TECHINQUES FOR VISUALIZING MULTI-VALUED FLOW DATA

A PROJECT COMPLETED AT THE UNIVERSITY OF MINNESOTA

BY

Timothy Matthew Urness

Victoria Interrante
advisor

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

MAY 2003

Abstract

We present several techniques to effectively visualize multi-valued flow data using contrast, color, 3D visualization, and texture. These methods were developed to facilitate the effective simultaneous visualization of multiple derived quantities in experimentally acquired, stereo PIV data of wall-bounded turbulent flow at moderately high Reynolds numbers. Our ultimate goal through this work is to enable researchers to obtain a succinct, meaningful visual summary of the contents of a dataset through providing techniques that allow the creation of images in which the important features of multiple scalar distributions can be understood both independently and in the context of multiple other distributions.

1. INTRODUCTION & MOTIVATION

Researchers in the turbulence community have long been interested in developing a deeper understanding of the key physical mechanisms in turbulent flows. Of particular interest is the question of how eddies contribute to drag, which, if understood, could enable the development of strategies to control eddy generation, scale and/or organization in order to reduce drag. This would have a major impact on many industries including aerospace, transportation, energy, and chemical processing. For example, reductions in turbulent skin-friction drag over aircraft would result in significant decreases in fuel usage – lowering fuel costs, increasing aircraft range, and reducing pollutant and greenhouse gas emissions. Efforts to achieve a fundamental understanding of turbulent flows remain limited mainly because of a lack of understanding of the non-linear interactions that occur between the vortices that make up the motion. Since it is not computationally feasible to numerically simulate a wide range of eddy scales, researchers who are interested in probing these questions must rely on experimental methods to further their understanding.

Stereoscopic *particle image velocimetry* (PIV) is a technique that can be used to experimentally measure instantaneous components of a velocity field in a plane of a turbulent boundary layer in a moderate to high Reynolds number flow. Along with the experimentally generated vector field, values of vorticity, Reynolds shear stress, and swirl strength can be mathematically derived and are important in characterizing potential 'regions of interest'. In turbulent boundary layers, various theories have indicated that hairpin shaped vortices cause drag by producing Reynolds shear stress, and that this process may be enhanced when multiple hairpins travel together with similar speeds as a packet. In streamwise-spanwise planes parallel to the boundary layer surface, the packets can be characterized by zones of uniform but low streamwise velocity containing areas of high negative Reynolds shear stress and falling between cores of strong positive and negative vorticity [1]. Because the process of knowledge discovery related to this application is predicated on the ability to achieve an integrated understanding of the individual contributions of each variable and of how the variables inter-relate with each other, developing effective multivariate visualization methods is of critical importance to facilitating the understanding and analysis of results from the PIV experiments.

In this paper, we describe several techniques that have been developed for the purposes of effectively visualizing multi-valued flow data. We utilize methods of visualization based on line integral convolution (LIC), which provide a high-resolution output texture and begin to explore the potential of representing multi-valued flow data. We also explain in detail the algorithm developed to detect the 'regions of interest' or vortex packets in turbulent boundary layers and explain and motivate future areas of research.

All of the approaches were applied to the visualization of PIV data obtained in a wind tunnel turbulent boundary layer at Reynolds number $R_\theta = 2500$ ($Re_\tau = \delta U_\tau/\nu = 1060$) at a wall-normal location nominally in the logarithmic region ($z^+ = zU_\tau/\nu = 98$). Here z is the distance normal to the wall, δ is the boundary layer thickness, U_τ is the wall shear velocity, and ν is the kinematic viscosity of the fluid. Unless mentioned otherwise, the vector fields used to generate the LIC textures were obtained by subtracting the mean streamwise velocity from the in-plane velocity components and disregarding the out-of-plane velocity components. Swirl is characterized by the quantity $\lambda_{ci}\omega_z/|\omega_z|$, where λ_{ci} is the magnitude of the imaginary part of the eigenvalue of the local velocity gradient tensor and ω_z is the in-plane vorticity. The term *frame* refers to a single instance of the flow data.

2. PREVIOUS WORK

2.1 Texture-Based Flow Visualization

Textures have traditionally been a popular and effective method for representing vector and scalar fields. In pioneering work, van Wijk [2] introduced the concept of 'spot noise', a texture constructed from weighted and randomly positioned spots deformed in accordance with the direction of flow. Cabral and Leedom [3] shortly afterward introduced *line integral convolution* (LIC), a versatile and widely-used technique in which intensities in an input texture are convolved along streamlines defined by an accompanying vector field to produce a texture that is highly correlated in the flow direction. Stalling and Hege [4], in the Fast-LIC algorithm, achieved an order of magnitude increase in the efficiency of the LIC algorithm by taking advantage of coherence along streamlines and increased the fidelity of the resulting output image by using a fourth-order Runge-Kutta method for streamline calculations. This results in the computation of the output texture being streamline oriented, not pixel oriented.

In traditional LIC images, the direction of movement in a flow is ambiguous, and animation is required to make that information explicit. However Wegenkittl, Gröller, and Purgathofer [5] introduced a technique called Oriented Line Integral Convolution (OLIC) that addresses this issue. In essence, the OLIC algorithm works by taking as input a sparse texture resembling ink droplets on a page and using a ramp-like convolution kernel to smear the droplets according to the vector field, resulting in a collection of streaks in which intensity increases from tail to head. Computation time for this method was significantly reduced with the introduction of Fast Oriented Line Integral Convolution (FROLIC) [6]. More recently, in another approach similar to OLIC, Sanna *et. al* [7] propose a Thick Oriented Stream Lines (TOSL) method, in which the orientation of a flow is depicted by increasing the luminance along calculated streamlines. This method is advantageous because it allows for dense vector fields to be represented since as each streamline is considered separately.

Shen, Johnson, and Ma [8] added color to LIC images through the use of simulated dye advection.

Multiple frequency input textures were first used with LIC by Kiu and Banks [9] to incorporate indications of velocity magnitude. In this method, a single input texture is constructed from a number of different frequency components, each assigned to discrete regions defined by specific intervals of velocity magnitude, and the LIC algorithm is run on this input. In their implementation, the kernel filter length used in defining the output intensity at each point is also varied in proportion to the magnitude of the velocity of the flow at the corresponding point. The results that follow are longer and thinner lines in regions where the flow is fast, and shorter and thicker lines where the flow is slow.

Ware and Knight [10] proposed the use of Gabor functions to create texture-like images of flow data in which information is encoded along the perceptually significant texture dimensions of scale, orientation, and contrast.

Research has recently been done on representing scalar values with texture-based vector field representations such as LIC through the use of bump mapping [11] and contrast enhancement [12]. These techniques essentially enable the visualization of additional scalar values without requiring the use of color.

2.2 Multivariate Visualization

Healey and Enns [13] have contributed methods to use texture elements on underlying 3D height fields to visualize multivariate data. They combine texture dimensions of height, density, and regularity along with perceptually uniform colors to increase the number of attributes that can be simultaneously represented.

Weigle *et. al* [14] propose a texture generation technique, based on the layering of patches of oriented slivers, which uses orientation and luminance to encode information about multiple overlapping scalar fields.

Laidlaw et. al [15] showed how shape, orientation, and color attributes of ellipsoids could be used to represent multivariate components in diffusion tensor images of the mouse spinal cord. Their method was inspired by the brushing and layering techniques used in oil painting. Similarly inspired by concepts from painting, Kirby, Marmanis, and Laidlaw [16] showed how different sized icons, color, elongated ellipses, and layering could be used to portray multivariate data from 2D compressible flows.

2.3 Introduction to Line Integral Convolution

Throughout the paper, we utilize methods of visualization based on line integral convolution (LIC) which provides a high-resolution output texture depicting flow information. To ensure all readers are familiar with the technique, we provide a brief explanation of the Stalling and Hege [4] algorithm we have implemented.

We first assume we have an input texture, vector field, and a kernel length (denoted as ‘N’) as input parameters. Streamlines are calculated from the points on the vector field grid, where a fourth-order Runge-Kutta scheme determines incremental positions along the calculated streamline. Adaptive step sizes are employed to speed up the computation, while error monitoring limits step sizes to ensure high accuracy. Once the streamlines were calculated, they are re-sampled at equally spaced points for use in sampling the input texture. Each pixel in the final image was determined by averaging 2N pixels of the input texture along the streamline section centered on that pixel. The resulting image has texture patterns depicting the streamline orientation throughout the field.

