Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Techniques for Visualizing Multi-Valued Flow Data

Timothy Urness[†1] Victoria Interrante[1] Ellen Longmire[2] Ivan Marusic[2] Bharathram Ganapathisubramani[2]

[1] Department of Computer Science and Enginneering
[2] Department of Aerospace Engineering and Mechanics

University of Minnesota

Abstract

In this paper we discuss several techniques to display multiple scalar distributions within an image depicting a 2D flow field. We first address how internal contrast and mean luminance can effectively be used to represent a scalar distribution in addition to an underlying flow field. Secondly, we expand upon a current technique to more effectively use luminance ramps over dense streamlines to represent direction of flow. Lastly, we present a new method, based on embossing, to encode the out-of-plane component of a 3D vector field defined over a 2D domain. Throughout this paper, we limit our focus to the visualization of steady flows.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

The goal of scientific visualization is to represent information in a manner that is easy to interpret, accurate, and lends itself to a fundamental understanding of the underlying organization of the information being displayed. Whether consciously considered or not, the process of evaluating and analyzing scientific visualization images relies upon human visual perception and aesthetics. To produce an image that successfully reflects multi-valued data it is necessary not only to be accurate in the representation of individual distributions, but also to portray each specific component in a way that does not interfere with the accurate perception of the other components. This process of successfully combining variables has important applications in analyzing the scientific phenomena represented by multi-variate data. As an alternative to representing different variables on separate domains and attempting to piece the data together using side-by-side displays, it is highly desirable to create images that allow each distribution to be understood both individually and in the context of one or more of the other distributions.

While several techniques exist to visualize flow fields (spot noise, line integral convolution, hedgehogs, etc.) displaying scalar fields within flow field representations remains a ubiquitous problem. This paper presents techniques that can be useful for representing such data.

We begin by reviewing many of the existing techniques to represent scalar variables in addition to flow fields. We continue with a more detailed discussion on how the concepts of manipulating mean luminance or contrast can be used to represent scalar values over a 2D vector field. Additionally, we analyze a recent technique for using luminance ramping over dense streamlines to represent flow direction and advocate a few modest modifications. Finally, we present a technique designed to visualize a 3D vector field on a 2D domain through the use of variable embossing to give the perception of depth.

2. Background and Previous Work

2.1. Visualizing Vector Fields

Perhaps the most prevalent and straightforward approach to representing a 2D vector field is to use a series of glyphs known as vector plots or hedgehogs. While effective and efficient, one limitation is that such plots can only provide information at relatively sparsely sampled points over a domain, as each glyph will require several pixels to be drawn. Even when the vector field is not downsampled, the collection of glyphs may not easily lend itself to the perception of global fluid flow as the segments must be perceptually in-

† urness@cs.umn.edu

terpolated and connected in order to understand the path of a particle. Selecting samples along a uniform grid may lead to artifacts in which the structure of the grid interferes with correct perception of the direction indicated by the vectors [LKD*01].

Vector fields have traditionally been effectively visualized through the use of texture-based algorithms. In pioneering work, van Wijk [vW91] introduced the concept of *spot noise*, a texture produced from weighted and randomly positioned spots deformed in accordance with the direction of flow. Cabral and Leedom [CL93] presented *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. Stalling and Hege [SH95] increased the efficiency of the LIC algorithm by taking advantage of coherence along streamlines. This results in the computation of the output texture being streamline oriented, not pixel oriented.

One disadvantage of traditional LIC images is that the direction of movement in a flow is ambiguous. Animation can be used to make that information explicit [SH95, CL93, JL97b]. Wegenkittl, Groller, and Purgathofer [WGP97] introduced a technique called Oriented Line Integral Convolution (OLIC) that addresses this issue in a single static image. The OLIC algorithm, in essence, uses a sparse texture resembling ink droplets on a page as input and a ramp-like convolution kernel smears 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) [WG97]. More recently, in another approach similar to OLIC, Sanna et al. [SMMS01] propose a Thick Oriented Stream Lines (TOSL) method, in which the orientation of a flow is depicted by increasing the luminance along calculated streamlines.

Kiu and Banks [KB96] propose the use of multi-frequency input textures along with increased filter kernel lengths to incorporate indications of velocity magnitude. Khouas, Odet, and Friboulet [KOF99] use a 2D autoregressive synthesis method to simulate a 3D fur-like texture in order to represent two dimensional flow fields. This technique allows control over streamlet orientation, length and density, and has been used to produce striking visualizations of vector orientation and magnitude.

