Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 05-014

Flow Visualization Using Natural Textures

Timothy Matthew Urness, Viictoria Interrante, Ellen Longmire, and
Ivan Marusic

April 15, 2005

FLOW VISUALIZATION USING NATURAL TEXTURES

Timothy Urness
Victoria Interrante
Ellen Longmire
Ivan Marusic

University of Minnesota
Minneapolis, Minnesota

April 2005

The use of natural textures provides a richly diverse set of possibilities for the visualization of flow data. In this paper, we present methods that utilize the qualities and attributes of natural textures to visualize multiple scalar distributions and multiple vector fields obtained across a 2D domain in a turbulent boundary layer flow. First, we illustrate how different attributes of textures can represent scalar quantities along streamlines. We then present a technique that allows for the perception of two separate vector fields within the same image by utilizing different textures. Finally, we illustrate how textures have the ability to indicate specific regions of interest within flow images.

1 Introduction

Textures have traditionally been used to visualize vector fields for the purpose of analyzing the form and behavior of flow consistent with theoretical models and to infer the underlying behavior of experimentally-generated flow fields. The use of textures allows for a consistent and highly-detailed representation of a vector field, allowing an observer to both analyze and better understand the dynamics of fluid flow.

In addition to local velocity magnitude and direction, researchers in turbulent flow have long been interested in a deeper understanding of additional vector and scalar distributions that contribute to theories of drag and the formation of vortices and vortex packets. Specific scalar fields such as *strain rate*, used for stretching or compression, *Reynolds shear stress*, used to characterize regions where drag is generated in turbulent boundary layers, and *swirl*, used to analyze rotation in the form of a coherent vortex, are significant components for the analysis of turbulent flow. Additionally, the *vorticity* vector field measures the curl of velocity and is an important component to current theories of turbulent flow. With few exceptions, classical texture-based techniques, such as line integral convolution, allow only for vector direction and magnitude to be visualized. Successfully displaying multiple variables has significant applications in analyzing scientific phenomena represented by multi-variate data. As an alternative to representing multiple scalar distributions in a side-by-side manner, it is highly desirable to create unified images that allow for multiple distributions to be interpreted both individually and in the context of one or more of the other distributions.

A goal of this research is to better understand how the properties of textures can effectively be used to represent various components of flow data. Natural textures have the ability to provide a richly diverse set of possibilities that allow various aspects of the underlying flow field to be visualized. We introduce textures to convey this information in a way that preserves the integrity of the vector field while also taking advantage of the many perceptual dimensions that textures can contribute such as regularity, directionality, contrast, and spatial frequency.

We begin with a review of some of the existing texture-based approaches to visualizing vector fields. We continue with a discussion on mapping textures to streamlines to produce images that reflect the flow field and describe how attributes of textures can be used to represent different scalar quantities. We then discuss how a sparse texture created with equally-spaced streamlines can be combined with a different texture to represent two vector fields within a single image. Lastly, we present a method that utilizes natural textures to delineate separate regions within the flow.

2 Related Work

There have been many texture-based techniques developed for flow visualization. Spot Noise [19] synthesizes a high-resolution image by deforming an input image of random spots according to the vector field.

Line integral convolution (LIC) [2] convolves a white-noise input texture according to calculated streamlines. There have been several extensions to these original methods improving running time and image quality [3, 14].

In these traditional methods, however, the flow direction in static images is ambiguous. The technique of OLIC [23] addresses this issue through the use of a ramp-like convolution kernel. Computation time is improved with the introduction of FROLIC [22]. Sanna et al. [12] also address the issue of flow direction in static images using dense streamlines and a luminance ramp to depict flow direction.

Kiu and Banks [10] propose the use of multi-frequency input textures along with extended filter kernel lengths to produce an image that indicates fluctuations in velocity magnitude. Khouas, Odet, and Friboulet [7] simulate a 3D fur-like texture using a 2D autoregressive synthesis method in order to represent 2D flow.

In analyzing textures for information display, Ware and Knight [21] 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.

Verma, Kao, and Pang [20] introduce the concept of texture-mapping streamlines in a technique called PLIC. Taponecco and Alexa [15] use a texture synthesis approach to visualize vector fields. This technique also has the ability to represent scalar values through the use of adjusting the size of the sample texture.