3. REGION DETECTION

Our first task was to accurately and automatically detect the size and shape of hairpin vorticity packets in the experimentally acquired PIV data. Once developed, the algorithm allowed for the automatic detection of frames that would reveal the structures consistent with hairpin vorticity packets since many of the frames do not contain these ‘regions of interest’. These regions are characterized by elongated positive vorticity regions aligned directly below elongated negative vorticity regions with points of high Reynolds shear stress and low streamwise velocity between the regions of extreme vorticity. As this technique has already been documented [1], we present a short summary of the detection algorithm and a picture of the visualization techniques used.

First, regions that exceed a threshold value of high positive vorticity or low negative vorticity are detected. Once the significant patches of vorticity are determined, areas where elongated positive vorticity regions are aligned directly below elongated negative vorticity regions are identified and marked as this fits the pattern of legs of a hairpin vortex. Within these areas, points that have a high Reynolds shear stress are used as seed points for a region-growing algorithm. The regions are grown based on low streamwise vector velocity, which results in the final coherent regions representing patches of uniform momentum.

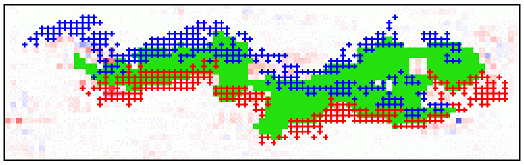


Figure 1: A vorticity packet detected by the region detection algorithm. The algorithm detects elongated regions of positive vorticity (red crosses) below elongated regions of positive vorticity (blue crosses) with points of high Reynolds shear stress and low streamwise velocity (green) between the areas of extreme vorticity.

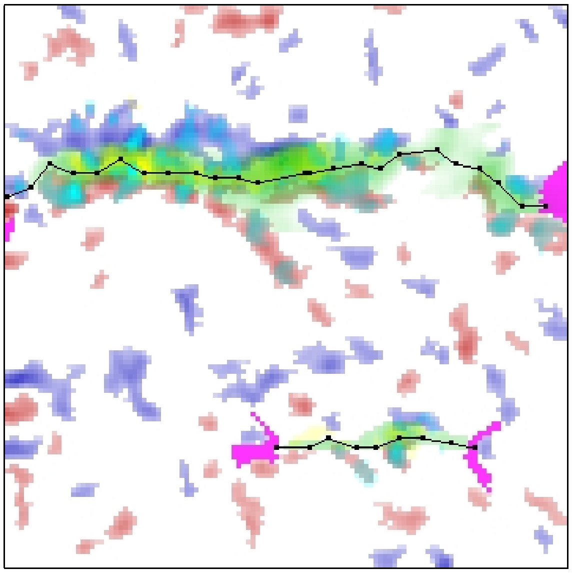


Figure 2: An image used to show components of the region detection algorithm. Color is used to represent different variables and is applied in the following order: negative (blue) and positive (red) vorticity, Reynolds shear stress (yellow), swirl (cyan). A transparent green is used to highlight low streamwise velocity. The centerlines (black) of the detected region are approximated and the pink regions denote areas inside the 45-degree angle that have been scanned for related connecting regions.

Separate regions detected may be close in proximity to one another and highly correlated. An additional step is added to determine if neighboring areas of uniform momentum should be connected or considered related. Once the individual patches of uniform momentum are found and identified, they are run through a proximity algorithm that has two parts. First, regions that are directly adjacent or are connected by regions of high Reynolds shear stress are joined. Second, the centerlines of the regions are approximated and the end of each region is scanned within a 45-degree angle with respect to the projected centerline. Patches that are found in the scanned area are not joined, but outputted as being in a reasonable or potential joining area.

4. ISSUES IN FLOW VISUALIZATION

Images similar to figure 3 are often used to display flow information through the use of glyphs, such as arrows, to indicate the direction of flow at a discrete set of points. The length of the arrow may be used to represent the velocity magnitude. Although effective, a drawback of such a scheme is that the vector field is often down-sampled resulting in course spatial resolution as each arrow will require several pixels to be drawn. Even when the vector field is not sampled, the collection of glyphs does not easily lend itself to the perception of global fluid flow as the line segments must be perceptually projected and connected in order to represent the potential flow of a particle. Samples that are selected along a uniform grid create artifacts in which the structure of the grid interferes with correct perception of the direction indicated by the vectors [17].

Texture-based visualization methods produce high-resolution output images that allow structures of the flow to be perceived more easily than with the field-of-arrows technique (figure 4). An additional scalar variable may be represented through the use of a color encoded background underlying the array of vector glyphs or as a color overlay in the texture-based approach. Later in this paper, we investigate different techniques of using texture and color to represent variables in multi-valued flow data.

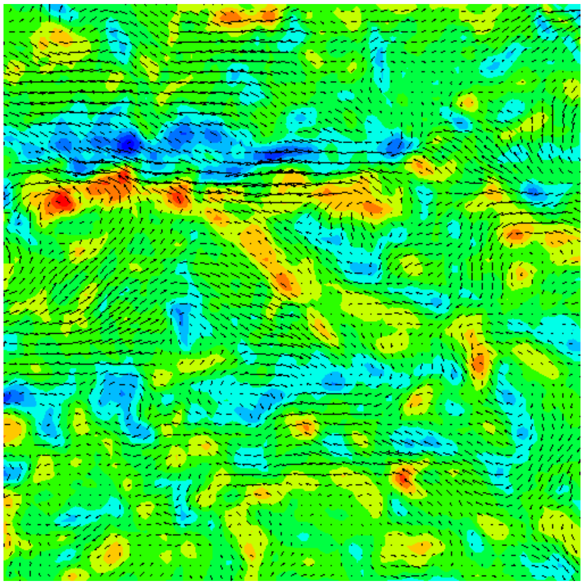


Figure 3: A traditional field-of-arrows image of an experimentally acquired PIV dataset in which arrow glyphs are used to represent flow direction. The size of the glyph represents velocity magnitude, and vorticity is represented with color.

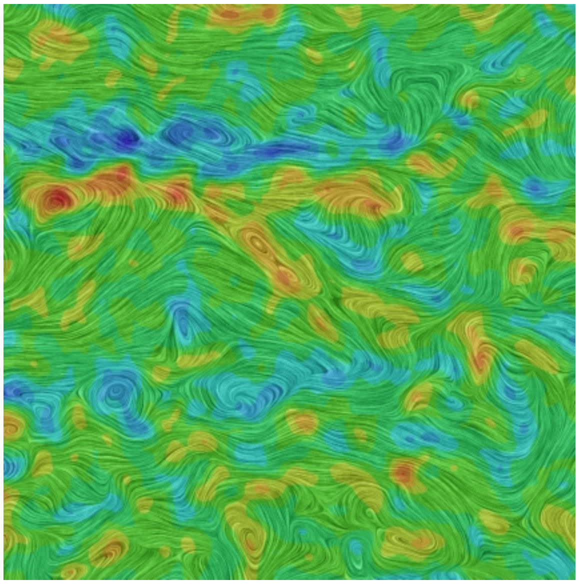


Figure 4: A texture-based approach is used to represent the same data as in figure 2. Advantages of texture-based visualizations over field-of-arrow techniques include higher resolution and a continuous representation of flow. A color overlay is used to depict vorticity.

The vector field in any visualization is dependant on the relative velocity of the observer. Traditionally, the average value of the streamwise component of the global vector field is calculated and subtracted from each vector. However, the resulting LIC image, critical points, and vector field features are greatly influenced by the magnitude of the value subtracted from the streamwise velocity. An example of this phenomenon is shown in figure 5. In these images, LIC is applied to the experimentally acquired PIV data and superimposed in color is the swirl strength (λ_{ci}), which corresponds to vortex cores. In the leftmost image, the swirling eddies in the bottom left of the flow field are only apparent because we have chosen a frame convecting at a speed that matches the convection velocity of those eddies. Swirl strength, however, is Galilean invariant and does not vary with the magnitude subtracted from the streamwise velocity as can be seen when a different magnitude is subtracted in the rightmost image.