Indications of velocity magnitude and direction have thus been successfully reintroduced into texture-based flow representations through the use of luminance ramps and variations in spatial frequency. In the next section we review current techniques designed to represent additional scalar values in addition to an underlying vector field.

2.2. Visualizing Additional Data

As early as 1991, Crawfis and Allison [CA91] presented a technique for mapping multiple scalar fields in addition to a vector field on a non-planar surface using the visual variables of color, texture, and height. An inherent limitation in visualizing vector data over surfaces in 3D is that it can be difficult for observers to disambiguate the effects of projection – possibly causing false interpretation of the orientation information. Also, the 3D nature of the new surface may occlude other regions of the domain.

Scheuermann et al. [SBH99] combined surface deformation with LIC to portray 3D vector data defined over a 2D domain. A similar approach presented by Sanna and Montrucchio (BLIC) [SM00] uses bump mapping to encode an arbitrary additional scalar variable over a vector field.

Using light sources and shading can work well for representing 3D data on a planar domain. One must take care, however, to ensure that the display of information via shading does not interfere with the interpretation of the texture.

Kirby, Marmanis and Laidlaw [KML99] used a layering technique, inspired by methods from oil painting, to represent 2D flow images, encoding up to three or more different variables, in addition to the basic flow field, through the use of glyph size, shape, contrast, and color.

Sanna et al. [SMZM02] introduce an additional value to a flow field by accentuating an underlying flow texture with local contrast. Contrast is also addressed by Ware and Knight [WK95] who use 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.

3. Using Contrast and Luminance

The role of the luminance component has a prominent effect on how features in an image are perceived [War00]. 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 a texture. We illustrate these methods on LIC images.

3.1. Contrast

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

Once an original image is created to display a flow field, contrast can be manipulated by altering the grey-level values in an image depending on the values in a scalar field that is intended to be displayed. At the locations of prominent scalar values, we reassign pixel values to darker greys

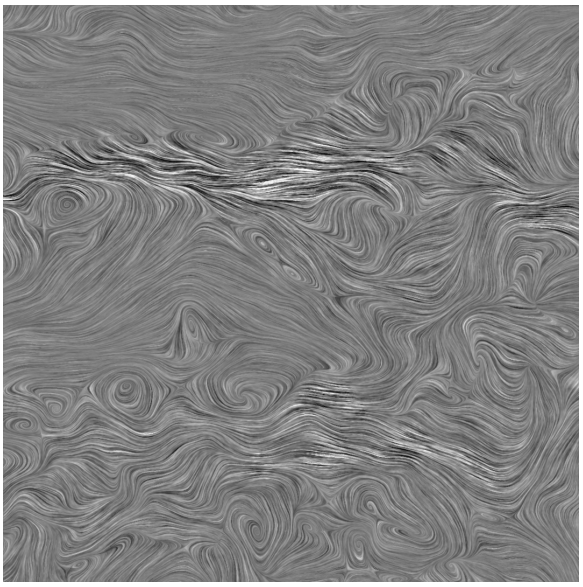

Figure 1: *Manipulation of contrast is used in this image to represent a scalar distribution. Local differences between black and white values are used to portray the calculated quantity of uniform momentum.*

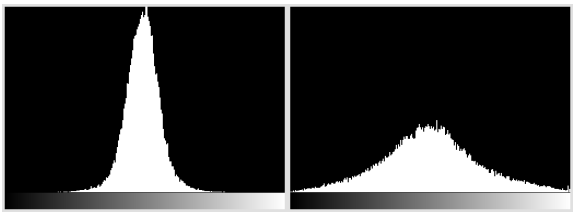

Figure 2: *Two histograms taken from different regions of figure 1. The left image depicts the intensity distribution of a low-contrast region. The rightmost image depicts a high-contrast region.*

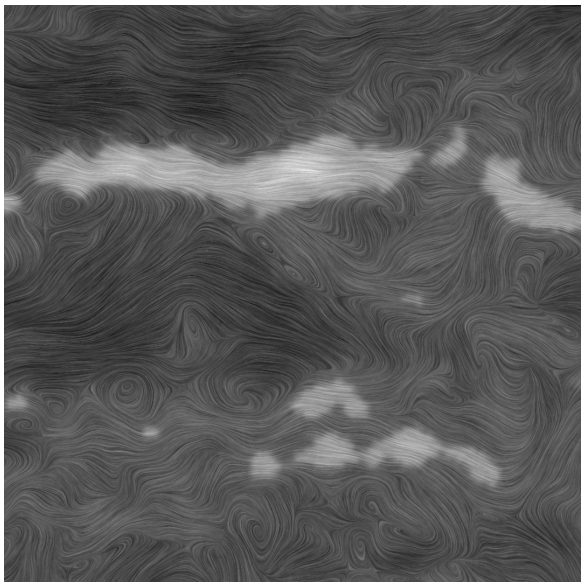

Figure 3: *Manipulation of mean luminance is used in this image to represent a scalar distribution. Luminance values in an original LIC image are shifted according to the values in an auxiliary scalar distribution – in this case representing uniform momentum.*

