effectiveness of flow visualization methods.

1.3 Methods for flow visualization:

Visualization of 2D flow data involves mapping vector field data to images by using an algorithmic technique. There are various techniques for visualizing flow fields which can be broadly classified into two types: scalar and vector field visualization techniques.

1.3.1 Scalar field visualization techniques:

Scalar field visualization involves the representation of only the magnitude information of the flow data and not the directional information. A common method for representing scalar flow fields is color coding, where the entire scalar data is mapped onto a range of colors. The change of colors in such scalar mappings can show the transition in the magnitude of the flow field. Research suggests that to be easily read, color data values should be mapped into a monotonically perceptually ordered color sequence (Ware, 2004). Also the use of a pure hue change (e.g. from red to green) cannot show much detail. A luminance component in the color mapping is important to show detailed changes (Ware, 2004). Another scalar field visualization technique, called isocontouring, represents the scalar data using constant valued 2D curves or 3D surfaces. Color coding when used with isocontouring can refine the visualization of scalar flow fields in displaying the transitions in the magnitude of the field.

1.3.2 Vector field visualization techniques:

Vector field visualization involves the representation of directional information in the visualization. A very basic technique for visualizing vector field information is to represent the vector field as grid of arrows oriented along the direction of the field. These arrows can be of unit length to display only the direction, or can be of varying length to display both the direction and the magnitude of the vector field along that direction. Klassen and Harrington (1991) developed a number of variations on this (that they called hedgehogging). Mostly these consisted of modifying the arrowhead although they also experimented with wedge shapes. One problem associated with visualizing vector fields using arrows is the visual clutter that results with the increase in the density of the data points.

Techniques for visualizing 2D flow fields exist that make use of smooth continuous contours tracing the direction of the flow field. There are two kinds of approaches possible in such visualizations: the Lagrangian approach and the Eulerian approach. The Lagrangian approach displays the path of a single particle traversing through space with time. The contour generated by that particle is termed as the particle pathline. The Eulerian approach produces a streamline which is defined as the contour in a flow field that is tangential to the velocity vector at every point for a particular instant of time. Like pathlines, streamlines can be generated by particle tracing and many streamlines can be generated to produce a visualization. Unlike pathlines, streamlines cannot cross. Pathlines and streamlines are identical in the visualization of steady flows that do not change with time. Turks and Banks (1996) make use of streamlines to develop an algorithm for representing 2D flow fields in a hand-drawn style. Their algorithm follows an iterative approach to generate flow fields that are neither too sparse nor too dense by allowing three kinds of operation: changing the position and length of the streamlines, joining streamlines that nearly abut, and, creating new streamlines to fill large unfilled gaps. Jobard and Lefer (1997) developed a faster method of generating constant spaced streamlines using a single pass algorithm that began with a single streamline and then added adjacent ones until the field was filled.

There are other vector field visualization methods that make use of texture based techniques to produce dense flow field images in fine grain detail. The spot noise algorithm proposed by Van Wijk (1991) makes use of spots, represented by small motion blurred ellipses that are randomly splattered in space with their major axis oriented towards the direction of the flow field, to produce dense textured images of flow fields. Another texture based approach called Line Integral Convolution (Cabral and Leedom, 1993) uses textures convolved along pathlines or streamlines to reveal either steady or unsteady flows. The result is a kind of blurred texture pattern oriented with the direction of flow.

1.4 Analysis:

It is notable that all the vector field visualization methods discussed produce contours oriented either in the direction of streamlines or pathlines. In the case of Line Integral Convolution these are elongated blurred patches; in the case of hedgehogs these are short line segments; in the case of particle tracing methods these are long contours that may change in color along their path. This suggests a method for generalizing the problem. We may consider all of the flow methods as producing "streaklets". Streaklets may vary in length, and change in shape or color along their paths. Streaklets may be fuzzy or sharp and they may be transparent or opaque. Considered in this way, the problem of flow visualization becomes a problem of perceptual optimization – how can the flow field be mapped to the visual properties of a set of streaklets so that the flow field can be perceived most effectively? This is the central idea underlying this thesis.

The study conducted by Laidlaw et al (2001) compared six different representations of the same flow field and showed that some representations were better at representing certain flow features than others. However, this study can be criticized by pointing out that there are many parameters than can be manipulated in each flow representation. For example, the study made use of the line integral convolution algorithm, but used a form that only showed orientation and was ambiguous with direction. A variation of this – oriented line integral convolution might have been used [Wegenkittl & Groller, 1997]. Also, for each of the flow representations used in the study different parameters setting line width, arrow length, etc. might have been used. It is not known if the representations used in the study were the best that were possible for each algorithm. In order to produce some general results about the relative values of different representations it is essential to have them optimized in some way. The goal of this thesis, hence, is to perceptually optimize the display of streaklets in the visualization of two layered flow fields by evaluating a number of parameter values like streaklet size, shape and color that determine the mapping from vector field to streaklets in the visualization.

CHAPTER 2

PERCEPTUALLY OPTIMIZING THE VISUALIZATION OF STATIC TWO LAYERED FLOW FIELDS

The study of the effectiveness of visualization has been traditionally approached by means of studies that vary at most one or two display parameters while keeping the others constant (for example, Cumming et al, 1993). The problem is that for flow visualization the number of parameters is quite large and conducting a systematic study on all of the parameters of the visualization problem using the traditional experiment with objective measures of performance would be impossible. If the parameters interact, the total number of measurements required increases exponentially with the number of display parameters. Such a study would take thousands of hours of participation by each subject if traditional psychophysical methods were used. The problem gets even worse for the display of multiple layers of flow, a common requirement for visualizing ocean and estuarine currents.

This thesis develops and applies a new method for creating near optimal visualizations of two layered flows. The method has two parts: The first is to analytically determine those variables that are most important to the problem of displaying two layered 2D flow fields. This is done based on a review of the perceptual factors (discussed in Chapter 1). The second part is to devise a method for empirically searching the parameter space for high quality solutions. This method involves the development of a static flow field display algorithm that maps the parameters to a set of streaks. This is then applied in an experiment involving human subjects.

Finally an algorithm for visualizing time varying flow data is developed that maps the high quality solutions obtained from the perceptual optimization study to produce effective time varying visualization of layered flows.

2.1 Overall approach:

The overall approach to this study, having its motivation from a previous study on the perception of two layered surface textures (House and Ware, 2002), involves the following steps that are considered essential during the optimization process for visualizing two layered flow fields.

1. Generating a parameterization: The strategy involved in the optimization process is to search for high quality solutions in a well-defined parameter space. Thus it is necessary to define a set of parameters that determine a mapping from the flow data to the visual representation. In the case of this thesis, it is necessary to determine how such parameters as streak width, length and color can be based on vector field direction and magnitude.
2. Identifying the visualization method to display the vector of parameters: The next stage in the optimization strategy is to develop a visualization algorithm that can map these parameters to produce unique visual representations of flow fields. A number of techniques for vector field visualization have already been discussed to illustrate their significance as a vehicle for displaying flow data. A new algorithm has been developed for this study that generates smooth, continuous streamlines to display flow fields. This allows for visualizations to be generated consisting of

streamlines with different widths, spacings, color, transparency and degree of blur.

3. Conducting a human-in-the-loop optimization: It is necessary to use human subjects to explore the parameter space to determine those solutions that are effective and those that are not. So an exhaustive study was carried out using 5 subjects who each assessed a very large number of displays of the same two layered flow field.
4. Characterizing the solutions: The output of the optimization process is analyzed to determine general guidelines for producing good solutions to the visualization problem.

In the following sections these steps are described in more detail.