Mapping multiple distributions in addition to a flow field has been addressed by several visualization techniques. Kirby, Marmanis and Laidlaw [9] utilized techniques inspired by oil painting to combine several layers of glyphs to represent up to three or more different variables in addition to the basic flow field. Urness et al. [17] discuss the use of contrast and mean luminance to represent an additional scalar distribution and present an embossing technique to encode an out-of-plane component of a 3D vector field over a 2D domain. Multiple colors can also be combined with textures to visualize multiple distributions over a 2D flow field [18]. Several other techniques have been introduced to display a scalar field in addition to flow including bump mapping [11] and surface deformation [13].

3 Texture Mapping Streamlines

With few exceptions, LIC textures are predominately used in flow visualizations. While effective, LIC textures lack the richly diverse set of possibilities that can be obtained through the use of natural textures. In the next section, we illustrate the capabilities of natural textures as they are applied to streamlines.

3.1 Geometry

Applying a texture to a streamline requires that the streamline be extended to include width as textures are inherently 2D and do not project well to streamlines or single pixels [3, 20]. We construct a *thick streamline* by first calculating a 1D streamline given the vector field that defines the flow to be visualized. The streamline is then given width by considering the normal component to the streamline at each point. A user-specified width is multiplied by the normal component to give the location for each point of the thick streamline. This amount is added to both sides of every pixel in the streamline resulting in a thick, 2D streamline (figure 1).

An adaptive step size is used during the streamline integration computation to construct polygons that can effectively represent the streamline around areas of high curvature [14]. Using the fourth-order Runge-Kutta

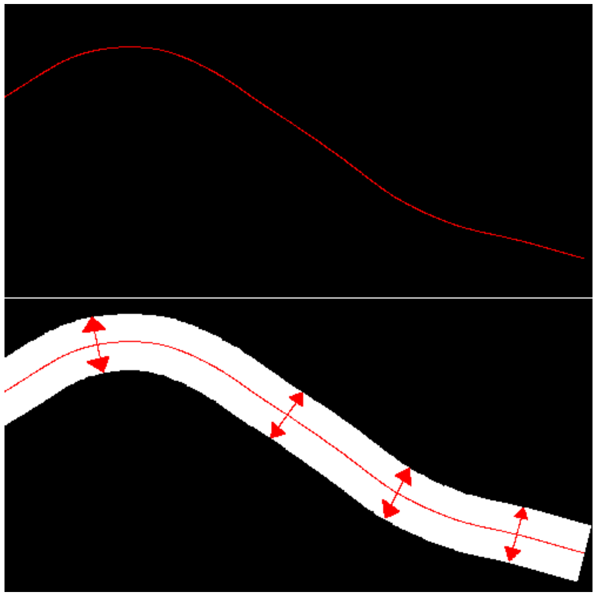

Figure 1: A thick streamline is constructed by first calculating a 1D streamline (top). The normal component at each point in the streamline is multiplied by a user-defined width to calculate the coordinates for the thick streamline (bottom).

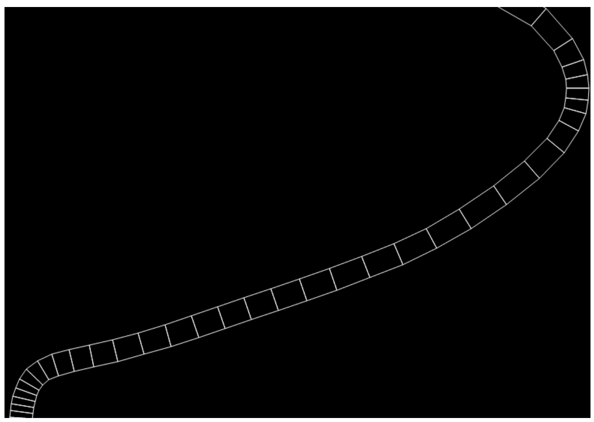

Figure 2: Polygons are generated according to an adaptive step-size algorithm that allows for smaller polygons to be generated around areas of high curvature.

formula and given a user-defined error tolerance, an adaptive step size approach chooses a large enough step size to define each polygon while observing the tolerance specified by the user. The effect of this approach is that smaller polygons are generated in areas of high curvature (figure 2) .