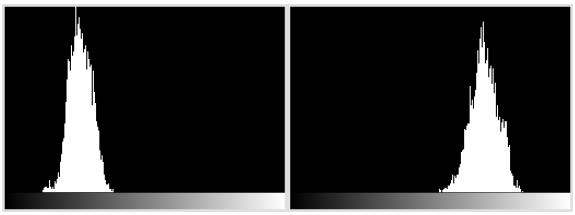

Figure 4: *Two histograms taken from different regions of figure 3 showing that the shapes of the histograms remain similar, and only the average luminance value is changed.*

or lighter whites than the original and reassign lesser values of the scalar field to be closer to the average intensity. The resulting intensity histogram for the image will be different, but the average pixel value over the new image will remain similar to that of the original. An example is shown in figure 1.

Images created in this manner can work to effectively display a scalar distribution because the human visual system is sensitive to different levels of disparity between the blacks and whites in an image. Figure 2 displays the histogram taken from two separate regions of figure 1. Both histograms suggest that the average pixel value is in the middle of the range. However, high-contrast regions require the entire range of intensity values while a much more narrow range is utilized in low-contrast regions. A continuous scalar distribution can thus be encoded over a texture image through variations in the internal dynamic range of the pattern.

3.2. Mean Luminance

Mean luminance refers to the average intensity value of the pixels in a given region and can be characterized as the overall brightness of a region. The default 8-bit grey-level values of a LIC image generated from a random white-noise input texture typically have an average value close to 127. The average luminance of the image is changed by adding or subtracting an amount proportional to the scalar distribution. We ensure the luminance value does not go out of range by applying the following formula where α is an arbitrary fixed value determined by the user.

$$I_{new}(x,y) = \alpha * scalar(x,y) + (1-\alpha) * I_{old}(x,y)$$

Applying this formula essentially shifts the histogram with respect to the scalar value while maintaining the overall shape. Using this technique, the contrast between the black and white values of lines remains the same as in the original image and the average luminance value encodes the scalar distribution (figure 3). Care must also be taken to consider the nonlinearity of intensity perception when using this or any other intensity remapping.

4. Depicting Flow Direction

Several techniques have been proposed to overcome the problem of ambiguous flow direction that occurs in static LIC images [WGP97, WG97, SMMS01]. These techniques typically involve either animation or the use of a monotonically increasing luminance ramp to disambiguate the direction of the flow.

Sanna et al. [SMMS01] developed a space-filling method they called Thick Oriented Stream Lines (TOSL), in which the orientation of a flow is depicted by increasing luminance values along calculated streamlines. An advantage of this technique is that it provides a dense representation of the vector field.

The first step in the TOSL method, as in the LIC method, is to 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 initial 8-bit pixel value is randomly set within a range of 30 and 120 and the algorithm continues by stepping along each pixel calculated in the streamline and assigning an increasing intensity value. Local vector magnitude is taken into account, as the value of each pixel is incremented by an amount that reflects the velocity magnitude at that point. Each vector magnitude is normalized with respect to the maximum velocity on the local streamline. If the vector has a high relative velocity along the streamline, the increment in grey tone is proportionally high at that point in the image. A high density output texture is obtained by creating the image in two passes. The first pass creates a sparse texture by coloring only a percentage of the pixels, using a specific procedure to select randomly spaced seed points. After a user-defined percentage of the image has been filled, the remaining pixels are considered in scan-line order to ensure that the entire image is completed.

The TOSL technique is particularly advantageous because of its high density output, ability to accurately depict flow direction, relative simplicity, and potential for efficient implementation. Using the original TOSL algorithm as inspiration, we extend the technique to enhance the visual effect and improve perception of the flow field.