The complete animation can be viewed at *www-users.cs.umn.edu/~urness/download/newflow8.mpg.* An interactive java applet can be viewed at *www.dtc.umn.edu/~urness/licapplet/licapplet.html*

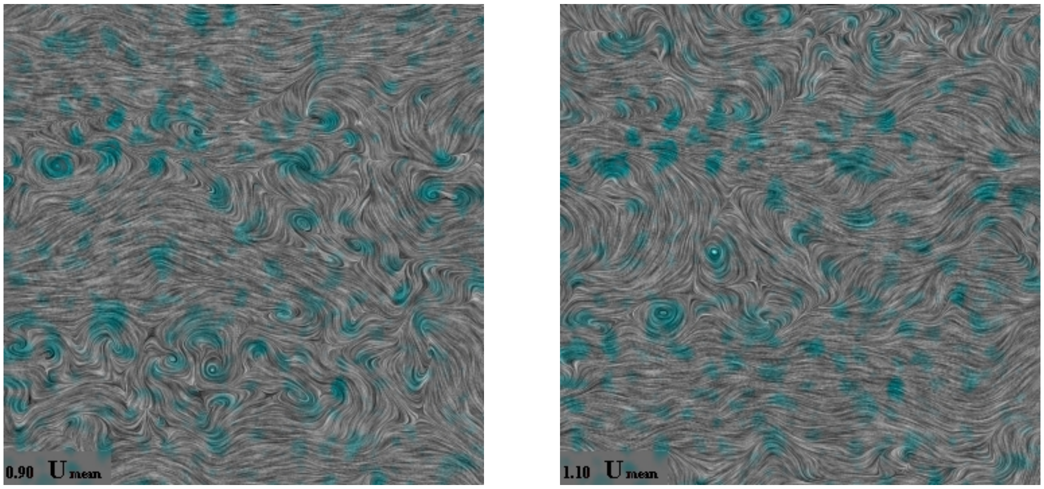
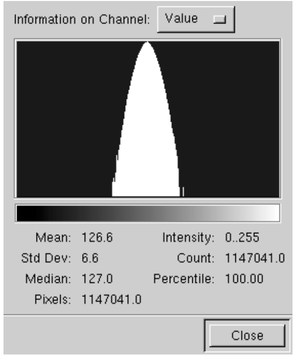


Figure 5: LIC images with swirl strength (cyan) superimposed. The flow is from left to right. The value shown in the lower left hand corner has been subtracted from the streamwise velocity.

5. CONTRAST & LUMINANCE

The role of the luminance component has a prominent effect on how features in an image are perceived [18]. Manipulations of mean luminance or contrast have the ability to enhance characteristics of an image with the intent of representing a scalar value in addition to the flow data already depicted by LIC. There are many ways in which contrast or luminance has been used to display information in texture-based flow visualization [5, 6, 7, 10, 11, 12]. We illustrate two such methods to represent a scalar variable to the flow field: mean luminance and contrast, and also briefly discuss how it relates to a technique that depicts the direction of flow using a luminance ramp along calculated streamlines.


Information on Channel: Value

Mean: 126.6 Intensity: 0..255
Std Dev: 6.6 Count: 1147041.0
Median: 127.0 Percentile: 100.00
Pixels: 1147041.0

Figure 6: Histogram of a typical LIC image. Extreme values are not represented in the histogram due to the convolution process that averages input samples along the calculated streamline.

5.1 Mean Luminance

Mean luminance refers to the average intensity value of the pixels in a given region and can be generally characterized as the overall brightness of a region. The default grey-level values of a LIC image generated from a random white-noise input texture typically have an average value close to 127 and a histogram that resembles figure 6. If the average luminance of the image is changed, the prominence of a scalar value can be encoded in a post-process manner by scaling the pixel values according to the following formula.

$$I_{new}(x,y) = constant * scalar(x,y) + (255-constant)*I_{old}(x,y)$$

The effect of applying this formula essentially shifts the histogram with respect to the scalar value, but maintains the overall shape. Using this technique, the contrast between the black and white values of lines remain the same as in the original image and the average luminance value depicts the scalar value.

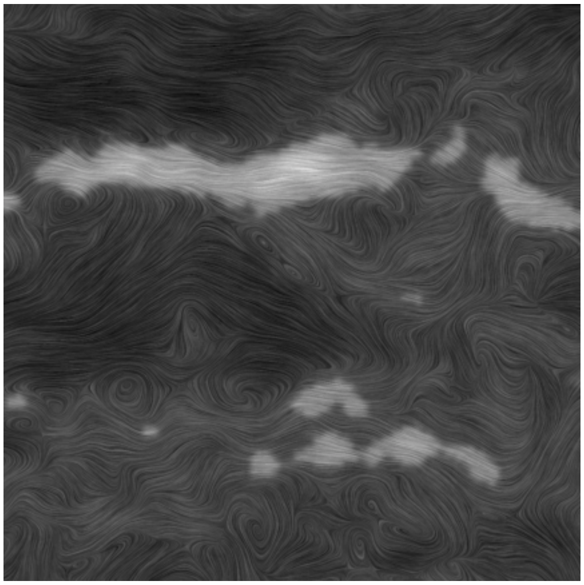
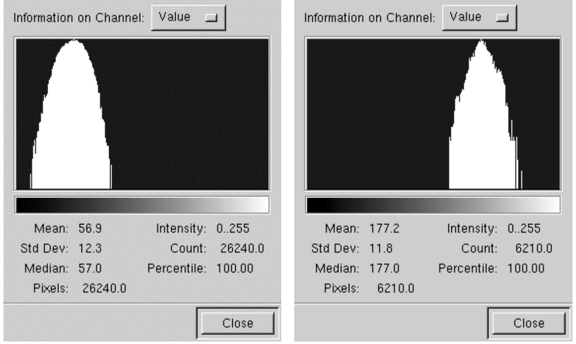


Figure 7: Manipulation of mean luminance to represent a scalar variable. The average luminance value of the default LIC image is shifted according to the values of a scalar field. In this image, mean luminance is used to accentuate regions of uniform momentum.



Information on Channel: Value

```
  Mean:  56.9        Intensity:  0..255
Std Dev:  12.3           Count:   26240.0
 Median:  57.0      Percentile:  100.00
  Pixels:   26240.0
```

Information on Channel: Value

```
  Mean:  177.2       Intensity:  0..255
Std Dev:  11.8           Count:    6210.0
 Median:  177.0     Percentile:  100.00
  Pixels:    6210.0
```

Figure 8: Two histograms taken from different regions of figure 7 showing that the shape of the histogram remains similar, only the average luminance value is changed.

5.2 Contrast

While the average intensity values for an image may remain the same, differences in the contrast between the blacks and whites can be used to effectively convey information about a scalar distribution.

Ranges in contrast are achieved by stretching the image histogram. To create this effect, first the LIC image is calculated and the default grey values are compared against the global average (usually around 127). The new pixel intensity is assigned a linear combination of the scalar value and the old intensity. For example, if a pixel value is above the average, its new intensity is a value close to the maximum only if the scalar value is significant, otherwise it will be close to the default value. Similarly, if the original pixel value is below the average, the new intensity will be minimal only if the scalar value is significant, otherwise it will be close to the default value. The resulting histogram for the image will be different, but the average pixel values for the global image will remain similar to the original LIC image. The varying contrast between the white and black regions is sufficient to depict the scalar value as shown in figure 9.

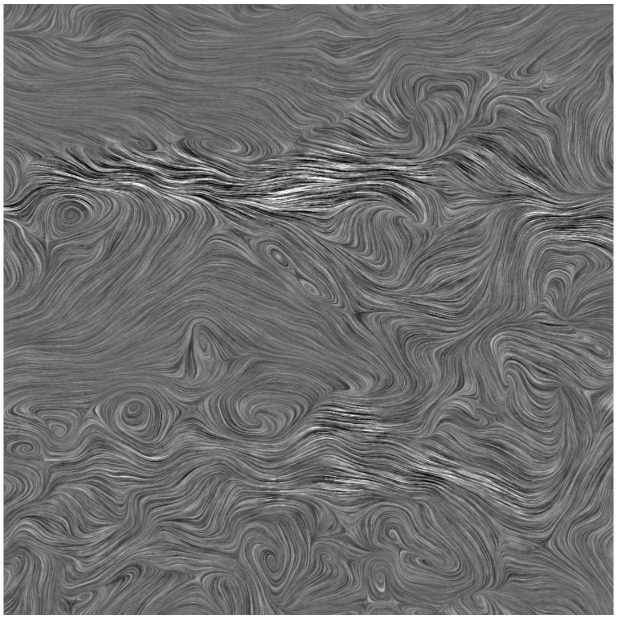
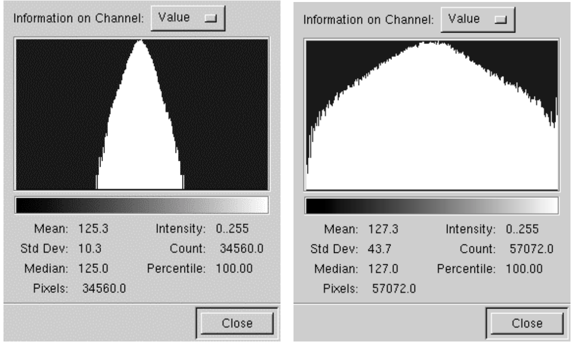


Figure 9: Manipulation of contrast to represent a scalar variable. Local differences between black and white values are used to accentuate regions of uniform momentum.



```
Mean:    125.3      Intensity:  0..255
Std Dev: 10.3           Count:   34560.0
Median:  125.0     Percentile:  100.00
Pixels:   34560.0
```

```
Mean:    127.3      Intensity:  0..255
Std Dev: 43.7           Count:   57072.0
Median:  127.0     Percentile:  100.00
Pixels:   57072.0
```

Figure 10: Two histograms taken from different regions of figure 9. The left image is a histogram of a low-contrast region. The rightmost image is a histogram of a high-contrast region.