2.2 Step 1: Generating a parameterization:

There is a huge variety of possible streaklet parameters that can be used in the optimization procedure. These parameters include regularity, width, shape along streaklet, streak length, spacing, overall color, degree of blur, degree of transparency, background color and other dynamic properties of streaklets. From the review of the literature concerning human perception of contours the following perceptual parameters were determined to be the most relevant to the problem of displaying layered flow fields: contour width, contour spacing, contour color, degree of "blur" and transparency for the top flow layer. Choosing a continuous range of real data values for all of the parameters will make the parameter space infinitely large. Even if many discrete values are chosen for the various parameters, the parameter space will still be too large for an exhaustive search. In order to make the study feasible the set of parameter values was reduced as much as possible while still covering an interesting range of values. These sampled parameter values were chosen based on the results obtained through preliminary experiments. The various parameter values used for the perceptual study are as follows:

1. Four streaklet widths: 2, 4, 8 and 16 pixels
2. Three streaklet spacings: represented as a ratio (width_of_streak:spacing_of streak): 1:1 (no spacing), 1:2 (medium spacing), 1:4 (large spacing)
3. Two color schemes: Red foreground streaks & Green background streaks, and Gray scale for both foreground & background streaks
4. Three transparency levels (only for the foreground flow field): Alpha values of 0.5, 0.75 and 1 were used.
5. Three degrees of "blur": No blur, medium blur, full blur. A blur effect was created by using a texture, of size 64 by 64, mapped onto the streak path that had different across streak profiles in the alpha values (mapped to transparency). The "no blur" texture is a simple box function; the "medium blur" texture is a semi-circular function; the "full blur" texture is a sinusoidal function (see Figure 1). All these textures have a sawtooth ramp along the streamline to give sense of direction (see Figure 2). The streaks of the flow field are drawn in a black background and are allowed to be as long as possible.

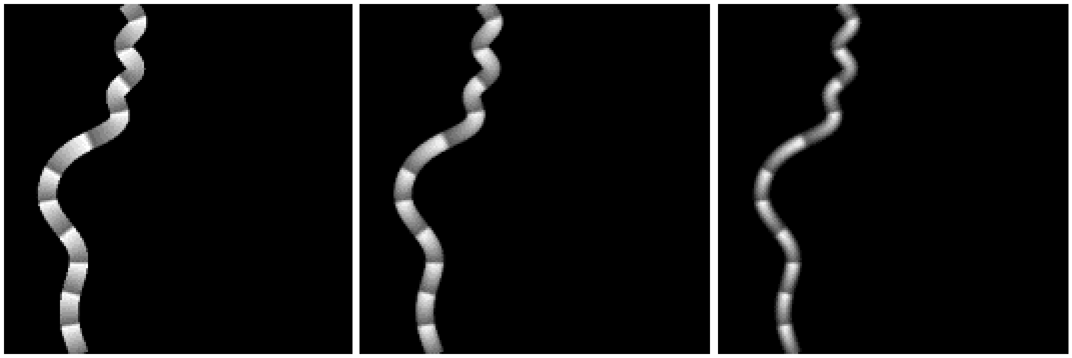

Figure 1: Streaklets with No blur (left), medium blur (center) and full blur (right) textures.

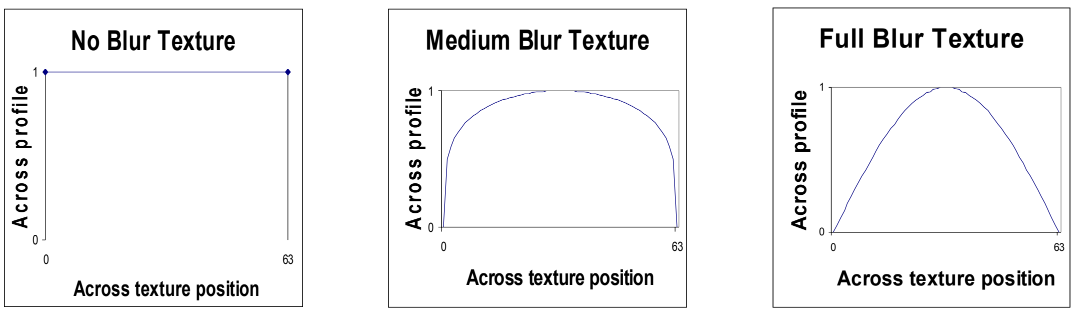

Figure 2: Across profiles of no blur (left), medium blur (center) and full blur (right) textures.

All of these parameters, except for transparency were applied to both the foreground and background flow renderings. This yielded 3456 different representation of the same two flow layers. Eight of these are illustrated in Figure 6.

2.3 Step 2: Developing a streaklet generation algorithm:

The next step in the optimization process is to develop an algorithm that maps the vector of parameters to their respective visual displays for static flow fields. This algorithm has to meet the following requirements of the study:

1. generate streaks that follow streamlines in a flow field

2. allow for specified streak width and spacing
3. allow no overlaps between streaks
4. make the streaks as long as possible

The method formulated is based on a modification of an algorithm originally developed to produce a poisson disc distribution (Cook, 1986). This is similar to the method developed by Jobard and Lefer (1997). The strategy is to randomly choose streamline starting positions and then trace the contour generated by the flow field in the forward and backward directions until the streamline can no longer be extended. Two buffers, namely the *spacingBuffer* and the *drawBuffer*, are used for this purpose to draw streamlines that have a specified streak width and spacing.

The input to the algorithm is a flow field that can either be a function or a numerical model or stored grid of vectors. All that is required for input is a function

v = **GetFlowVector(*p*)**

that returns a flow vector *v* for a location *p* in the flow field. The data structures used in the algorithm include a *spacingBuffer*, that is initialized to black, and the *drawBuffer*, that is used to render the streaklets onto the display. The streaks are stored in a *streakArray* that holds a series of positions that define an advection path.

The following pseudo code presents an outline of the algorithm:

DrawFlowPattern(*spacing, width*)

{

while (*spacingBuffer* unfilled)

{

1. *startPosition* = ***findEmptyPixel()***

2. Start a streak at *startPosition* and ***traceForwards(****startPosition****)***

3. Trace the streak backward in the same fashion.

4. If (*streakLength* > *lengthThreshold*)

   a. Draw the streak in white color, into the *spacingBuffer* at *spacing* times the *width*.

   b. Draw the steak into the *drawBuffer* using the correct *width*

}

}

findEmptyPixel()

{

repeat

randomly choose an x,y position

color = ***spacingBuffer(****x,y****)***

until (*color* == black OR number of attempts exceeds 1000)

}

traceForwards(*startPosition****)***

{

p = *startPosition*

repeat

set the middle of the *streakArray* to *startPosition*

v = ***GetFlowVector(****p****)***

p = *p* + *v*;

store *p* in next position in the *streakArray* until(*p* falls on a white pixel OR v < *threshold* OR p outside *viewport*)

}

The algorithm iteratively populates the *drawBuffer* with streaks until it is filled completely. Each iteration starts with sampling the *spacingBuffer* for a black pixel that can be used as a starting position for the streak. From this starting position the streak is traced along the contour followed by the flow field in the forward direction until one of the following conditions are encountered:

1. The trace crosses the display viewport
2. The flow vector is less than a set threshold
3. The advected position of the trace falls on a white pixel

The streak is also traced in the same manner in the backward direction. During the entire course of the forward and backward tracing, the advected particle positions are stored in the *streakArray*. The *streakArray* is then used to draw the streak into the *spacingBuffer* with a width that is a multiple of the required amount of spacing and the same streak is drawn in the *drawBuffer* using the correct width. In order to avoid the presence of extremely small streaklets that do not convey any useful flow information, only those streaklets whose length is more than a fixed threshold are drawn.

Figure 3 & Figure 4 show the state of the *drawBuffer* and the *spacingBuffer* after the generation of five streaks, and after the *drawBuffer* is completely filled.