We have found that starting with an initial intensity value between 30 and 120 can lead to artifacts due to streamlines that have a similar range and start from a similar point (such as the edge of the domain or a singularity). Figure 5 shows that this can result as a darkened and uniform artifact along the edge of the domain where the streamlines are introduced. The authors of TOSL suggest this starting range in order to avoid initial dark grey and also to avoid *jumps* in which the values of neighboring pixels would be 255 and 0. It has been our experience that computing long streamlines in which several cycles of pixel values ranging from 0 to 255 occur, including *jumps*, can be advantageous. This allows one streamline to carry several repeated luminance ramps indicating the direction of flow and results in a fluid final image. The problem of edge artifacts can be alleviated by allowing the starting value to be randomly assigned to any value between 0 and 255.

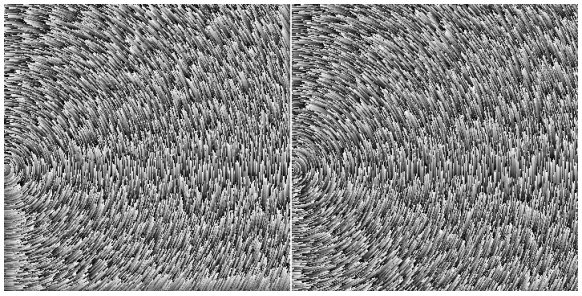

Figure 5: *Restricting the initial value of the streamline can lead to artifacts at boundary of the image domain, as seen in the lower left and bottom of the leftmost image. This problem can be alleviated by allowing the initial pixel values to span the entire range from 0 to 255.*

Secondly, we feel that a more accurate representation of the entire vector field could be obtained by using the maximum global vector magnitude to normalize the step size. Incrementing the intensity of pixels with a step size that is directly proportional to the local vector velocity magnitude produces appealing results at the expense of global inconsistencies. With this approach, one cannot compare line lengths in different areas of the image to determine if the velocity magnitude is at a global maximum or simply a local maximum. By adjusting the factor in which the vector magnitudes are normalized, it is possible to provide a more globally consistent portrayal of the scientific phenomenon.

Finally, we find it appropriate to make the step size inversely proportional to the vector velocity magnitude instead of directly related to the velocity magnitude. While an observer of an image may learn to read short lines as representing high speed areas, and long lines as representing slow moving flow, we find this approach counterintuitive. Reversing the mapping results in creating long smooth lines where the flow velocity is at the global maximum. This is evocative of the result that a spot smeared out over a period of time

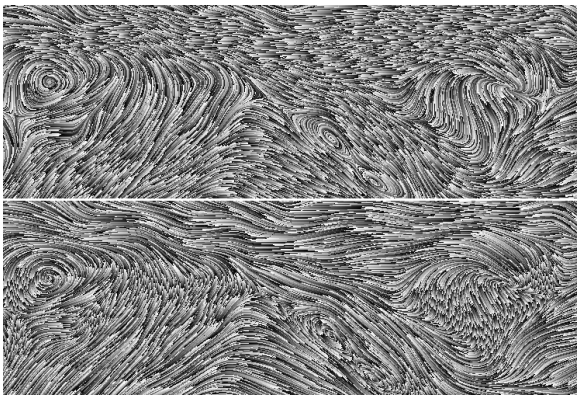

Figure 6: *In the topmost image, velocity magnitude is normalized with respect to the values along the local streamline only. This results in short streamlines at places where the velocity is greatest along each individual streamline. In the lower image, velocity magnitude is normalized with respect to the global velocity magnitude. The luminance ramp along streamlines is also defined using an inverse relationship between the step size and the vector magnitude, resulting in long streamlines where the velocity is at the largest magnitude globally over the domain*

will produce a long streak where the flow is faster. Figure 6 illustrates the effects of making these changes.

5. Color

Using the technique presented in section 4 to visualize a vector field with dense streamlines, color can be effectively added in a way that allows multiple distributions to be represented simultaneously [UIL*03]. An effective visualization of multi-valued flow data can be achieved by using different hues for different scalar distributions and applying color in a manner that maintains the intensity of the original dense streamline image and saturation according to the respective scalar value (see figure 7 in color section).

6. Streamline Density

Controlling streamline density facilitates several effective methods of visualizing 2D vector fields. Here we discuss a few variations on previous methods [TB96, JL97a] that present information over vector fields by controlling the density of the placement of streamlines. Our methods allow further techniques to visualize additional distributions on these images.