While manipulations of luminance and contrast can certainly be used to represent scalar variables, we are interested in investigating how these quantities can be used together and how closely they are correlated. For example, is it possible for mean luminance and contrast to represent two different variables in the same image? This is an area of future research. Also, grey values by default are not perceptually linear. We plan on investigating how to use this phenomenon to allow the perception of these images to be more accurate.

5.3 Luminance Depicting Flow Direction

There are several techniques used to overcome the problem of ambiguous flow direction that occurs in static LIC images [5, 6, 7]. These techniques typically involve using a luminance ramp to disambiguate the direction of the flow along the line or texture orientation instead of a constant box filter used in the computation of LIC. Sanna *et. al* [7] developed a texture-based method named Thick Oriented Stream Lines (TOSL) method, in which the orientation of a flow is depicted by increasing the luminance along calculated streamlines. The advantage of this technique over others that use more feature-based approaches is that TOSL provides a high-resolution image of the vector field.

The first step in the TOSL method, as in the LIC method, is to first numerically calculate streamlines according to the given flow field. The two approaches differ, however, in that the TOSL method does not use an input texture and does not initiate a convolution process. Instead, intensities for pixels along streamlines are incremented according to the local vector magnitude. The suggested initial intensity of the starting pixel along the streamline is selected from a random number between 30 and 120 at the starting point. The algorithm continues by stepping along each pixel calculated in the streamline and assigning an increasing intensity value. When the value reaches 255, the assigned value starts over at 0. The step size is normalized with respect to the maximum velocity on the local streamline. If the vector has high local velocity, the increment in grey tone is proportionally high at that point in the image.

While the TOSL method is effective and very efficient, we believe there can be some small improvements made to enhance the perception of the images it produces. Starting with an initial intensity value between 30 and 120 may cause artifacts due to streamlines having a similar range that all start from a similar point (such as the edge of the domain or a singularity). This problem can be alleviated by allowing the starting value to be randomly assigned anything between 0 and 255. Secondly, incrementing the intensity of pixels with a step size that is directly proportional to the *local* vector velocity does not give an accurate representation of the flow field. The maximum *global* vector magnitude should be used to normalize the step size. Finally, we find it more appropriate to make the step size inversely proportional to the vector velocity magnitude instead of directly related to the velocity magnitude. Our approach results in creating long smooth lines where the flow velocity is at the global maximum. One justification is that a spot smeared out over a period of time will produce a long streak where the flow is faster.

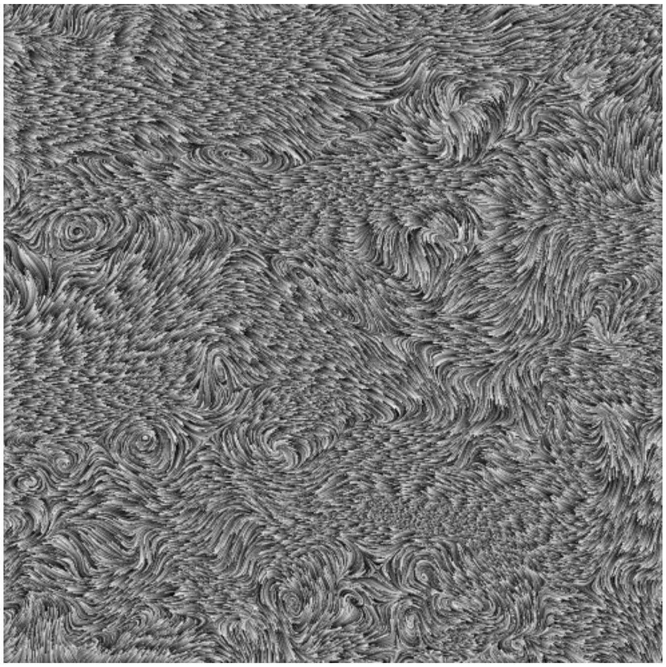


Figure 11: TOSL image of PIV data. This image is subject to artifacts due to the limited range of the starting values, intensity step size being directly proportional to the vector magnitude, and normalizing vectors magnitudes according to the local streamline maximum.

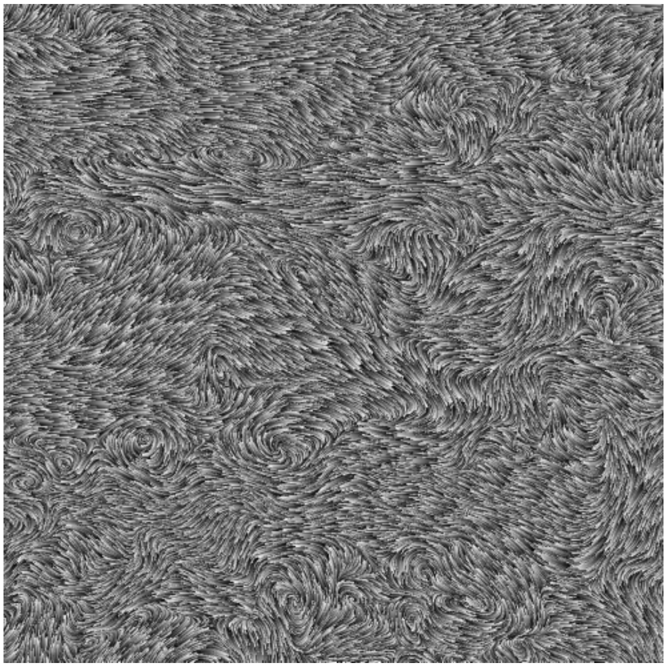


Figure 12: Adjusted TOSL image of PIV data. The artifacts mentioned above are remedied though the use of an unlimited starting range, intensity step size being indirectly proportional to the vector magnitude, and normalizing vector magnitudes according to the global maximum.

While these improvements help the perception of images using this technique, we feel that there are still some issues that need to be addressed. Namely, if the vector field has a stream of vectors that remain correlated in direction and magnitude throughout the image, they will continue to have the same pattern repeated over regardless of the range of the initial starting conditions. Another issue we would like to address is that of contrast. How can we alter the contrast to make the image more aesthetic, and what would the effect be of a more perceptually conscious ramping function instead of a simple linear incrementing scale? These questions will be addressed in future work.

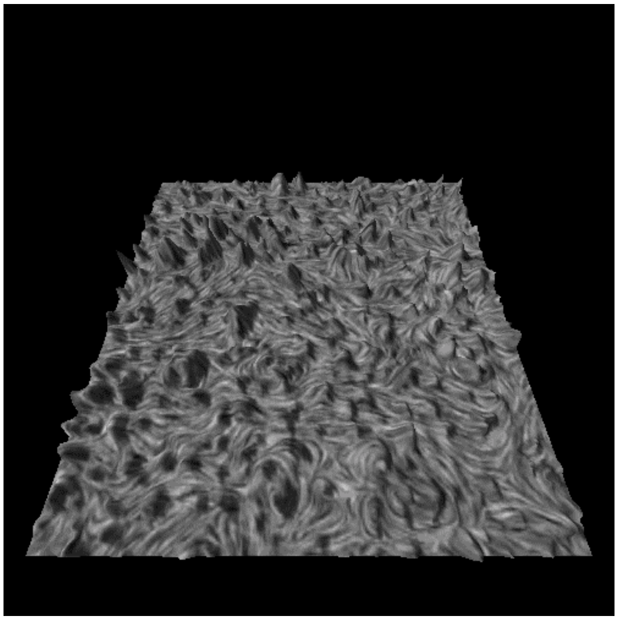


Figure 13: LIC field using 3D visualization where height indicates swirl strength.