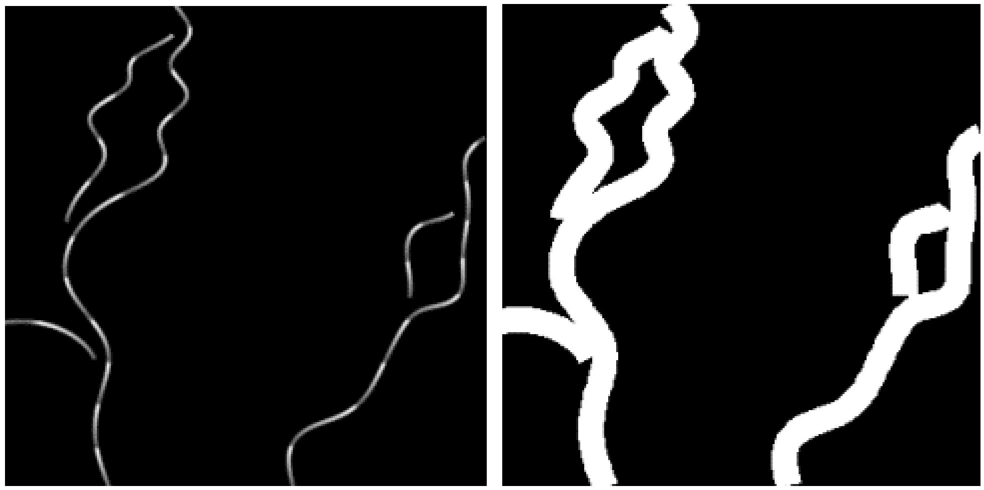

Figure 3: *drawBuffer*(Left) and *spacingBuffe*(Right) after generation of five streaks.

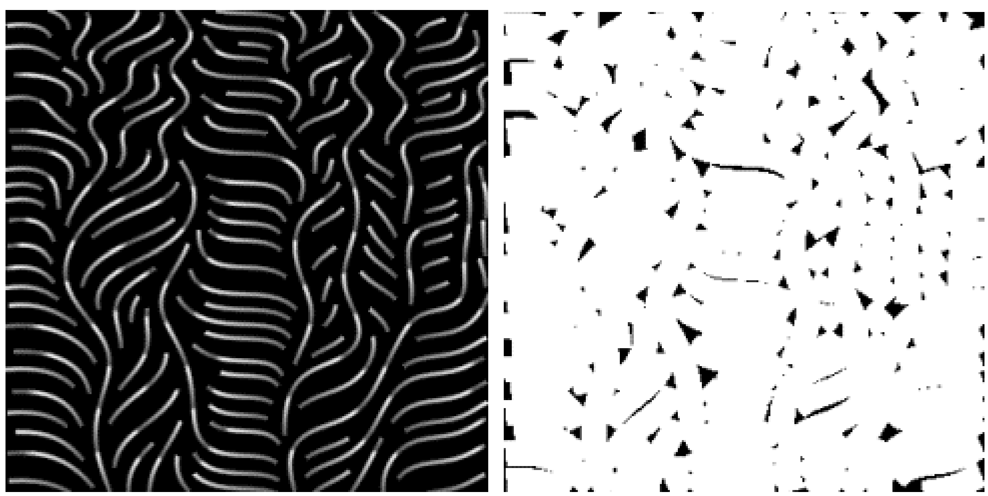

Figure 4: *drawBuffer*(Left) and *spacingBuffer*(Right) after the *spacingBuffer* is completely filled.

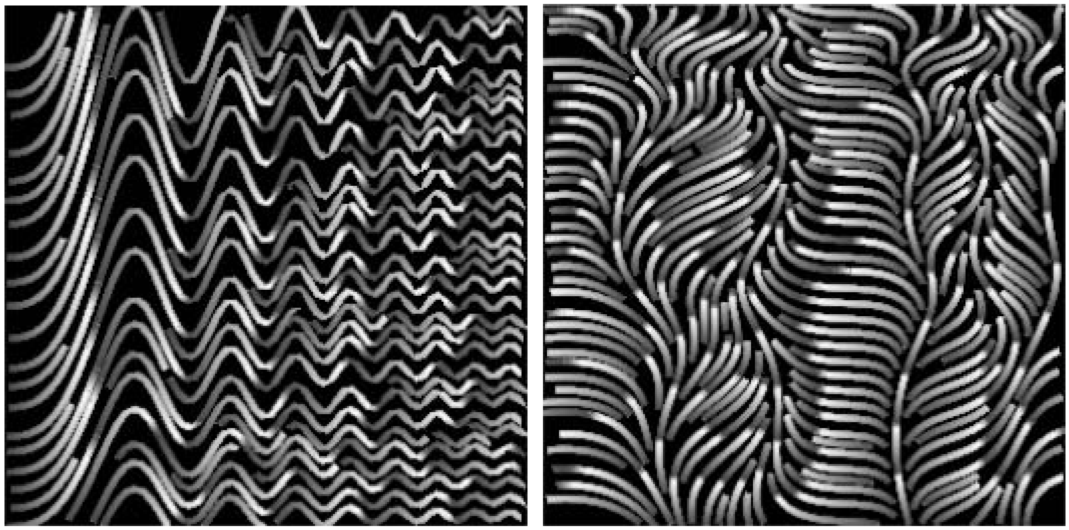

Figure 5: Sample rendering of the two different flow fields used in the study. Subjects saw these with the right hand flow pattern layered above the left hand flow pattern.

2.4 Step 3: Experiment design:

2.4.1 High resolution display:

One of the goals of the perceptual optimization process is to carry out the study at a resolution that is close to the limits of the resolution of the human eye. For this purpose, the study was conducted using a high-resolution ViewSonic VP2290b display having a maximum resolution of 3,840 by 2,400 pixels, on a screen measuring 48cm x 30cm. The individual pixel size is 0.125mm and at a viewing distance of 1 meter, this yields a visual angle of 0.42 minutes of arc, closely approximating the limits of resolution of the human eye. The visual acuity of the human visual system is very non-uniformly distributed throughout its visual field and the central 25 degrees of the visual field activates more than 60% of the visual cortex (Ware, 2004). So, even though the visual angle of the entire high-resolution display is only about 27 degrees, it captures most of the human ability to perceive patterns.

2.4.2 <u>Flow field representations:</u>

The 3456 different visual representations of the same two-layered flow pattern were obtained using the combination of the following parameter values:

1. Width and degree of blur (8 values each for the foreground and background streaks):

   2S, 4S, 8S, 8M, 8B, 16S, 16M and 16B

   where,

   S - Sharp (No Blur)

   M – Medium Blur

   F – Full Blur

2. Spacing (3 values each for the foreground and background streaks):

   1:1 (no spacing), 1:2 (medium spacing) and 1:4 (large spacing)

3. Color (2 values):

   Red foreground streaks & Green background streaks, and Gray scale for both foreground & background streaks

4. Transparency (3 values only for the foreground streaks):

   Alpha values of 0.5, 0.75 and 1

The parameter values chosen are based on the results of the preliminary experiments conducted during the initial course of the study.

2.4.3 <u>Experiment:</u>

Because of the large number of two-layered flow representations it was necessary to have a very rapid evaluation method. The method chosen was for the subjects to rate each display in terms of the visual quality of the foreground and background flow patterns. They had to give each a rating between 0 and 9, where 0 represents a visualization that is least perceivable and 9 represents the most perceivable visualization. So that subjects could know a good representation from a bad one, the same pair of foreground and background patterns was used for all display solutions. These patterns were constructed so that they both contained different amounts of detail in different regions. The two patterns are shown separately in Figure 5.

All of the 3456 displays were presented to each subject in a different random order. In order to speed up the rendering of the different display solutions in a single screen of the experiment, the advection paths of the flow patterns were pre-computed and stored in the form of binary files. Rendering the flow patterns by reading the advection paths from these files reduced the total duration of the experiment by a considerable factor. Figure 5 represents one screen taken from the experiment. The experiment was also designed in a way that allowed subjects to stop the experiment after some time and resume it from the same point. This allowed the subjects to finish the experiment in multiple sessions.