3.2 Flow Fields

Controlling streamline density allows an entire field of thick streamlines to be created and equally spaced so that applied textures can be perceived. Turk and Banks [16] first address the issue of streamline density. We employ the algorithm developed by Jobard and Lefer [6] for ease of implementation and efficiency purposes.

Once an initial seed point in the 2D domain is selected randomly, a streamline is calculated in both the positive and negative direction. The 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. A new seed point is generated by randomly selecting a point on the defined streamline and moving it a distance greater than the width of the thick streamline in the direction normal to the flow at this point. Controlling the distance of a new seed point from the previous streamline allows flexibility in the density of the streamlines of the resulting image. To generate an image of dense streamlines, the seed point should be at a distance that is approximately the thick streamline width from the previously defined streamline. A distance that is greater than the streamline width will create more space between streamlines and result in a sparse final image. The algorithm continues until no more valid seed points can be found.

We have found it beneficial to construct streamlines with maximum possible length when creating the final images presented in the paper. Streamlines that do not have a sufficient length are not displayed and another seed point is used. To avoid artifacts that may occur with a repeated texture on the same streamline, a sufficiently large texture sample is used. To avoid artifacts that may occur with repeated texture being applied at the same interval on neighboring streamlines, a random texture offset is used when constructing the first texture-mapped polygon of the thick streamline. These artifacts could also be avoided by synthesizing the texture separately or along each streamline. Additionally, where portions of streamlines overlap, pixels are assigned an opacity value of zero, giving priority to streamlines already defined. The result is the ability for streamlines to effectively "merge" due to convergence or divergence of the flow but not to obstruct a previously placed streamline (figure 3).

Figure 3: Using texture-mapped thick streamlines to visualize a flow field.

3.3 Texture Outline

Texture-mapping a natural texture to a field of thick streamlines may not create an effective visualization if the orientation of the applied texture is not obvious. Figure 4 (top) shows the result of applying an anisotropic texture to streamlines and the ambiguous orientation of streamlines that results. The orientation of the

Figure 4: An illustration of texture outlining used to disambiguate streamline orientation. Top: a birdseed texture applied to streamlines. Middle: the outlining texture applied to streamlines. Bottom: combination of the above two images.

streamlines can be specified by combining the texture with an outline of the calculated streamlines. The outline of the streamlines is constructed by mapping an *outlining texture* to the calculated streamlines defined by the vector field. The outlining texture consists of a luminance ramp, from black to white, emanating from each side of the texture. The intention is to mimic a diffuse lighting effect that would be created if the thick streamline were three dimensional and tubular in shape. The effect of applying this outlining texture to streamlines is displayed in figure 4 (middle). Finally, the two images can be overlaid allowing the orientation of the flow field to be displayed (figure 4 bottom).

It is important to limit the proximity between streamlines in creating the outlined streamlines texture. If the outlines of thick streamlines are allowed to overlap, areas of high overlap produce distracting regions when the luminance ramp of the streamline is abruptly halted where one streamline overlaps another. To avoid this artifact, the computation of a streamline is stopped if it approaches another streamline within one half the width of the thick streamline.

3.4 Texture Attributes

We have the ability to show a scalar field in addition to the flow field by changing how the texture is mapped to the streamline. The ability to choose texture parameters freely independent of polygonal geometry provides a great amount of flexibility in depicting scalar quantities.

Starting at the beginning of the streamline, the geometric coordinates for the thick streamline are calculated and vertices for the first polygon are defined. The length of the texture, u, or the width of the texture, v, can be scaled according to the a scalar value and applied to the polygon. Texture continuity between polygons is preserved by ensuring that the next polygon starts with the texture coordinates most recently used by the previous polygon. Figure 5 shows a pine texture representing a scalar distribution, ranging from low to high from the top of the image to the bottom, by scaling the u component of the texture while the v component remains constant.

Proportionally varying both the u and v components of the texture influences the relative scale of the texture. This technique creates a difference in the spatial frequency of the texture to reflect the magnitude of a scalar distribution. Figure 6 shows how the scale of a texture can be used to display the magnitude of Reynolds shear stress – a scalar field used to characterize regions where drag is generated in turbulent boundary layers.