One way to create a sparse texture that accurately reflects a vector field is to begin by using a random distribution to select seed values for streamline calculation. Once a starting point has been selected, the streamline is calculated in both the positive and negative direction. Following [JL97a], the

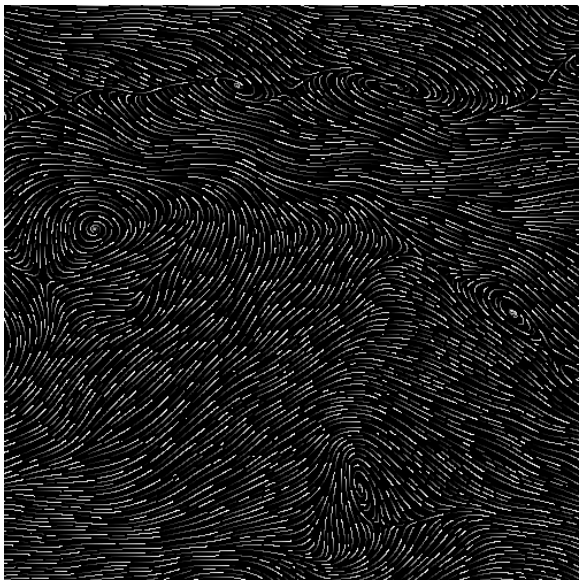

Figure 8: *A sparse texture of evenly distributed streamlets.*

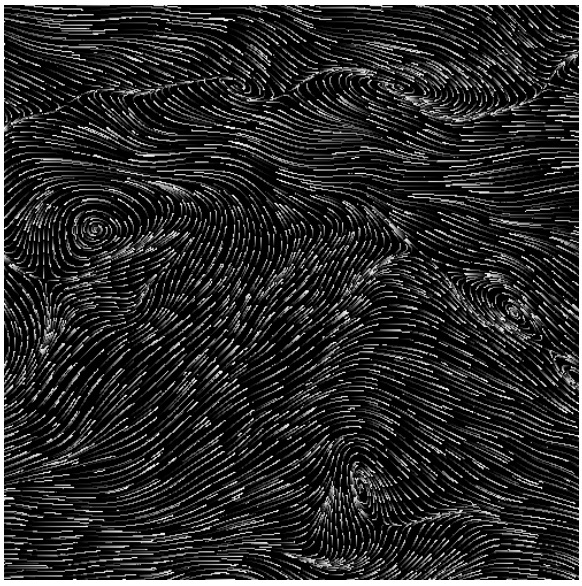

Figure 9: *A sparse texture in which streamlines are not terminated when their proximity becomes too great. This enhances bifurcation lines in the data.*

streamline is traced in each direction until one of the following occurs: a singularity is reached, the streamline reaches the edge of the domain, or the streamline comes within some user-defined distance of another streamline that has already been calculated. Intensity values are assigned beginning at the negative end of the streamline. The starting intensity is randomly selected within the range of [0,255] and subse-

quent pixels along the calculated streamline in the direction of flow are assigned monotonically increasing intensity values, wrapping around from 255 to 0, until the end of the streamline is encountered. Streamlines whose total length is less than a user-specified minimum are not colored in.

If the restriction of proximity is not enforced, a remarkably different image results. The initial pixel selected at random is checked to determine if another streamline has been computed within a defined proximity. Once a streamline has been initialized, the entire streamline gets defined and colored in regardless of whether it comes too close to another streamline that has already been computed. The effect is that areas where streamlines converge are indicated by a much more dense coverage of streamlines than in the previous technique. This increased density highlights the interface between converging flow regions and can be interpreted as a bifurcation line.

7. Embossing

Applying 3D shading or lighting effects, such as bump mapping or embossing, to 2D images can be an effective method for producing the perception of three dimensional shape.

In order for the embossing technique to be effective, the image best not contain a large number of dense, high-frequency discontinuities. Employing embossing techniques on images such as figure 6 would not be advisable as there is not sufficient space within the image to perceive the results of the shading equation. For this reason, we use the sparse texture shown in figure 8 as input for the embossing techniques.

For the images presented in this section, the additional distribution that we have chosen to visualize is the out-of-plane vector component w. The characteristics of w is significant to various theories and other derived quantities that are valuable for analysis of the turbulent flow data. However, this quantity is often ignored when producing 2D images of 3D vector fields because it is more convenient to simply portray only the in-plane components. With embossing, we can represent the vector field component w in a manner that everywhere reflects its depth distance (both positive and negative) from the base plane.

7.1. Light Direction

Light direction plays an important role with regard to the perception of depth. Embossing algorithms typically mimic a standard lighting equation with a single point light source.