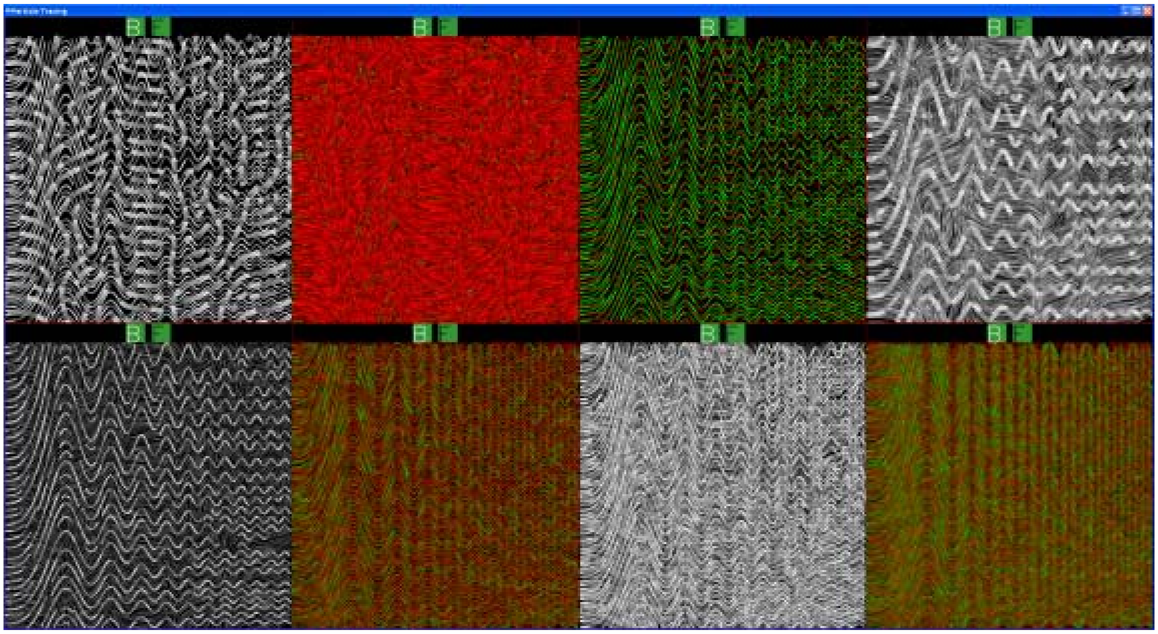

Figure 6: A single screen from the experiment showing 8 different solutions randomly selected from the total set of 3456.

2.4.4 Subjects:

The subjects chosen for the experiment were four undergraduate students in various disciplines and a graduate student in Computer Science at the University of New Hampshire. Since the perceptual optimization process includes the perception of color in the visualization of the two layers, it was made sure that the subjects of this experiment were not color blind. The subjects were paid for their participation.

2.4.5 Experiment procedure:

Before the subjects started the experiment, they were told the general goals of the experiment and its significance to two layered flow visualization research. In judging the different solutions the subjects were asked to take into account how much information they could see in each layer and how distinct the layers were. They were also asked to take into account whether or not they could see detail in the layers. A good representation of the foreground and background flow pattern, that clearly displayed the flow field information and detail, were provided as printed copies to the subjects. Subjects were advised to use these representations as comparisons, in case of any confusion while rating the different displays in the experiment.

A training session was conducted wherein the subjects were allowed to go through a small subset of the experiment. During this, the subjects were asked to explain their rationale in rating a particular visualization in order to check their understanding of the visualization problem. Once the subjects were comfortable and confident in proceeding with the experiment, the actual experiment was started that showed 8 visual mappings chosen in a random order from the 3456 different displays in a single screen. The experiment was not timed. This gave enough time to the subjects for comparing the eight display solutions for the amount of information visible in the two layers. Subjects had to rate the background layers of all the eight displays first and then rate the foreground layers. The individual layers were rated on a scale from 0 to 9, with 0 given to the visualization that is least perceivable and 9 given to the visualization that is most perceivable. The background and foreground ratings were stored in a text file along with the corresponding perceptual parameters. When the current screen was completed, subjects could proceed to the next screen. On average, the experiment took 10-15 hours of observation per subject.

2.5 Step 4: Characterizing the solutions – Analysis and Results

The data collected from the perceptual optimization process was analyzed to infer general design guidelines for the visualization of two layered 2D flow fields. The metric used as a measure of quality was the product of the foreground rating and the background rating. One of the problems with analyzing data of this kind is that the conventional tools such as analysis of variance were not really applicable because we were only interested in the characteristics of relatively small percentage of solutions that were judged to be "good". Only 20 percent of the solutions had a rating of 40 or better (compared to a maximum possible rating of 9x9 = 81).

Two approaches were taken in characterizing the solutions. First, the top 20 solutions were characterized. Second, a tool was created that allowed the visualization of a pattern involving multiple factors of foreground and background streaklet spacing and width.

2.5.1 The top 20 rated solutions

Out of the top 20 solutions, 19 had red-green coloring while only one had grey-scale coloring. It was also notable that all of these solutions had a background width of 16 pixels with all but one having a background spacing of 1:4. For the degree of blur in the background, 10 had maximum blur, 6 had medium blur and 4 were sharp. The foreground width was more varied for the top 20 solutions: 4 had a width of 2 pixels, 6 had a width of 4 pixels, 10 had a width of 8 pixels but none had a width of 16 pixels. The narrower widths were always sharp (not blurred). However, the 8 pixel width solutions were preferred with full blur 8 out of 10 times. Of the foreground spacings, 1 had a spacing of 1:1, 4 had a spacing of 1:2 and 15 had a spacing of 1:4. 15 out of the 20 top solutions had some degree of transparency in the foreground (either 0.5 or 0.75).

Thus it can be said that the best judged solutions were red-green, had the widest possible background widths (16) and spacing of 1:4, mostly had a foreground width of 8 (blurred) and a spacing of 1:4. Eight of the top 20 solutions fit this description. The very highest rated solution is illustrated in Figure 7. It had most of the characteristics described above but with a foreground width of 4 and a medium background blur:

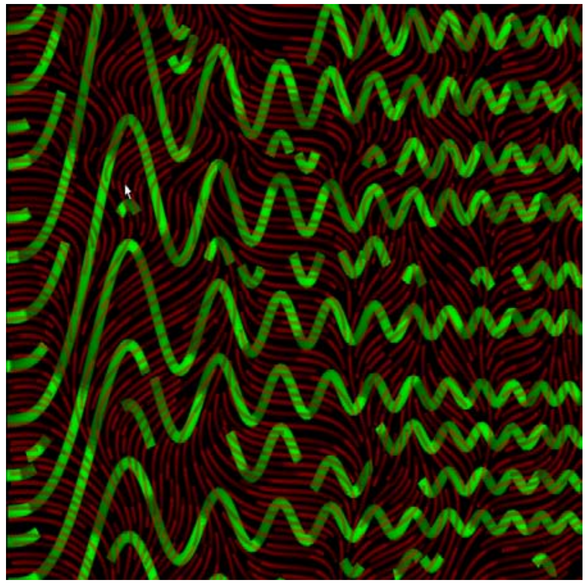

Figure 7: The highest rated solution

2.5.2 **Pattern analysis tool**

A tool was constructed to allow us to see the effect of different combinations of foreground and background width and spacing. This was used to analyze the pattern of data obtained with different values of blur and transparency. It is illustrated in Figure 8