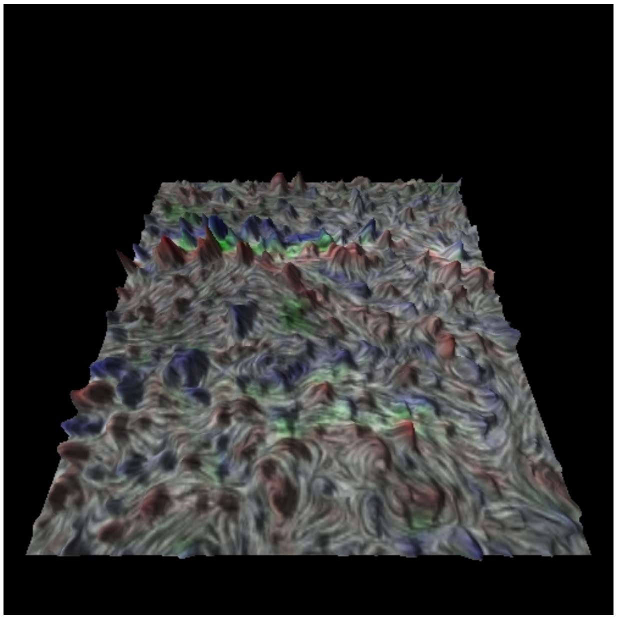


Figure 14: LIC field using 3D visualization. Height indicates swirl strength, contrast represents low momentum zones, red and blue indicate positive and negative in-plane vorticity respectively, and green represent instantaneous Reynolds shear stress.

6. 3D VISUALIZATION

The challenge to represent multiple variables that coexist has inspired us to explore 3D visualization to represent an additional parameter. Figure 13 shows a LIC field that has been texture mapped onto a three dimensional terrain in which height corresponds to swirl values. Swirl values are linearly mapped to the height of the peaks displayed.

With the addition of color overlays as seen in Figure 14, the use of height effectively highlights the existence of large swirl peaks surrounding the zones of low streamwise momentum (brightness) characterized by significant Reynolds shear stress (green), and in between regions of positive (red) and negative (blue) vorticity. However, it is possible that other features are being obscured, as the information is very dense and complicated. The shading that occurs from the height map may result in a misrepresentation of the patterns that occur naturally on the texture. Additionally, there may be confusion due to distortion of the texture patterns as the planar LIC image is projected onto the height-varying surface. Thus, the information perceived from the texture or from the height may be ambiguous. Future research in this area is needed to explore ways to effectively use height in a manner as to not produce spurious results from the perception of other components of the image.

Animation solves a few of the ambiguity problems because of the cues obtained from the motions of the texture and flight path. This animation can be viewed at *www.cs.umn.edu/~urness/download/combo_mov4.com*.

7. COLOR WEAVING

With few exceptions, the use of color with LIC has traditionally been limited to the simplest of color compositing operations in which a LIC texture image is in effect overlaid with a single continuous semitransparent color wash image, with the resulting effect that blacks are left black and the whites are shifted toward the specified hue at each point. While effective for conveying a single scalar distribution in the context of the flow, this post-process method does not allow for the effective simultaneous representation of multiple scalar fields, due to the perceptual difficulty of and inherent ambiguity in color decomposition (figure 15).

As an alternative, we propose a technique in which multiple colors are allowed to coexist on neighboring streamlines, resulting in multicolored images that resemble a tapestry woven with different colored threads (figure 16).

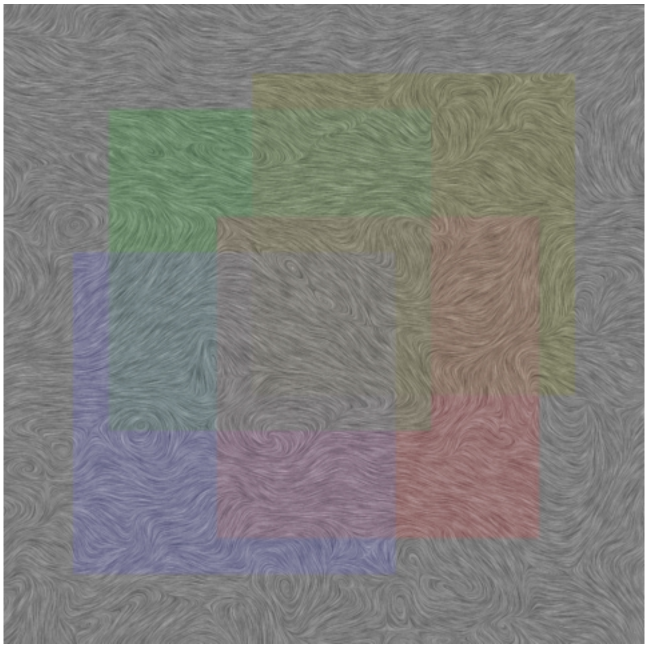


Figure 15: Four artificially defined, mutually overlapping regions, overlaid on a LIC image. The color combinations are obtained by averaging in RGB color space.

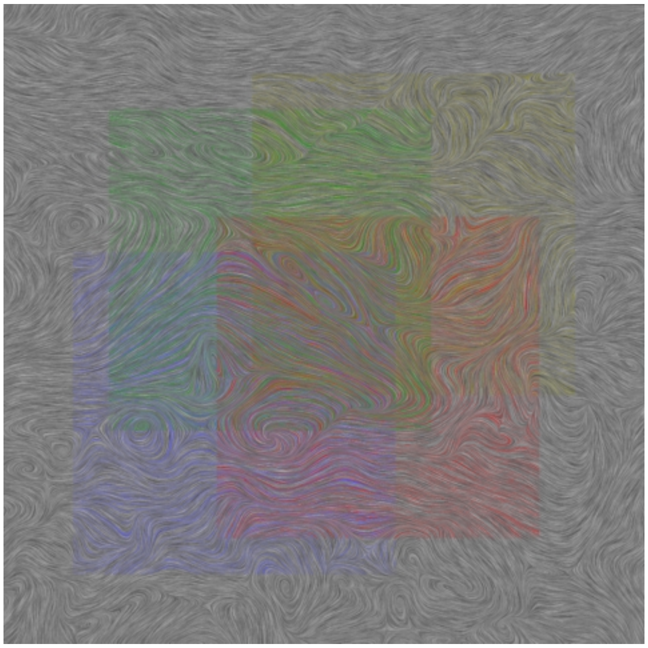


Figure 16: The same four regions, combined in the same LIC image via *color weaving*. Note the continuity of color along individual streamlines in a region and the ability to accurately perceive combinations of component colors in the areas of high overlap (characterized by the presence of three or more layers).

We begin by selecting several perceptually iso-luminant colors offered by Kindlmann [19]. As luminance plays a primary role in how features are perceived [18], selecting base colors that are as perceptually uniform as possible helps to achieve a final image in which similar concentrations are represented with reasonably equivalent prominence across the multiple distributions.

We introduce color on a streamline-by-streamline basis during the computation of a LIC image. At each pixel along a streamline we define the final image color as a linear combination of a selected base color and the default grey value in the LIC image, weighted by the magnitude of the value in the scalar field driving the color encoding of the variable associated with that base color. This results in the color being fully saturated at points where the scalar variable reaches its maximum and fading to the default LIC value where the magnitude of the scalar variable falls below threshold significance.

In areas characterized by the presence of above-threshold values in multiple distributions, alternate colors are applied along adjacent streamlines. The streamline based fast-LIC [4] algorithm is critical to our implementation as the color index is only incremented when a new streamline is calculated. (A texture-stitching algorithm, described below, is used to achieve the appropriate termination of color at region boundaries.) For the images in this section, a 1071x1071 input texture was used, and values for maximum streamline length were 380.

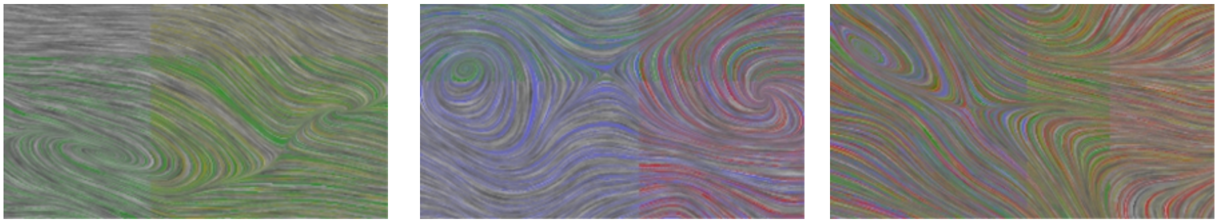


Figure 17: Three close-up excerpts from the color overlap image shown in figure 14. Left: background, yellow only, green only and the yellow + green overlap region. Middle: blue only, blue + green overlap, blue + red overlap, and blue + green + red overlap. Right: red + yellow everywhere, partially overlapped by green, and partially overlapped, additionally, by blue.