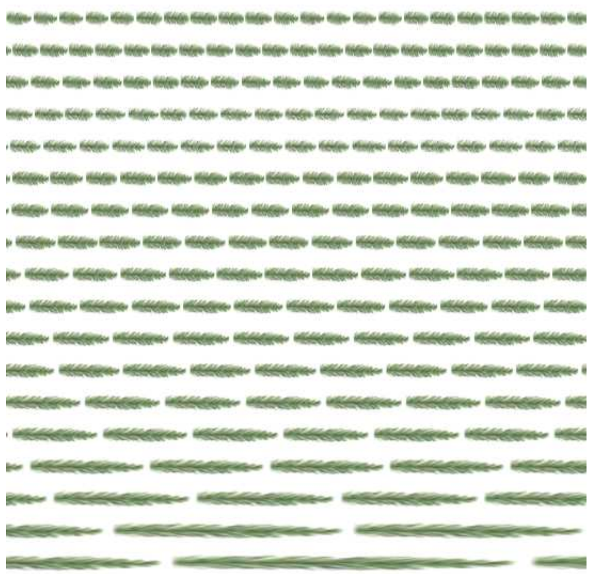

Figure 5: Texture parameters can be adjusted to display a scalar distribution in addition to the vector field. The u component of the texture is mapped according to an arbitrary scalar component that increases from the top of the image to the bottom.

4 Multiple Textures

Texture-mapping streamlines gives great flexibility in the number of different appearances that a vector field representation can have. Figure 7 shows a sample of how natural textures are capable of many different appearances when applied to a circular flow. We can use this flexibility and diversity of appearance to represent multiple quantities within the same image.

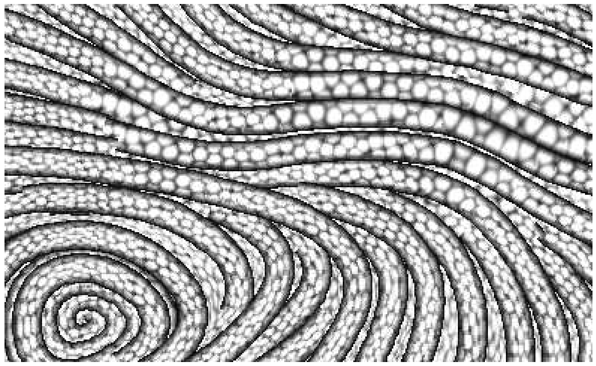

Figure 6: An illustration of using texture attributes to represent a scalar distribution. The scale of the texture is directly related to Reynolds shear stress – a scalar field used to characterize regions where drag is generated in turbulent boundary layers.