An embossed image that represents a scalar distribution that contains both positive and negative values can be created by combining two images in the following manner. First, an image is created by applying the emboss algorithm with a light source from above, and then a second image is created with a light source from below. The final image is produced by selecting pixel values from the image lit from above wherever the quantity is positive and pixel values from the region lit from below wherever the quantity is negative. The desired effect is that positive values appear to be raised and negative values appear to be sunken (figure 10).

Figure 10: *An embossed and depth-shaded image representing the gain-adjusted magnitude of the out-of-plane component of a vector field.*

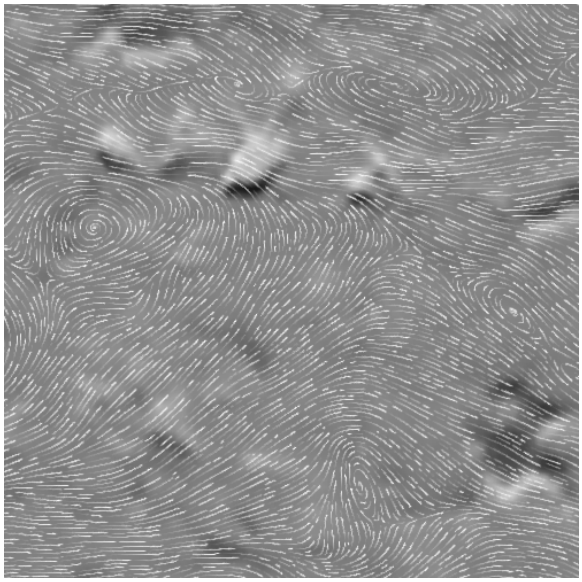

Figure 11: *Overlying a sparse streamline texture on an embossed representation of the out-of-plane vector component.*

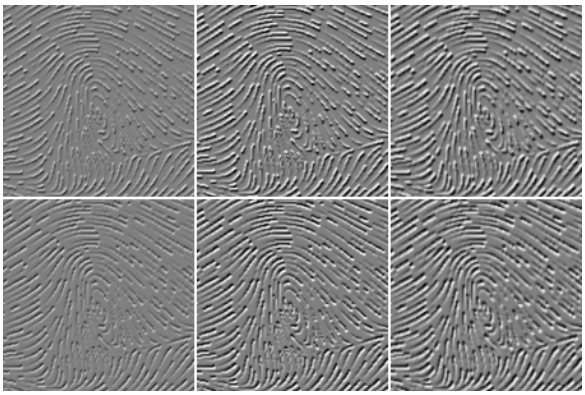

Figure 12: *Different levels of embossing applied to streamlines to represent the magnitude of an auxiliary scalar distribution.*

The first method we present is to overlay the sparse texture image presented in figure 8 and the embossed image of figure 10 in such a way that only the values in the former that are lighter than the values in the latter get written to the resulting image. The result is shown in figure 11. While simple, this technique allows us to effectively visualize both the in-plane and out-of-plane velocity components together in a single image.

7.2. Representing Values with Embossed Streamlines

To represent a scalar distribution through the use of embossed streamlines, the magnitude of the scalar value must be encoded in the depth of the embossing. The process is started by creating a discrete number of embossings of the image at different *depth levels*.

If the distribution desired to be displayed has positive and negative components, then two images are created for each level lit in opposing directions – one from above and one from below (figure 12). We found a linear interpolation between only a few levels to be sufficient to create the images presented in this paper.

The final image representing the flow field and scalar distribution is created on a pixel-by-pixel basis depending on the magnitude of the scalar component at each point. The value of the scalar field is queried at each point and a linear combination of the appropriate levels of embossing for the respective direction is recorded. This process is continued until all pixels are covered (figure 13).

7.3. Combining Embossing with Streamlines

A second and more sophisticated approach to representing the flow field in conjunction with the scalar distribution begins with the creation of an embossed streamline image of the vector field. The embossed streamline image is added to