The first step in creating a color weave image is to create, using the same white noise input texture, a suite of multi-colored component LIC images corresponding to all possible combinations of presence or absence of supra-threshold values at any point in each scalar distribution. (Typically we define the threshold level to be the average expected value within each distribution.) For example, if we wish to create a color weave image that represents three completely independent scalar distributions (a,b,c) over a vector field, we would have to compute component images corresponding to the seven cases of: (a), (b), (c), (a and b), (a and c), (b and c), (a and b and c). We use a sparse and consistent mapping of individual colors to individual streamlines in order to ensure that an image representing a combination such as (a and b) differs from an image representing (a only) only to the extent that it includes additional streamlines colored, instead of left in the default grey, according to the hue associated with the distribution (b). This enables the multiple component images to be stitched together in a seamless manner. The final composite image is formed by sampling from the appropriate component image, on a per-pixel basis, according to which distributions have values above significance at that point. Explicitly, to define the color of a pixel (x,y) in the output image, we determine whether the value in distribution a is above the threshold for distribution a, and whether the value in distribution b is above the threshold for distribution b, and similarly for distribution c. If we determine that none of these values are above threshold, we sample from the uncolored component image at (x,y). If we determine that only distribution a has a value above threshold at (x,y), we sample from the component image corresponding to the case (a only). This would be an image in which sparsely distributed streamlines lines are colored with the color chosen to be associated with distribution a, according to the concentration of the values of a at each point. Similarly, if we determine that each of the distributions (a,b,c) have values that are above their respective thresholds at (x,y) we would sample from the component image corresponding to the case (a,b,c), which would be an image in which alternate streamlines assume colors associated with each of the distributions a, b, and c, as well as not assuming any color. Note that the strength of each variable has already been represented by the amount of saturation in the colors at the various points in the appropriate component image.

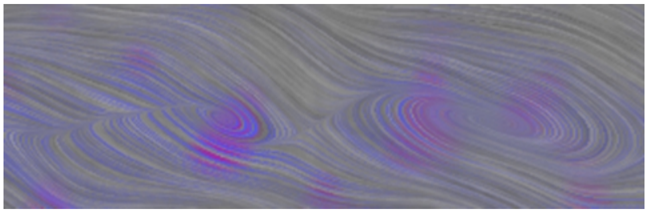


Figure 18: A close-up view of a portion of the component image in figure 19, center. The blue color corresponds to the presence of significant negative vorticity while magenta indicates areas of significant swirl strength within regions of negative vorticity. Note how the strengths of the values in each scalar distribution are reflected in the saturation of the added color at different points across the image.

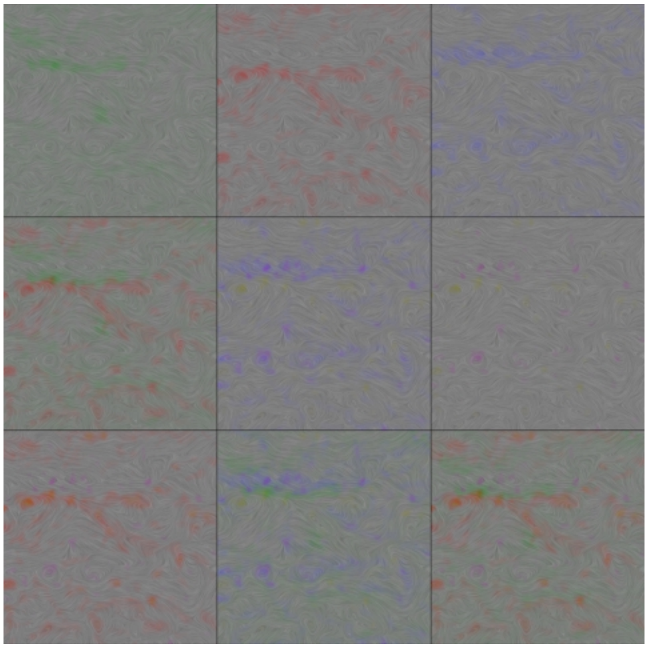


Figure 19: Members of the set of input components needed to create the final composite image in figure 20.

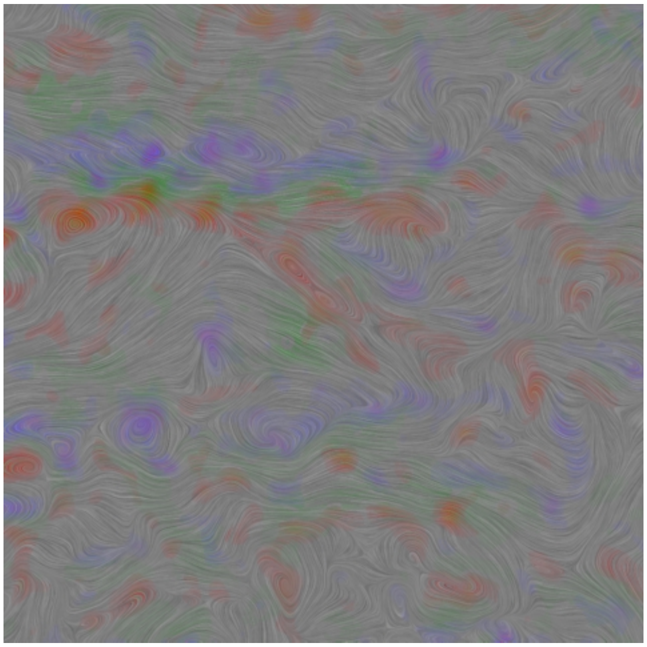


Figure 20: A composite ‘color woven’ image of an experimentally acquired PIV dataset in which we simultaneously highlight areas of significant positive vorticity (red), negative vorticity (blue), strongly negative Reynolds shear stress (green), and high swirl strength (yellow and magenta).

Figure 19 shows nine of the component images needed to create the final image shown in figure 20, in which four scalar distributions are simultaneously represented. Output pixels are sampled from these images, respectively, under conditions of: below average negative Reynolds shear stress only (top left: green), above average positive vorticity only (top middle: red), below average negative vorticity only (top right: blue), above average swirl strength only (center right: yellow/positive and magenta/negative), above average swirl strength and below average negative vorticity (center middle: blue + yellow/magenta), below average Reynolds shear stress and above average positive vorticity (center left: red + green), above average swirl strength and above average positive vorticity (lower left: red + yellow/magenta), all notable conditions except above average positive vorticity (lower middle: blue + green + yellow/magenta), and all notable conditions except below average negative vorticity (lower right: red + green + yellow/magenta). Because the conditions of above average positive vorticity and below average negative vorticity can never be simultaneously satisfied at any single point, there is no need to create a component image corresponding to that condition (blue+red). The default uncolored case, corresponding to the situation in which none of the distributions assumes a significant value, is also needed, but for layout reasons is not explicitly shown in figure 19.

8. TEXTURE STITCHING

Figure 21 illustrates the classical problem with attempting to apply a color wash to an input texture, before running LIC, to indicate the distribution of values in a scalar field associated with the vector data: the effect of the LIC is to smear out the colors, distorting the appearance of the scalar distribution in the final image and impeding efforts to accurately interpret the value of the distribution from the value of the color at any particular point. For this reason, color encoding is universally applied post-LIC, unless it is explicitly desired to use the color to demonstrate the effects of advection. Being aware of these issues with respect to the use of color, and wishing to use spatial frequency to encode the presence of discrete regions of interest in our data, we sought to develop ‘texture stitching’ – a post-LIC variant of the pre-LIC multi-frequency method proposed by Kiu and Banks [9] in which it would be possible to preserve the fidelity of region boundaries implicitly indicated by spatial frequency differences in the texture pattern in the final image, without introducing unnecessary discontinuity artifacts.

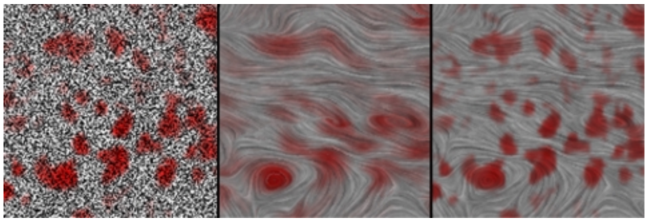


Figure 21: An illustration of the problem with trying to use color to indicate regions of interest pre-LIC. Left: color wash applied to the input texture. Middle: results after running LIC – the region definition is not well preserved. Right: results of applying the color wash post-LIC.

Following Kiu and Banks, the first step in our approach is to construct a set of correlated noise texture images by low pass filtering an initial high frequency noise pattern and equalizing the intensity histogram of the result to the intensity histogram of the original. We only needed to generate two noise texture patterns (high and low) because for our application we were primarily interested in using spatial frequency to indicate the locations of computed ‘regions of interest’ within a larger surrounding flow field. To create the images, we applied a Gaussian filter of width 20 and standard deviation 2.0 to the white noise shown in figure 22 (left) to achieve figure 22 (right).