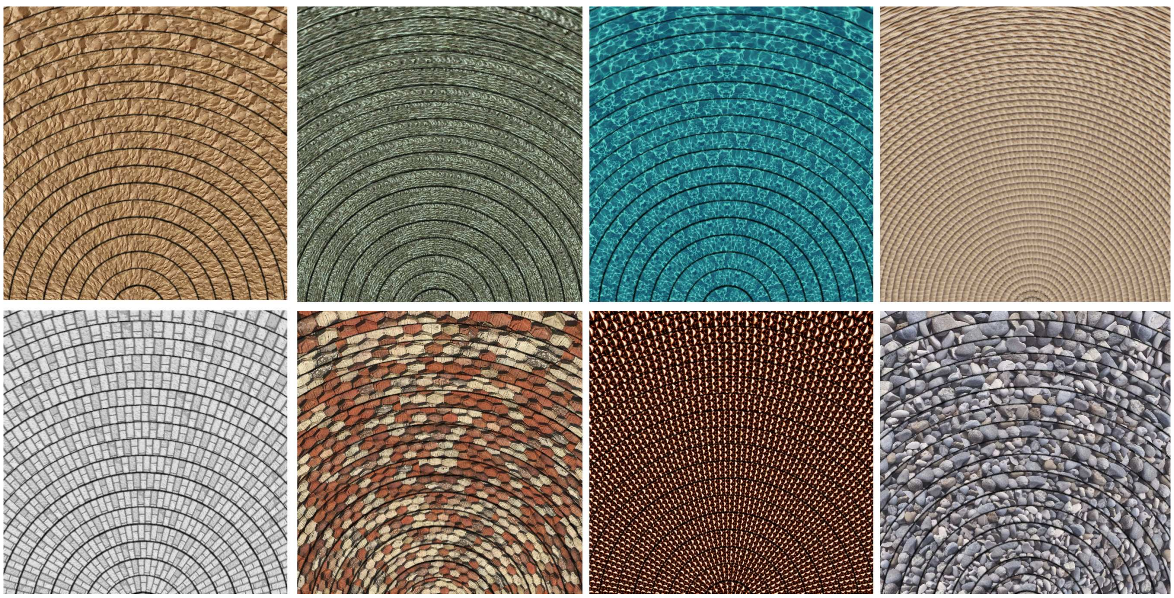

Figure 7: Examples of the diversity of natural textures that can be applied to a vector field. A circular flow is used demonstrate each example.

4.1 Multiple Vector Fields

We introduce a technique that facilitates the analysis of the form and behavior of a vector field within the context of an additional vector field. There are many applications in fluid dynamics for the analysis of two vector fields on the same domain. The relationship between a vorticity vector field and velocity vector field within a turbulent flow is used as an example in this section.

Creating an image that accurately displays the nature of two related vector fields in a single image is a challenging problem. It is necessary to be able to differentiate separate vector components while maintaining the perception of the pattern in each field with the ability to understand how the different fields interact with, relate to, and are unique from one another. We present a method in which two separate texture images are

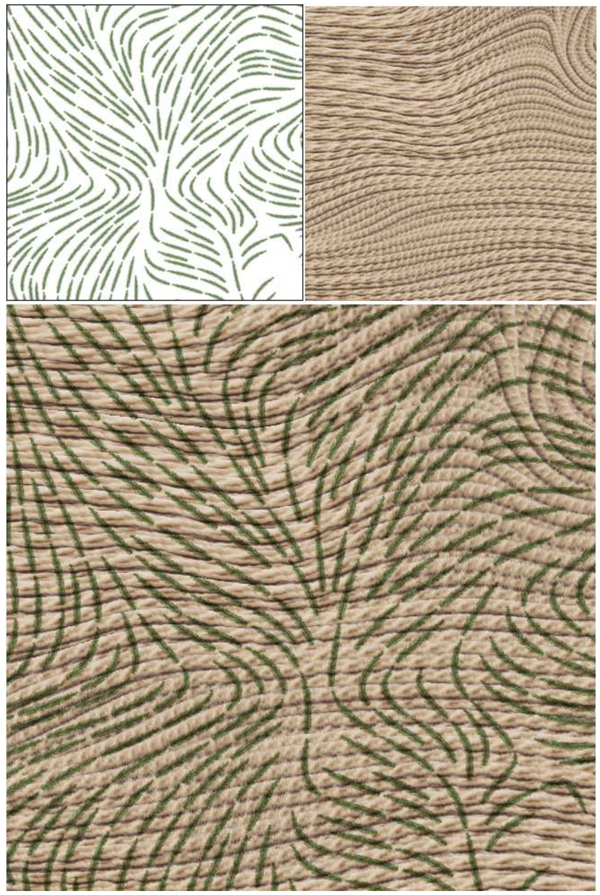

Figure 8: Overlaying two vector field representations. Top left: The overlaying image. The pine texture represents the in-plane vorticity vector field. Top right: The underlying image. The yarn texture represents the in-plane velocity vector field. Bottom: the composited image. The continuation of the texture patterns and spaced streamlines allow for both vector fields to be viewed simultaneously.

combined to display two related vector fields. The terms *overlaying* and *underlying* are used to describe the two layers of vector field images that comprise the final image.

Since two textures cannot co-exist at the same location, we employ an equally-spaced streamline algorithm with a large distance between streamlines to depict the overlaying vector field. We have found success in using glyph-like textures, such as the pine texture, to represent the overlaying vector field as the orientation and direction of the vector field are easy to perceive without the need to use an outline texture (figure 8 top left). Both vector fields in this example have been magnified and interpolated to a scale that allows the sparse streamlines to cover the domain adequately without sacrificing accuracy by not displaying streamlines over the entire domain.

The underlying texture has more flexibility in the type of texture and spacing of the streamlines available to represent the flow accurately. A texture with dense streamlines works well to display the flow field. In figure 8 (top right) a yarn texture is used with densely packed streamlines. The *u* component of the texture is used to reflect velocity magnitude which results in the yarn being “stretched” in areas of high vector magnitude.