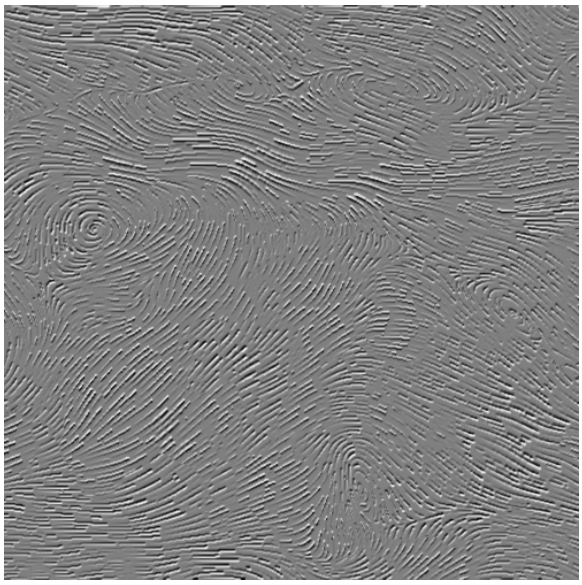

Figure 13: *Embossed streamlines*

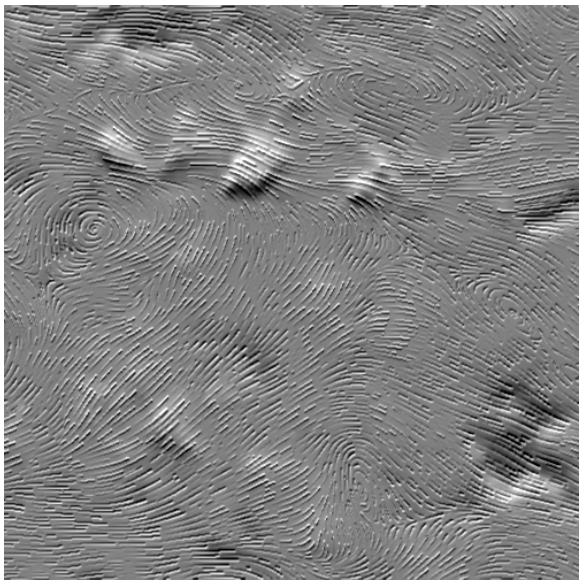

Figure 14: *Embossed streamlines on an embossed representation of the out-of-plane vector component.*

the scalar representation once again in a pixel-by-pixel fashion. Once the two pixel values are added, the background color from the embossed streamline image is subtracted. This results in an embossing of the original embossed scalar field as values that were below the average are subtracted from the original image and values above the average are added to the original image (figure 14). Any values outside the range of 0 or 255 are effectively clamped.

7.4. Limitations

Representing a scalar distribution using this embossing technique is largely impressionistic and contains a few limitations. Namely, the ability to perceive numerous discretized levels of a scalar variable is limited. While all images that reflect scalar values are limited in this manner to a degree, the embossing technique utilizes a number of surrounding pixels and different shades of grey to produce a depth effect. The combination of space and limited number of different levels of grey that can be effectively used to produce the depth effect significantly limit the number of perceptually distinguishable levels using this technique. Additionally, this technique is best employed when the data is relatively continuous and does not contain adjacent sporadic positive and negative jumps. This would cause the embossed streamlines to appear segmented in a manner that does not accurately reflect the underlying flow field.

8. Summary

The challenge of visualizing multiple scalar fields in combination with flow data has inspired many different techniques. In an attempt to better understand the components of visualizing multi-valued flow data, we have analyzed the basic concepts of contrast and mean luminance to visualize a scalar distribution within a flow field. We presented modifications to the TOSL algorithm that make it a more effective tool for displaying a flow field as a series of dense streamlines depicting flow direction using a luminance ramp. Finally, we described a technique using embossing to simulate normals of a 2D image and a lighting equation to produce the perception of 3D shape.

9. Acknowledgements

This research was supported by a grant from the National Science Foundation (CTS-0324898).

References

[CA91] CRAWFIS R., ALLISON M.: Scientific visualization synthesizer. *Proc. of IEEE Visualization '91* (1991), 262–267.

[CL93] CABRAL B., LEEDOM C.: Imaging vector fields using line integral convolution. *Proc. SIGGRAPH 93* (1993), 263–269.

[JL97a] JOBARD B., LEFER W.: Creating evenly-spaced streamlines of arbitrary density. *Proc. of the 8th Eurographics Workshop on Visualization in Scientific Computing* (1997), 43–55.

[JL97b] JOBARD B., LEFER W.: The motion map: Efficient computation of steady flow animations. *Proc. of IEEE Visualization '97* (1997), 323–328.

[KB96] KIU M.-H., BANKS D.: Multi-frequency noise for lic. *Proc. of IEEE Visualization '96* (1996), 121–126.

[KML99] KIRBY R., MARMANIS H., LAIDLAW D. H.: Visualizing multivalued data from 2d incompressible flows using concepts from painting. *Proc. of IEEE Visualization '99* (1999), 333–340.

[KOF99] KHOUAS L., ODET C., FRIBOULET D.: Vector field visualization using furlike texture. *Eurographics/IEEE TCVG Symposium on Data Visualization* (1999), 35–44.