There is a direct correlation between the size differences of the spots in the two input textures and the filter kernel length differences that are required to achieve output textures that will appear to differ by only a uniform (isotropic) scaling factor. Although we did not intend to attempt to use filter kernel length to encode any information about the flow, we determined that we would need to use different filter kernel lengths in the computation of the LIC in the high frequency and low frequency cases to avoid introducing the appearance of significant differences in the amount of texture anisotropy in the two regions. Since the lower frequency lines are less effective at conveying details of the flow orientation, we decided to use the low frequency texture to demarcate the regions of interest, which are characterized by uniform momentum and low velocity.

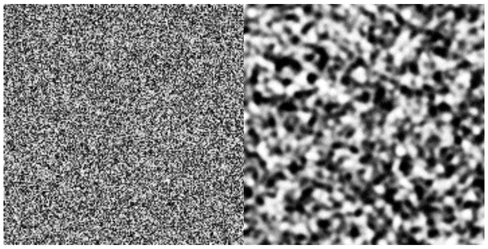


Figure 22: Samples from input textures used in our texture stitching technique. Left: a high frequency noise input texture. Right: the low frequency pattern achieved after Gaussian blurring and histogram equalization.

We proceeded by using the high frequency noise input and the low frequency noise input textures to create two separate LIC images. We also created a binary mask corresponding to the results of our trial region detection algorithm [1] (one of the goals of the visualization effort was to determine the suitability of the results produced by our region detection method and possibly to provide insight into how it might be refined to achieve greater effectiveness). We used the binary mask, shown in figure 23, to composite the results post LIC. One issue that arose in this process was the question of whether it might be desirable, or not, to minimize the incidence of contrast differences between the low frequency and high frequency texture patterns. Perceived contrast will inevitably be lower for the higher frequency pattern (unless there is a huge reduction in filter kernel lengths) because more different grey values will be averaged together, bringing the result closer to the mean than in the case of the low frequency pattern. Retaining the ability to equalize contrast, which can easily be done in the texture stitching approach, reserves the potential to use contrast differences to encode yet another, related, scalar distribution.

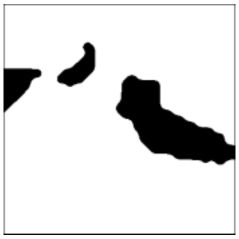


Figure 23: The region of interest binary mask.

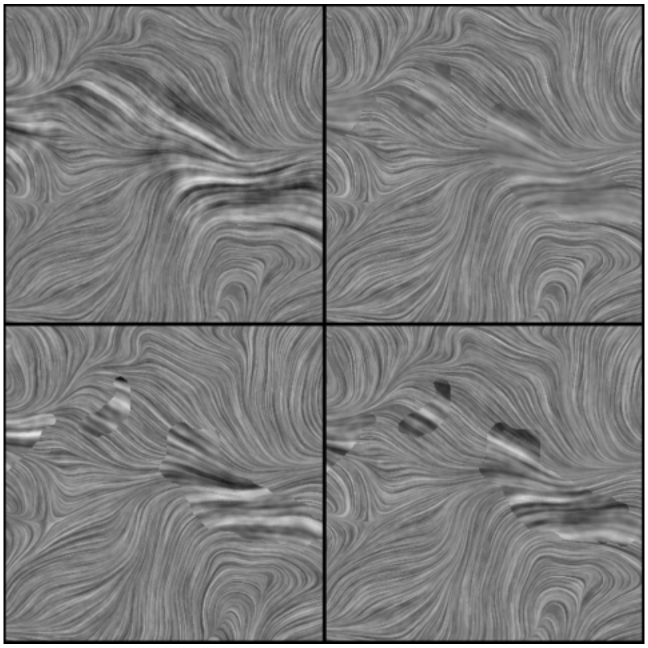


Figure 24: Comparison of texture stitching techniques. Images are obtained, clockwise from upper left, using: the Kiu/Banks algorithm; texture stitching with histogram equalization; texture stitching plus contrast enhancement of low frequency regions; texture stitching using unrelated input patterns.

The main drawback of texture stitching, compared to the multi-frequency LIC approach taken by Kiu and Banks, is that it allows region boundaries to be noticeable in the final image. They are far worse in the pathologic case where the input texture patterns are obtained from different random noise sequences, but even in the control image they remain quite apparent. Hence the texture stitching approach will not be suitable for applications in which one hopes to approximate a continuous series by a finite set of different spatial frequency patterns, which was the target application for Kiu and Banks.

9. COMBINATION OF COLOR WEAVING AND TEXTURE STITCHING

The color weaving and texture stitching techniques can be combined to visualize multiple variables and emphasize regions of interest in the data. The first step in the process of merging the two techniques is to create color weave component images using the different spatial frequency input patterns.

Using a low spatial resolution input texture does not change the resolution of the streamlines calculated in the present implementation. If the same technique is used for color selection in the low and high frequency texture cases, the result will be an appearance of high frequency color changes within a lower frequency luminance-dominated pattern, as seen in figure 25. The low frequency color weave image can also be combined via texture stitching, as shown in figure 26, with the high frequency image color weave image computed earlier, using the region of interest mask partially shown in figure 23.

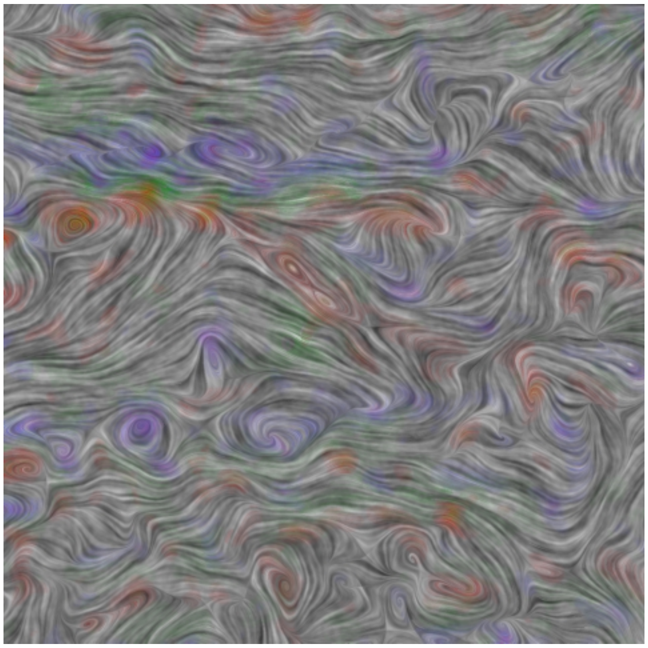


Figure 25: A color weave image created using a low frequency input pattern.

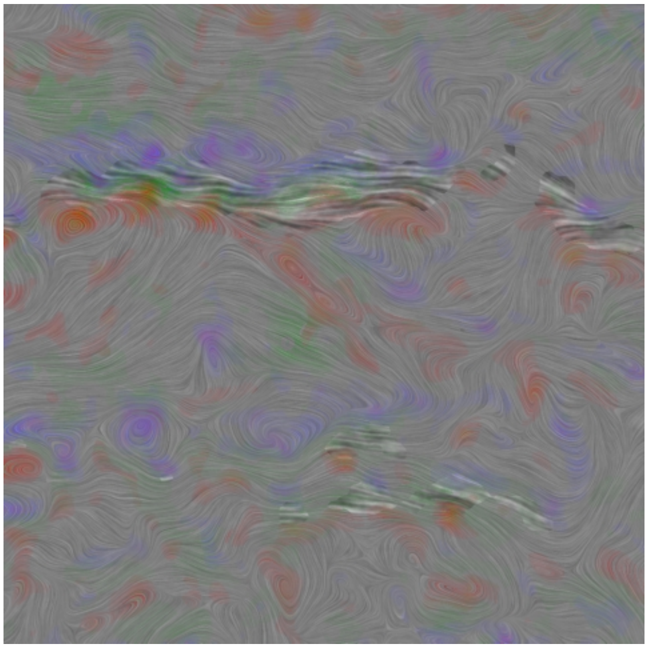


Figure 26: A combination of color weaving and texture stitching, with contrast enhancement applied in order to attract attention to the low spatial frequency regions corresponding to the probable presence of coherent packets of hairpin vortices.