The composite image of the overlaying and the underlying images is composed on a pixel-by-pixel basis.

Final pixel values are calculated by taking the normalized product of respective pixel values from the two individual vector field images. The long texture-mapped streamlines give a unique and natural continuation to the vector field which allows for the two vector fields to be visualized simultaneously.

This technique allows the careful analysis of how velocity and vorticity may be related. As the flow data is sampled from a turbulent boundary layer where flow is from left to right, the predominant rotation (caused by wall-induced shearing) should yield “vertical” vorticity lines. This is evidenced in figure 8 (bottom) by the vertical nature of the vorticity lines denoted by the pine texture compared to the more horizontally oriented velocity lines denoted by the yarn texture.

To increase the perception of both flow fields within the same image, we have found the best results using two textures that contain unrelated color schemes.

4.2 Texture Splicing

Allowing different appearances within an image can be useful in preserving the fidelity of region boundaries while avoiding the introduction of discontinuity artifacts. The goal of *texture splicing* is to produce an

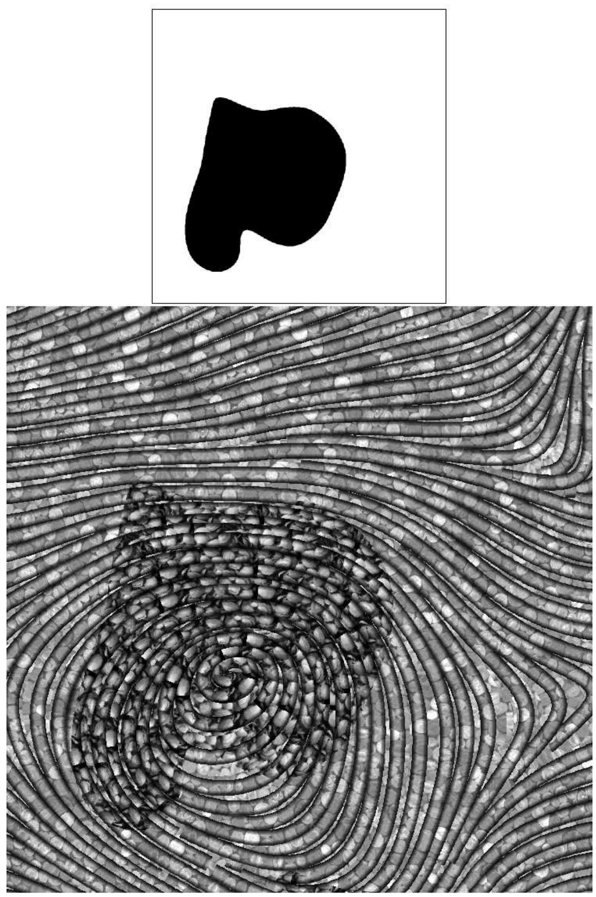

Figure 9: Top: Binary region of high swirl strength – a quantity used to identify coherent vortices. Bottom: A greyscale coin texture and shell texture are spliced to delineate the region of high swirl strength.

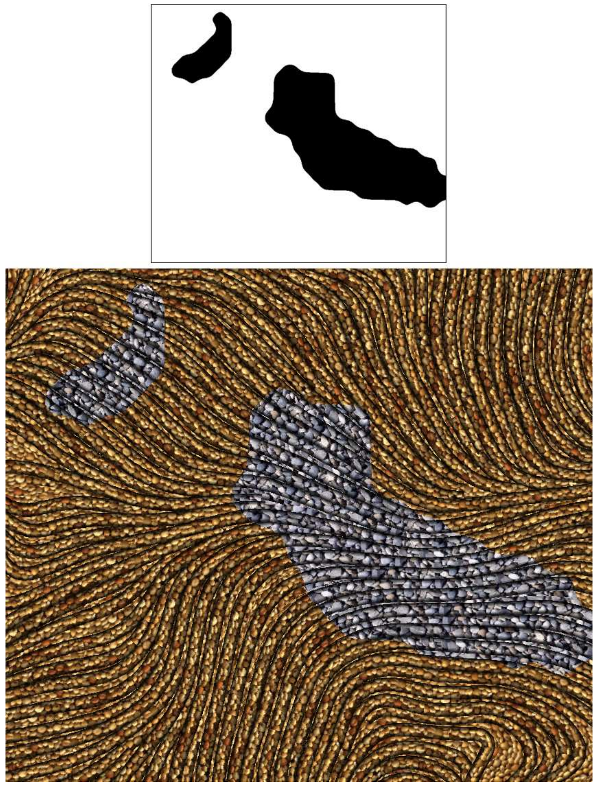

Figure 10: Top: Boundary regions of potential vortex packets defined by a feature extraction algorithm. Bottom: A birdseed texture and a stone texture are spliced to specify the boundaries.