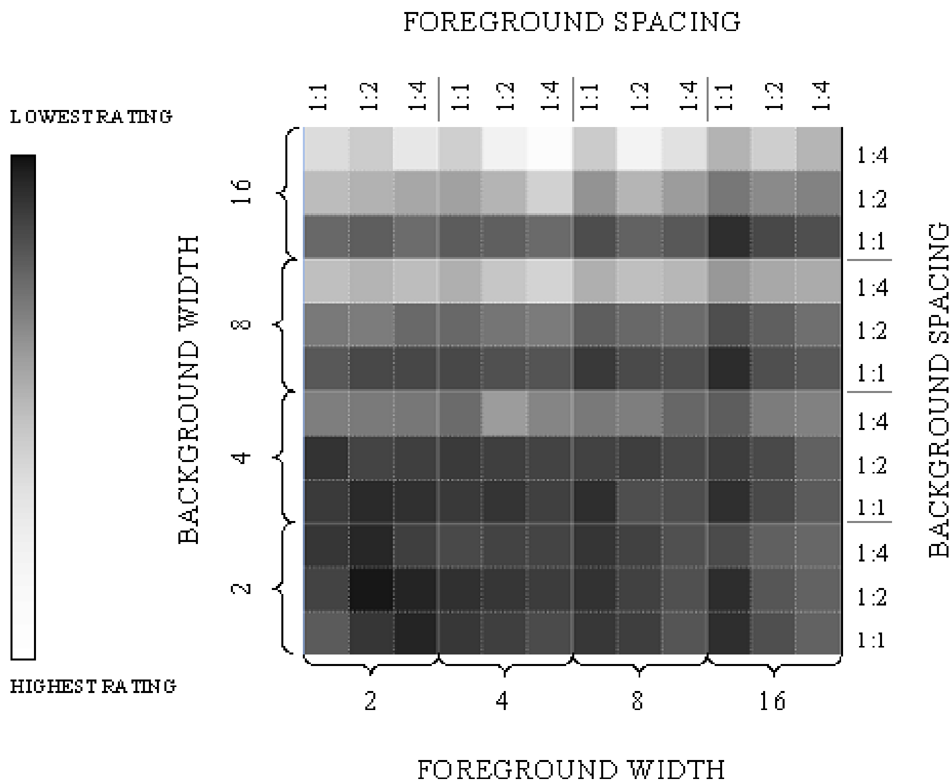

Figure 8: Gray scale level (Left) used in the Pattern analysis tool (Right)

The tool consists of a 12 by 12 grid of cells that plots the 12 values of foreground width & spacing to the 12 values of the background width & spacing (4 values of width * 3 values of spacing). The other parameter values: foreground blur, background blur, color and foreground transparency, are the constant valued inputs to the display. In this way, each cell in the display refers to exactly one display solution among the 3456 different display solutions. The average of the ratings of all the subjects for the display solutions are mapped to the gray scale color of the cell referring to the particular display solution.

Through exploration with this tool we determined that degree of blur had little effect on the overall pattern but that degree of transparency had a relatively large effect on the overall pattern. Red-green color was generally better than gray scale but the effect seemed to be a constant increment in rating. Therefore we averaged over degree of blur and color to produce the summary shown in Figure 9.

The three plots show all combination of foreground and background width and spacing for the three transparency levels. Figure 9 (left) shows the results pattern for 1.0 transparency. Figure 9 (center) summarizes the result for 0.75 transparency and Figure 9 (right) summarizes the result for 0.5 transparency. As can be seen the pattern of results was quite different for the different transparency levels.

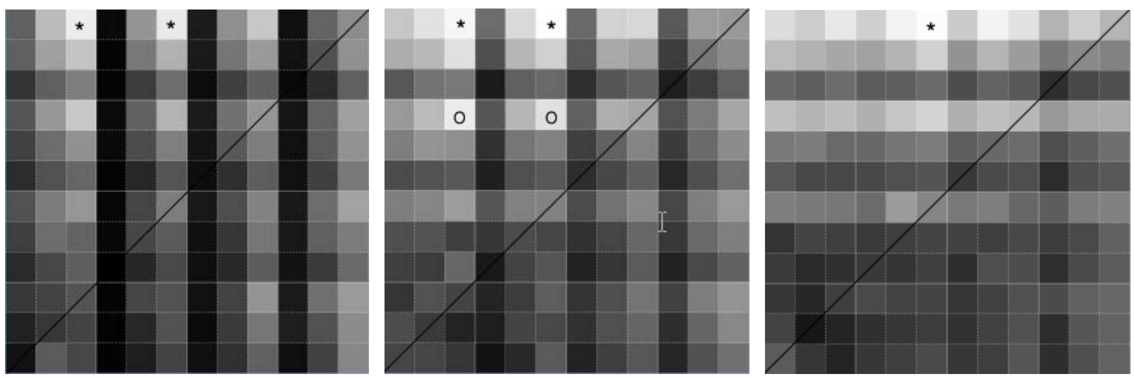

Figure 9: Patterns for transparencies 1.0(Left), 0.75(Center) and 0.5(Right).

With transparency level 1 (opaque) the dominant pattern is vertical stripes indicating that the most important factor is background width and spacing. With a 1:1 spacing and no transparency ratings were low resulting in the black vertical lines. The obvious reason for this is that if lines are packed together and not transparent it is difficult to see between them. The best overall ratings were obtained in the cells marked with a '*'. These have foreground widths of 2 and 4 pixels with spacing of 1:4. The background widths were 16 with a spacing of 1:4.

With transparency level 0.75 a similar overall pattern is obtained but there is less overall difference between the conditions. The best solutions is marked with a ‘*’. And these are the same as the best two solutions for the no transparency condition. However it is also worth noting that the two cells marked with an ‘o’ also provided good solutions. This is important because these patterns, having thinner background streaks (8 instead of 16) would enable more detail to be seen in a flow pattern. (See Figure 10).

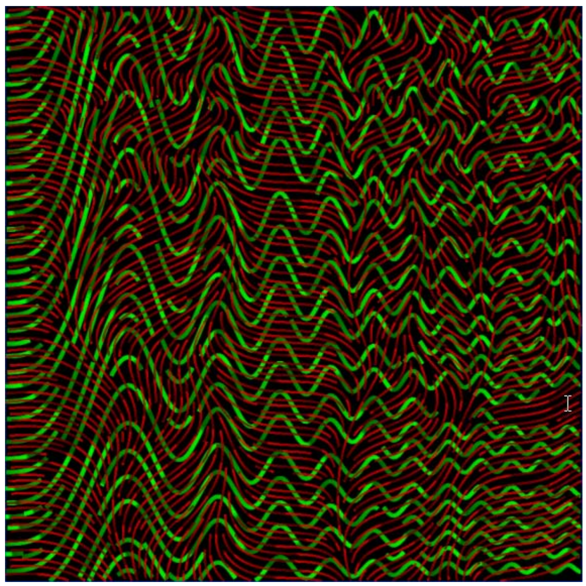

Figure 10: A good solution with thinner background streaks

With transparency level 0.5 a horizontal pattern is most evident in the results, indicating that foreground width and spacing is more important than background width and spacing (the opposite to the case with transparency level 1). However the highest rated solution overall (marked with a ‘*’) is the same as for the other two levels of transparency.

It is also notable that for the 0.5 transparency and the 0.75 transparency conditions, solutions above the diagonal are generally better than those below the diagonal. This means that subjects preferred solutions with wide background streaks and narrow foreground streaks over the reverse (wide foreground streaks and narrow background steaks). Overall it is notable that a spacing of 1:4 was preferred over spacing of 1:2 and 1:1 for almost all foreground and background conditions.

In order to measure the overall effect of color on the display solution, the top 100 patterns that were red and green colored were taken. Then the matching patterns that corresponded in every other variable were found and the mean ratings were determined. The mean ratings obtained:

with red foreground/green background = 57.992

with gray scale foreground and background = 53.148

Applying the same mechanism to the foreground blur, the mean rating obtained for:

no blur = 47.232

medium blur = 48.724

full blur = 52.612

Applying the same mechanism to the background blur, the mean rating obtained for:

no blur = 50.462

medium blur = 53.036

full blur = 54.108

2.6 Visualization of a real two-layered flow data:

The streaklet generation algorithm was used to visualize flow patterns, generated by NOAA, from the Chesapeake Bay Operational Forecast System (CBOFS). Two of these layers were selected and the visualization was applied to them. A subset of the data in a grid of size 416 by 240 was used as an input for the algorithm. One such visualization displaying two layers on an image of size 512x512 pixels is shown in Figure 11. The visualization parameters used for this display is from the right hand cell marked with an ‘o’ in Figure 9(Center).