10. LAYERING

Weigle *et. al* [14] contributed a texture generation technique, based on the layering of patches of oriented slivers, which uses orientation and luminance to encode information about multiple overlapping scalar fields. In their results, they presented an image that represented eight different variables through the use of layering. Although our work in this area is very preliminary, we feel that the use of layers can be effective in representing multi-valued flow data.

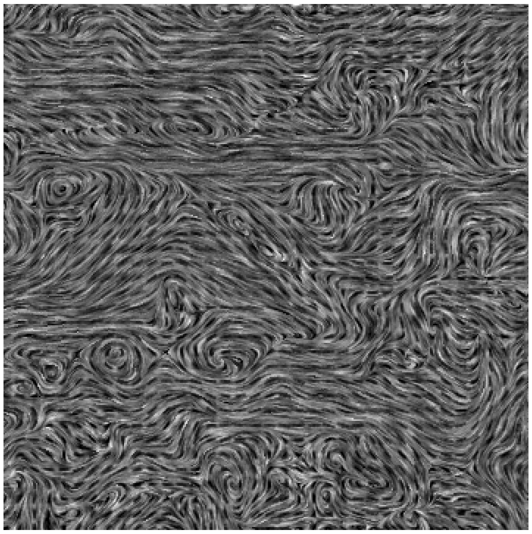


Figure 27: A layered LIC image. A traditional LIC image is overlaid with a LIC image using a sparse set of oriented spots as an input texture.

By using an oriented grid of points as an input texture, the LIC algorithm will output an image that convolves these spots in the direction depicted by the vector field. While this sparse sampling is not enough to get an accurate representation of the entire flow field, overlaying a traditional LIC image with this texture presents several opportunities. We have a few ideas for future direction in this area. Using the position or randomness of the sparse points in the input texture could effectively represent another variable. Another idea is to overlay two correlated LIC images from different planes of the same PIV data. The length or thickness of the oriented line segments could also be utilized. This is an area of future research.

11. SUMMARY & FUTURE WORK

The goal of this research is to effectively visualize multi-valued flow data by providing techniques that allow variables to be accurately perceived independently and also in context of other distributions. The analysis was conducted using experimentally acquired stereo PIV turbulent flow data. Multivariate visualization is of critical importance in facilitating the understanding of the results of such experiments because of the need to achieve an integrated understanding of the individual contribution of each variable and also how variables inter-relate.

We presented a structure identification algorithm that accurately determined regions of uniform momentum enveloped by cores of vorticity of opposite sign that contained seed points of significant Reynolds shear stress. The algorithm allowed for the automatic detection of frames that would reveal the structures consistent with hairpin vortex packets.

While there are many different methods for visualizing flow data, we have focused our efforts on texture-based methods due to the advantages they have over typical vector-glyph approaches such as high resolution and a global sense of flow. Using a texture-based approach, we investigated how to effectively represent multi-valued data using other components such as contrast, 3D visualization, and color.

Manipulations of mean luminance and contrast have the ability to enhance characteristics of an image with the intent of carrying information about a scalar distribution. Both techniques appear to be effective, but the differences and inter-relations between the two are subtle. Future research is needed to understand how contrast and mean luminance can be effectively used together.

The challenge to represent multiple variables that coexist has inspired us to use 3D visualization to represent an additional parameter. Using standard graphics utilities, we texture-mapped the LIC texture on a 3D terrain where height represents another parameter. While effective, this technique raises questions regarding the ability to accurately perceive the height field in context with the texture. We plan on investigating further methods that may allow the independent interpretation of both height and texture.

'Color weaving' provides an alternative to traditional color composition by allowing multiple colors to be closely interwoven via the assignment of distinct separate hues to individual streamlines, rather than blended. In order to allow each color to encode multiple values in a continuous distribution, we let the saturation of the color at each point vary according to the value in the corresponding scalar distribution. We achieve consistent combinations of colors by assigning color indices to streamlines in an alternating manner that depends on the order in which they are encountered in a deterministic walk through the pixel grid. Multiple multi-colored component images are created that represent all possible combinations of distributions that may contain values above significance at some point. A final image is generated by sampling from the appropriate component image indicated by the vector of data values at each point. The result is a multicolored LIC image that resembles a tapestry woven with different colored threads.

'Texture stitching' allows preservation of faithful region boundaries in multi-frequency LIC through the use of post-LIC merging of selected adjacent regions. We first obtain separate LIC textures based on correlated high and low frequency noise input patterns, then combine the results using a binary mask to force adherence to pre-defined boundary curves.

In the future, we would like to extend our texture stitching approach beyond LIC to the case of multiple different texture types, finessing the problem of maintaining continuity between adjacent texture patches through the use of novel texture synthesis methods that operate on the basis of multiple considerations in the flow.

12. ACKNOWLEGMENTS

We would like to thank Ivan Marusic, Ellen Longmire and Bharathram Ganapathisubramani for all of their help with the PIV data and analysis. We would like to acknowledge Noor Martin, Margaret Richey, and Kirti Kesavarapu for their help in development of the software used for this research. This work was supported by a grant from the National Science Foundation (ACI-9982274).

References

[1] B. Ganapathisubramani, E. Longmire, and I. Marusic, “Characteristics of Vortex Packets in a Turbulent Boundary Layer”, Journal of Fluid Mechanics, vol. 478, 2003, pp. 35-46.

[2] J. van Wijk, “Spot Noise – Texture Syntheses for Data Visualization”, Proceedings of SIGGRAPH 91, pp. 309-318.

[3] B. Cabral and C. Leedom. “Imaging Vector Fields Using Line Integral Convolution”, Proceedings of SIGGRAPH 93, pp. 263-269.

[4] D. Stalling and H.-C. Hege. “Fast and Resolution-Independent Line Integral Convolution”, Proceedings of SIGGRAPH 95, pp. 249-256.

[5] R. Wegenkittl, E. Gröller, and W. Purgathofer, “Animating Flowfields: Rendering of Oriented Line Integral Convolution”, Proceedings of IEEE Computer Animation ’97, pp. 15-21.

[6] R. Wegenkittl and E. Gröller, “Oriented Line Integral Convolution for Vector Field Visualization via the Internet”, Proceedings of IEEE Visualization ’97, pp. 309-316.

[7] A. Sanna, B. Montrucchio, P. Montuschi, and A. Sparavigna, “Visualizing vector fields: the thick oriented stream-line algorithm (TOSL)”, Computers & Graphics, 2001, **25**(5): 847-855.

[8] H.-W. Shen, C.R. Johnson, K.-L. Ma, “Visualizing Vector Fields Using Line Integral Convolution and Dye Advection”, Proceedings of the ACM Symposium on Volume Visualization ’96, pp. 63-70.

[9] M.-H. Kiu and D. Banks, “Multi-Frequency Noise for LIC”, Proceedings of IEEE Visualization ’96, pp. 121-126.

[10] C. Ware and W. Knight, “Using Visual Texture for Information Display”, ACM Transactions on Graphics, January 1995, **14**(1): 3-20.

[11] A. Sanna and B. Montrucchio, “Adding a Scalar Value to 2D Vector Field Visualization: the BLIC (Bumped LIC)”, Eurographics 2000 Short Presentations Proceedings, pp. 119-124.

[12] A. Sanna, B. Montrucchio, C. Zunino, and P. Montuschi, “Enhanced Vector Field Visualization by Local Contrast Analysis”, Eurographics/IEEE TCVG Symposium on Data Visualization 2002, pp. 35-41

[13] C. Healey and J. Enns, “Building Perceptual Textures to Visualize Multidimensional Datasets”, Proceedings of IEEE Visualization ’98, pp. 111-118.

[14] C. Weigle, W. English, G. Liu, R. Taylor, J. Enns and C. Healey, “Oriented Sliver Textures: A technique for local value estimation of multiple scalar fields”, Proceedings of Graphics Interface 2000, pp. 153-162.

[15] D. Laidlaw, E. Ahrens, D. Kremers, M. Avalos, R. Jacobs, and C. Readhead, “Visualizing Diffusion Tensor Images of the Mouse Spinal Cord”, Proceedings of IEEE Visualization ’98, pp. 127-134.

[16] R. Kirby, H. Marmanis, D.H. Laidlaw, “Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting”, Proceedings of IEEE Visualization ’99, pp. 333-340.

[17] G. Turk and D. Banks, “Image-Guided Streamline Placement”, Proceedings of SIGGRAPH 96, pp. 453-460.

[18] C. Ware, Information Visualization: Perception for Design, Morgan Kaufman, 2000.

[19] G. Kindlmann, E. Reinhard, and S. Creem, “Face-based Luminance Matching for Perceptual Colormap Generation”, Proceedings of IEEE Visualization 2002, pp. 309-406.