image that achieves a precise depiction of the spatial extents of discrete regions while not detracting from the perception of flow. In the examples illustrated in this section, the visualization is designed to delineate boundaries of significant structures or *regions of interest* within a 2D image of turbulent flow.

The process of creating the composited image can be conceptualized in three separate steps. First, two textures are selected and the same vector field is used to create two separate flow images. Next, when necessary, an outlined texture of the same computed streamlines is created and overlaid on both images to clarify streamline orientation. In order to avoid discontinuities of the flow, it is important to have both texture images and the streamline outlines generated from the same computed streamlines. Lastly, the images are composited or “spliced” on a pixel-by-pixel basis, as each pixel in the final image is sampled from the respective source texture determined by the location of the pixel with respect to the region to be delineated. For implementation purposes, it is advantageous for all three conceptual steps to be combined into an efficient one-pass algorithm in which the textures and texture outline are combined then applied along the streamline only in the appropriate region.

Figure 9 shows how subtle differences in greyscale textures allow for a specific region to be highlighted. In this case, a region of significant *swirl strength*, a measure of rotation about a central axis [24], is constructed using a shell texture. A texture comprised of coins, similar in size to the shell texture but different in luminance properties, is used for the other texture which allows for the swirl center to be visualized clearly. The spiraling streamlines around the area of swirl strength denote that the relative speed of the observer matches the convection velocity of this particular swirl center.

There is great flexibility for image appearances using different natural textures. Figure 10 illustrates an example using two unrelated textures, a birdseed texture and a stone texture. The region delineated by the stone texture represents boundaries of a potential vortex packet identified with a specific feature extraction algorithm [4]. The size of the components and luminance of each texture chosen are similar so they are not prominent factors in the perception of the difference in texture. While the two textures are most differentiable by color, the regions are still noticeable when the image is viewed as a greyscale image. The natural qualities allow each texture to be distinguishable while the flow of the streamlines remains easily perceivable.

5 Conclusion

In this paper we presented methods that utilize the qualities and attributes of natural textures to visualize scalar distributions and vector fields related to a planar velocity field in a turbulent flow. Our ultimate goal in this work is to enable researchers to obtain a succinct, meaningful visual summary of their data in a manner that is intuitive and leads to a fundamental understanding of the presence of individual features within a flow field.

We have shown how natural textures can effectively be mapped to streamlines and how additional scalar distributions can be represented through attributes such as scale and stretching of the texture. Additionally, we presented a method to use natural textures to display multiple vector fields on a 2D domain. Lastly, we presented texture splicing, a method that accurately displays the extent of specific regions identified within a flow field.

6 Acknowledgements

We would like to thank Bharathram Ganapathisubramani for his assistance in obtaining the experimental data visualized in this paper. The textures in this paper were used courtesy of bettyschristmashouse.com and www.bigfoto.com and obtained from the Brodatz texture album [1]. This research was supported by a grant from the National Science Foundation (CTS-0324898).

References

[1] Phil Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.

[2] Brian Cabral and Leith (Casey) Leedom. Imaging vector fields using line integral convolution. In *Proceedings of SIGGRAPH 93*, pages 263–269. ACM, ACM Press / ACM SIGGRAPH, 1993.

[3] Willem C. de Leeuw and Jarke J. van Wijk. Enhanced spot noise for vector field visualization. In *Proceedings of IEEE Visualization 95*, pages 233–239, 1995.

[4] Bharathram Ganapathisubramani, Ellen K. Longmire, and Ivan Marusic. Characteristics of vortex packets in turbulent boundary layers. *Journal of Fluid Mechanics*, 478:35–46, 2003.