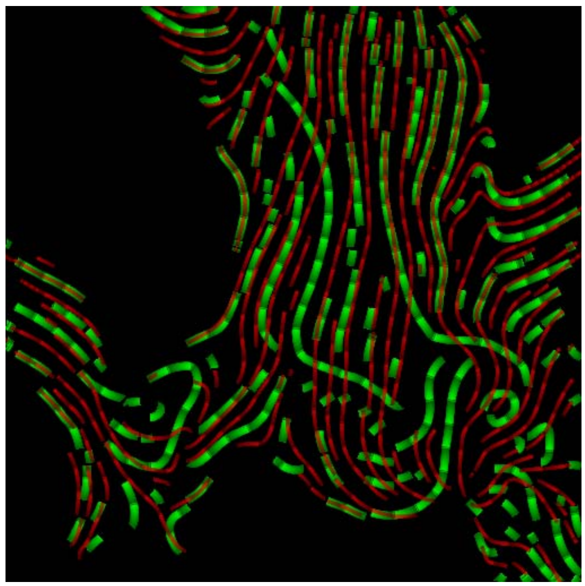

Figure 11: Two layers of the flow pattern generated by NOAA from CBOFS.

2.7 **Discussion:**

From the analysis of the data collected from the perceptual study a number of inferences on the design of good display solutions for visualizing two-layered flow data can be made. The best solutions mostly have red foreground and green background layers. This suggests that applying different color channels to the two layers helps in distinctly identifying the layers. Also in the visualization of flow patterns in two layers, it is definitely useful to vary the width of the streaks used to represent foreground and background layers to differentiate them. None of the good solutions have both foreground and background streaks of the same width. Most of the solutions that were highly rated by the subjects have wider background streaks and thinner foreground streaks. So displaying the patterns by varying the spatial frequency (size) component by approximately a factor of 3 (as discussed in Chapter 1) helps in separating the two layers clearly. Though subjects seemed to prefer such coarse solutions, such displays do not show much detail. Most of the top solutions had a foreground and background spacing of 1:4. 1:4 spacing in the foreground allows the background pattern to be seen through the spaces of the foreground streaks. The results also showed that some amount of transparency in the foreground layer helps perceive the background streaks better, since perfectly opaque foreground patterns will occlude the background pattern.

CHAPTER 3

THE VISUALIZATION OF TIME VARYING FLOW FIELDS USING PERCEPTUALLY OPTIMIZED STREAKLETS

The power of computers has made possible the generation of large amounts of time varying vector field data from various scientific and numeric simulations. This necessitates the design of good visualization solutions for time varying 2D flow data. In the visualization of time varying flow fields, a number of questions related to the display of the features of flow can be asked like: At what instant in time does a flow feature appear? Does the flow feature appear in a cyclic fashion? How much time has elapsed since the appearance and disappearance of a flow feature? How fast does the flow feature change? Is the formation of the various flow features related? Answers to these questions can be obtained by designing good time varying 2D flow visualizations. In many instances, time varying visualization of layered flow is necessary since estuarine and other ocean flow models are layered and evolve over time. Time varying visualization of two layers in such flow models can reveal useful information like the relative speed of the layers, which might allow researchers to study biological processes or the movement of physical elements between the layered ocean components.

A challenge in such visualizations of dynamic flows is to achieve coherence between successive frames of the animation. Also perceptually optimizing the display elements in each frame of the visualization will help perceive the attributes of the different flow layers as distinctly as possible. This chapter describes an extension of the streaklet generation algorithm (discussed in Chapter 2) to allow for the display of unsteady 2D flows. The algorithm allows for the perceptual parameters discovered for displaying 2D static flows to be applied to dynamic flows and also achieves frame-to-frame coherence.

As had been discussed in Chapter 1, there are various techniques for visualizing scalar and vector flow fields. Streamlines and pathlines are among these techniques that are used to visualize 2D vector fields. Streamlines and pathlines are identical in the visualization of static flow fields that represent a flow field at a particular instant of time. In the visualization of time varying flows it is almost impossible to prevent pathlines from crossing each other. On the other hand, the streaklet generation algorithm described in Chapter 2 produces streamlines that do not overlap. A single image of a collection of such streamlines can be used to represent the flow at an instant of time. Animating a sequence of such images of streamlines, which represent the flow at discrete time intervals and have frame-to-frame coherence, can produce good visualizations of time varying flows.

Line integral convolution (Cabral and Leedom, 1993) is a dense texture based technique for static flow visualization that has been extended to produce various time-varying visualizations of flow data. Forsell and Cohen (1995) produced one such algorithm that convolved textures along pathlines to represent time varying flow fields. Shen and Kao (1997) produced another LIC algorithm that convolved the input texture over time and used the output texture as input for the next frame in the animation. This produced frames with high spatial and temporal correlation. But because of the dense representations produced, Line Integral Convolution is unlikely to be suitable for showing two overlapping flow layers simultaneously.

Jobard and Lefer (2000) extended their algorithm for 2D evenly spaced streamline generation (Jobard and Lefer, 1997), to visualize time-varying 2D flow fields. Their approach was to compute a well spaced set of streamlines for the first frame. In the next frame they kept the streamlines that corresponded best with those in the previous frame and discarded the others. Then they filled in any gaps. This process was repeated for all subsequent frames. Streamlines were also mapped with moving textures to give a sense of direction during the visualization.

This chapter presents an algorithm similar to that of Jobard and Lefer (2000) that makes use of perceptually optimized streamlines to represent 2D time varying flow fields. The goal is to generate an algorithm that achieves spatial and temporal coherence between the successive frames of the animation, with each frame being represented by a collection of streamlines that are perceptually optimized. The important features of this algorithm are:

1. A two pass algorithm to achieve frame-to-frame coherence by building a list of the streaklets (which includes the starting position, frames of birth and death) drawn during all the frames in the animation.
2. Generation of individual frames in the animation using perceptually optimized streaklets. The streaklet generation algorithm, used for the perceptual study, is used to map these streamlines onto the display.

3. Animation of the texture mapped onto the streaklets to give an overall sense of direction of flow during the animation.

The rest of this chapter is organized as follows. The next section discusses the two pass algorithm for time varying flow visualization. The algorithm is then used for the time varying visualization of a real-time two layered 2D flow data after applying the results obtained from the perceptual study for visualizing static flow fields.

3.1 <u>Algorithm for time varying flow visualization:</u>

The algorithm for visualizing time varying layered flow fields is designed to extend the algorithm given in Chapter 2 to

1. Allow for all of the parameter settings discovered from the experiment described in Chapter 2 to be applied to a time varying two-layered flow.
2. Maintain the various parameters of the streaklets (width, spacing etc) as much as possible from one frame to the next.
3. Provide frame-to-frame coherence for the streaklets in the animation. In general streaklets should last as long as possible through the course of the animation.
4. Kill the streaklets that do not live long enough, in order to remove the sparkling effect of the rapid birth and death of the streaklets during the animation.
5. Handle the death of streaklets by allowing for the birth of new streaklets to fill any gaps in the screen.
6. Generate frames that can be played back to view the animation.

The input to the algorithm can either be a function that has a time variable or a numerical model or a stored grid of vectors for each time interval. So all that is required for input is a function

v = **GetFlowVector(**p,t**)**

that returns a flow vector v for a location p at time t.

The method formulated is a two-pass procedure. In the first frame of the first pass, the algorithm developed in Chapter 2 is applied to produce a set of streaklets. On subsequent frames streaklets are redrawn, but discarded if they are too short or if there is no space for them. Gaps are filled. The positions of all streaklets are stored in a list throughout this process. In the second pass, this list is used to delete all those streaklets that do not live long enough, and, then the list is sequentially traversed to draw all those streaklets that are alive in the particular frame being drawn. This algorithm makes use of all the data structures and procedures illustrated for streaklet generation in Chapter 2. The new data structure specific to this algorithm is the list of *streakletStatistics* which is a structure consisting of the streaklet *startPosition*, *birthFrameNumber* and *deathFrameNumber*.