[LKD*01] LAIDLAW D. H., KIRBY M., DAVIDSON J. S., MILLER T., DASILVA M., WARREN W., TARR M.: Quantitative comparative evaluation of 2d vector field visualization methods. *Proc. of IEEE Visualization '01* (2001), 143–150.

[SBH99] SCHEUERMANN G., BURBACH H., HAGEN H.: Visualizing planar vector fields with normal component using line integral convolution. *Proc. of IEEE Visualization '99* (1999), 255–261.

[SH95] STALLING D., HEGE H.-C.: Fast and resolution-independent line integral convolution. *Proc. of SIGGRAPH 95* (1995), 249–256.

[SM00] SANNA A., MONTRUCCHIO B.: Adding a scalar value to 2d vector field visualization: the blic (bumped lic). *Eurographics 2000 Short Presentations Proc.* (2000), 119–124.

[SMMS01] SANNA A., MONTRUCCHIO B., MONTUSCHI P., SPARAVIGNA A.: Visualizing vector fields: the thick oriented stream-line algorithm (tosl). *Computers and Graphics 25*, 5 (2001), 847–855.

[SMZM02] SANNA A., MONTRUCCHIO B., ZUNINO C., MONTUSCHI P.: Enhanced vector field visualization by local contrast analysis. *Eurographics/IEEE TCVG Symposium on Data Visualization* (2002), 35–41.

[TB96] TURK G., BANKS D.: Image-guided streamline placement. *Proc. of SIGGRAPH 96* (1996), 453–460.

[UIL*03] URNESS T., INTERRANTE V., LONGMIRE E., MARUSIC I., GANAPATHISUBRAMANI B.: Effectively visualizing multi-valued flow data using color and texture. *Proc. of IEEE Visualization '03* (2003), 115–121.

[vW91] VAN WIJK J.: Spot noise — texture synthesis for data visualization. *Proc. of SIGGRAPH 91* (1991), 309–318.

[War00] WARE C.: *Information Visualization: Perception for Design*. Morgan Kaufman, 2000.

[WG97] WEGENKITTL R., GROLLER E.: Oriented line integral convolution for vector field visualization via the internet. *Proc. of IEEE Visualization '97* (1997), 309–316.

[WGP97] WEGENKITTL R., GROLLER E., PURGATHOFER W.: Animation flowfields: Rendering of oriented line integral convolution. *Proc. of IEEE Computer Animation '97* (1997), 15–21.

[WK95] WARE C., KNIGHT W.: Using visual texture for information display. *ACM Transactions on Graphics 14*, 1 (Jan. 1995), 3–20.

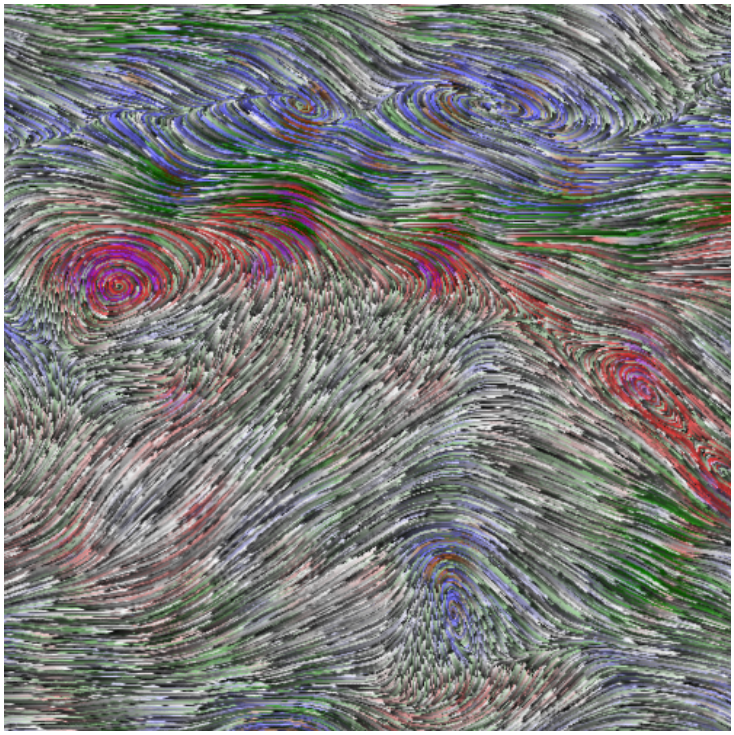

Figure 7: *Using luminance to depict flow direction and color to represent multiple distributions.*