[5] Christopher G. Healey. Choosing effective colours for data visualization. In *Proceedings of IEEE Visualization 96*, pages 263–270, 1996.

[6] Bruno Jobard and Wilfrid Lefer. Creating evenly-spaced streamlines of arbitrary density. In *Proceedings of the 8th Eurographics Workshop on Visualization in Scientific Computing*, pages 43–55, 1997.

[7] Leila Khouas, Christophe Odet, and Denis Friboulet. Vector field visualization using furlike texture. In *Eurographics/IEEE TCVG Symposium on Data Visualization*, pages 35–44, 1999.

[8] Gordon Kindlmann, Erik Reinhard, and Sarah Creem. Face-based luminance matching for perceptual colormap generation. In *Proceedings of IEEE Visualization 2002*, pages 299–306, 2002.

[9] Robert M. Kirby, H. Marmanis, and David H. Laidlaw. Visualizing multivalued data from 2d incompressible flows using concepts from painting. In *Proceedings of IEEE Visualization 99*, pages 333–340, 1999.

[10] Ming-Hoe Kiu and David C. Banks. Multi-frequency noise for lic. In *Proceedings of IEEE Visualization 96*, pages 121–126, 1996.

[11] Andrea Sanna and Bartolomeo Montrucchio. Adding a scalar value to 2d vector field visualization: the blic (bumped lic). *Eurographics 2000 Short Presentations Proceedings*, pages 119–124, 2000.

[12] Andrea Sanna, Bartolomeo Montrucchio, Paolo Montuschi, and A. Sparavigna. Visualizing vector fields: the thick oriented stream-line algorithm (tosl). *Computers and Graphics*, 25(5):847–855, 2001.

[13] Gerik Scheuermann, Holger Burbach, and Hans Hagen. Visualizing planar vector fields with normal component using line integral convolution. In *Proceedings of IEEE Visualization 99*, pages 255–261, 1999.

[14] Detlev Stalling and Hans-Christian Hege. Fast and resolution-independent line integral convolution. In *Proceedings of SIGGRAPH 95*, pages 249–256. ACM, ACM Press / ACM SIGGRAPH, 1995.

[15] Francesca Taponecco and Marc Alexa. Vector field visualization using markov random field texture synthesis. In *Eurographics/IEEE TCVG Symposium on Data Visualization*, pages 195–202, 2003.

[16] Greg Turk and David Banks. Image-guided streamline placement. In *Proceedings of SIGGRAPH 96*, pages 453–460. ACM, ACM Press / ACM SIGGRAPH, 1996.

[17] Timothy Urness, Victoria Interrante, Ellen Longmire, Ivan Marusic, and Bharathram Ganapathisubramani. Techniques for visualizing multi-valued flow data. In *Eurographics/IEEE TCVG Symposium on Data Visualization*, 2004.

[18] Timothy Urness, Victoria Interrante, Ivan Marusic, Ellen Longmire, and Bharathram Ganapathisubramani. Effectively visualizing multi-valued flow data using color and texture. In *Proceedings of IEEE Visualization 2003*, pages 115–121, 2003.

[19] Jarke J. van Wijk. Spot noise — texture synthesis for data visualization. In *Proceedings of SIGGRAPH 91*, pages 309–318. ACM, ACM Press / ACM SIGGRAPH, 1991.

[20] Vivek Verma, David Kao, and Alex Pang. Plic: Bridging the gap between streamlines and lic. In *Proceedings of the Symposium on Data Visualization 1999*, pages 341–348, 1999.

[21] Colin Ware and William Knight. Using visual texture for information display. *ACM Transactions on Graphics*, 14(1):3–20, January 1995.

[22] Rainer Wegenkittl and Eduard Gröller. Oriented line integral convolution for vector field visualization via the internet. In *Proceedings of IEEE Visualization 97*, pages 309–316, 1997.

[23] Rainer Wegenkittl, Eduard Gröller, and Werner Purgathofer. Animation flowfields: Rendering of oriented line integral convolution. In *Proceedings of IEEE Computer Animation 97*, pages 15–21, 1997.

[24] J. Zhou, R. J. Adrian, R. J. Balachandar, and T. M. Kendall. Mechanisms for generating coherent packets of hairpin vortices in channel flow. *Journal of Fluid Mechanics*, 387:353–396, 1999.