3.1.1 First pass:

The first pass of the algorithm is used to build a list of *streakletStatistics*. In the first frame, streaklets are drawn in the screen using the streaklet generation algorithm and each new streaklet is added at the end of the list of *streakletStatistics*. In the subsequent frames of the animation, this list is sequentially traversed to get the streaklets. If a streaklet's *startPosition* is already filled in the *spacingBuffer*, it is considered to be dead and its *deathFrameNumber* is updated to the number of the current frame being drawn. The empty spaces in the screen, caused due to the death of streaklets, are filled by drawing new streaklets and adding them at the end of the list of *streakletStatistics*.

A pseudo code description of the first pass follows:

animationFrameNumber = 1:

DrawFlowPattern(*spacing,width****)***

{

while (*spacingBuffer* unfilled)

{

1. *startPosition* **= *findEmptyPixel()***
2. *success* = ***traceForwardandBackward***(*startPosition,time, width,spacing*)
3. if (*success*)

   Add the newly born streaklet to the end of the list of *streakletStatistics*. The *birthFrameNumber* of this streaklet is set to the current frame number and the *deathFrameNumber* is set to the maximum number of frames in the animation.

}

}

for (*animationFrameNumber* > 1)

{

for (each alive *streaklet* in the list *streakletStatistics* (whose *deathFrameNumber* >= *animationFrameNumber*))

{

```
        startPosition = streaklet.startPosition
        if (startPosition already filled in spacingBuffer)
        {
                /* Death of the streaklet */
                streaklet. deathFrameNumber = animationFrameNumber
        }
        else
        {
                success = traceForwardandBackward(startPosition,time,
                width,spacing)
                if (not success) /*Death of the streaklet*/
                        streaklet. deathFrameNumber = animationFrameNumber
        }
}
while (spacingBuffer unfilled) /* fill any empty gaps */
{
```

1. *startPosition* = ***findEmptyPixel()***
2. *success* = ***traceForwardAndBackward*** (*startPosition, time, spacing, width*)
3. if (*success*)

    Add the newly born streaklet to the end of the list of *streakletStatistics*. The *birthFrameNumber* of this streaklet is set to the current frame number and the *deathFrameNumber* is set to the maximum number of frames in the animation.

}

animationFrameNumber = *animationFrameNumber* + 1

}

traceForwardAndBackward(*startPosition, time, spacing, width*)

{

1. Start a streak at *startPosition* and ***traceForwards***(*startPosition, time*)
2. Trace the streak backward from the *startPosition* in the same fashion
3. Draw the streak into the *spacingBuffer* at spacing times the width
4. if *streakLength* < *lengthThreshold*
    a. remove streak from *spacingBuffer*
    b. return false

    else

    a. Draw the streak into the *drawBuffer* using the correct width
    b. return true

}

traceForwards(*startPosition, time****)***

{

p = startPosition

repeat

set the middle of the *streakArray* to *startPosition*

v = ***GetFlowVector(****p,time****)***

p = p+v;

store *p* in next position in the *streakArray*

until***(****p* falls on a white pixel ***OR*** *v* **<** *threshold* ***OR*** *p* outside *viewport*)

}

3.1.2 Second pass:

In the second pass of the algorithm, all the streaklets in the list of *streakletStatistics* that do not live long enough are deleted from the list. The animation is then rendered frame-by-frame, drawing all remaining streaklets in each frame. Each frame drawn is stored in a format suitable for playback.

The pseudo code of the second pass is shown below:

DeleteShortLivingStreaklets()

{

for (each *streaklet* in the list *streakletStatistics*)

{

if (*streaklet.deathFrameNumber* – *streaklet.birthFrameNumber* < *MinimumLifetime*)

delete *streaklet* from *streakletStatistics*

}

}

for (each animation frame)

{

DeleteShortLivingStreaklets()

```
for (each streaklet in the list streakletStatistics)
{
    if streaklet.endFrameNumber < animationFrameNumber
        /* Death of streaklet */
        delete dead streaklet from list streakletStatistics
    else
        traceForwardAndBackward (streaklet.startPosition, time, width,
        spacing)
}
StoreFrame()
}
```

3.2 Time varying visualization of real-time two layered 2D flow data:

The time varying algorithm was used to visualize flow patterns, generated by NOAA, from the Chesapeake Bay Operational Forecast System (CBOFS). This model contains 10 flow layers. Two of these layers were selected and the visualization was applied to them. One of the best sets of parameters obtained from the perceptual study is applied to the two layers. A subset of the data in a grid of size 416 by 240 was used as an input for the algorithm. Screenshots taken from the animation of 200 frames of the flow is shown in Figure 12.

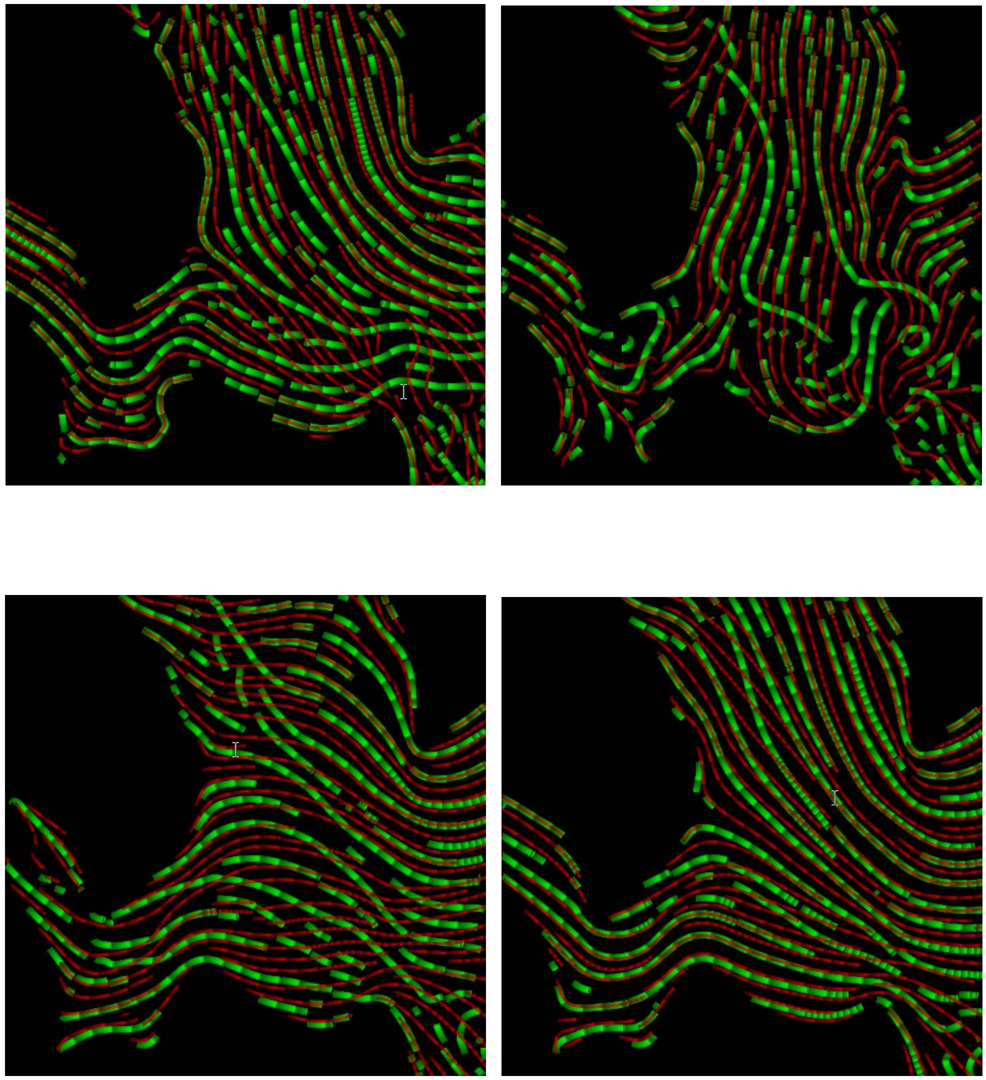

Figure 12: Time varying visualization of two layers from the CBOFS model

3.3 <u>Conclusion:</u>

The algorithm for visualizing time-varying flow fields discussed in this chapter operates at the streaklet level to achieve frame-to-frame coherence, similar to the algorithm developed by Jobard and Lefer (2000). All the perceptual parameters that were studied in Chapter 2 are maintained as much as possible during the entire sequence of the animation. These perceptual parameters that were optimized for the visualization of static flow fields do not include parameters that are only relevant to the visualization of time-varying flow fields. For example, visualizing time-varying flows include other additional perceptual variables such as the frequency of display of the subsequent frames of the animation. So it might not be the case that the animated flow visualizations are perceptually optimal to the same extent, although it is likely that the optimal parameters for displaying moving and static patterns would be similar.

CHAPTER 4

CONCLUSION

This thesis introduces a new method for perceptually optimized visualization of two layered flow fields. The problem of flow visualization is considered to be a problem of perceptually optimizing the visual properties of a set of streaklets used to represent the flow field in the two layers. These streaklets can vary in a number of parameters like size, shape, color etc. Using traditional psychophysical methods to optimize these streaklet parameters is complicated and requires thousands of hours of participation by each subject. This necessitated a development of a new approach to the problem of producing perceptually optimized solutions for two-layered flow visualization.

The perceptual optimization process is the heart of this thesis. The method adopted was to encode the visualization problem in the form of a vector of parameters that determines a search space within which lie all possible display solutions to the problem of visualizing two-layered flow fields. The dimension of this search space is the number of streaklet parameters to be optimized. From the review of the literature concerning human perception of contours the perceptual parameters that were most relevant to the problem of displaying layered flow fields were chosen. If the parameters take on values over a continuous range of real numbers, this parameter space will be infinite. So parameter values were chosen that covered an interesting range of values.

Thus, it is concluded that the perceptual optimization process developed is, out of necessity, approximate.

The process started by defining a parameter space of 3456 display solutions in which to search for the best solutions for visualizing two-layered flow fields. A streaklet generation algorithm was developed that mapped the parameters to produce unique visual representations of two-layered flow fields. An experiment was designed to show all 3456 different visual representations of the same two-layered flow fields to each subject in a random order. The experiment was conducted on a display with a resolution that is close to the limits of the resolution of the human eye. Subjects had to rate the different flow field solutions on a scale from 0 to 9 for both the foreground and background flow field layers. These ratings were stored for later analysis.

The analysis of the data collected from the perceptual optimization process produced a number of insights on the design of good display solutions for visualizing two layered flow data. Display solutions that have different foreground and background streak colors are found to be better than solutions with gray scale foreground and background streaks. A difference in width of the streaks used to represent the flow in the foreground and background layers is useful in differentiating the two layers distinctly. Most of the top solutions have wider background streaks and narrower foreground streaks. Most of the highly rated solutions also had a large spacing between streaks in the foreground and background layers. However, such solutions produce sparse representations of the flow in the background and foreground layers and do not show much detail. Perfectly opaque streaks in the foreground occlude the background flow pattern and so some amount of transparency is required in the foreground layer to visualize the background layer as well.

Finally, the streaklet generation algorithm developed for the perceptual optimization process was extended to the visualization of time-varying layered flow fields. The best display parameter values obtained from the perceptual study were used in the time-varying algorithm. Also, animated textures were used along the streaklets to give a sense of direction of the flow with time. The algorithm achieves spatial and temporal coherence by correlating the streaklets between successive frames.

4.1 Future work:

The perceptual optimization process developed is designed to obtain a set of guidelines for producing good solutions to the visualization of two-layered flow fields. This process incorporates a subjective evaluation of the visualization problem. It has been assumed that subjective assessments will have a strong relationship to actual performance. But to be sure of this, objective tests would have to be devised. For example, tests might be devised to measure the performance of subjects on the detection of critical points and advection of particles in a flow (Laidlaw et al, 2001).

Another limitation of the perceptual optimization process is that only two flow patterns were used to represent the layered flow in the experimental study. So the output of the optimization process is a function of these flow patterns. The results obtained might not produce perceptually optimized visualizations when used for different layered flows. Therefore, a study that makes use of more than two flow patterns can provide more general results for visualizing layered flows.

REFERENCES

Cabral, B., and Leedom, L.C. (1993). Imaging vector fields using line integral convolution. *SIGGRAPH'93 Proceedings*, 263-270

Cook, R.L. (1986). Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 51-72

Cumming, B.G., Johnston, E.B. and Parker, A.J. (1993). Effects of different texture cues on curved surfaces viewed stereoscopically. *Vision Research*, 33(5/6): 827-838

Field, D.J., Hayes, A., and Hess, R.F. (1993). Contour integration by the human visual system: Evidence for a local "association field". *Vision Research*, 33(3) 173-193

Forsell, L.K. and Cohen, S.D. (1995). Using line integral convolution for flow visualization: curvilinear grids, variable speed animation and unsteady flows. IEEE Transactions on Visualization and Computer Graphics, 133-141

Fowler, D. and Ware, C. (1989). Strokes for representing univariate vector field maps. *Graphics Interface '89 Proceedings*, 249-253

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.)

House, D., and Ware, C. (2002). A method for the perceptual optimization of complex visualizations. *Advanced Visual Interface*, Trento Italy, 148-155

Jobard, B., and Lefer, W. (1997). Creating evenly-spaced streamlines of arbitrary density. *Proceedings of the Eurographics Workshop*. Boulogne-sur-Mer, France

Jobard, B. and Lefer, W. (2000). Unsteady flow visualization by animating evenly spaced streamlines. Computer Graphics forum

Klassen, R.V. and Harrington, S.J. (1991). Shadowed hedgehogs: A technique for visualizing 2D slices of 3D vector fields. *IEEE Visualization*, 148-153

Koffka, K. (1935). *Principles of Gestalt Psychology*. Harcourt-Brace, New York

Laidlaw, D.H., Davidson, J.S., Miller, T.S., Silva, M., Kirby, R.M., Warren, W.H., and Tarr M. (2001). Quantitative comparative evaluation of 2D vector field visualization methods. *IEEE Visualization*, 143-150

Livingstone, M.S. and Hubel, D.H. (1988). Segregation of form, color, movement and depth: Anatomy, physiology and perception. *Science*, 240: 740-749

Shen, H.W. and Kao, D.L. (1997). UFLIC: a line integral convolution algorithm for visualizing unsteady flows. Proceedings of IEEE visualization, 317-322

Triesman, A. and Gormican, S. (1988). Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review* 95(1): 15-48

Turks, G. and Banks, D. (1996). Image-guided streamline placement. *SIGGRAPH'96 Proceedings*, ASM, 453-460

Van Wijk, J.J. (1991). Spot noise texture synthesis for data visualization. *SIGGRAPH'91 Proceedings*, 309-318

Ware, C. (2004). *Information Visualization: Perception for Design. Morgan Kaufman*, New York, 2004

Ware, C. and Knight, W. (1995). Using visual texture for information display. *ACM Transactions on Graphics* 14(1): 3-20

Wegenkittl, R. and Grollerm G. (1997). Fast oriented line integral convolution for vector field visualization via the internet. *IEEE Visualization*, 309-316

Wilson, H.R. (1983). Psychophysical evidence for spatial channels. In *Physical and Biological Processing of Images*, ed. O.J. Braddick and A.C. Sleight, 88-89. Springer, Berlin

APPENDIX – IRB APPROVAL DOCUMENTATION