Multi-dimensional Transfer Functions for Interactive 3D Flow Visualization

Sung W. Park* Brian Budge* Lars Linsen* Bernd Hamann* Kenneth I. Joy*

Institute for Data Analysis and Visualization (IDAV)†
Department of Computer Science
University of California, Davis
One Shields Avenue, Davis, CA 95616, U.S.A.

Abstract

Transfer functions are a standard technique used in volume rendering to assign color and opacity to a volume of a scalar field. Multi-dimensional transfer functions (MDTFs) have proven to be an effective way to extract specific features with subtle properties. As 3D texture-based methods gain widespread popularity for the visualization of steady and unsteady flow field data, there is a need to define and apply similar MDTFs to interactive 3D flow visualization. We exploit flow field properties such as velocity, gradient, curl, helicity, and divergence using vector calculus methods to define an MDTF that can be used to extract and track features in a flow field. We show how the defined MDTF can be applied to interactive 3D flow visualization by combining them with state-of-the-art texture-based flow visualization of steady and unsteady fields. We demonstrate that MDTFs can be used to help alleviate the problem of occlusion, which is one of the main inherent drawbacks of 3D texture-based flow visualization techniques. In our implementation, we make use of current graphics hardware to obtain interactive frame rates.

1. Introduction

Flow visualization has always been a significant area of scientific data visualization; it has also been one of the most challenging, especially when looking at volumetric data. A rich variety of applications attach high importance to the visual exploration of 3D flow fields. Many engineering and scientific disciplines including mechanical engineering, physics, chemistry, meteorology, geology, and medicine make use of 3D flow visualization for applications such as aero and fluid dynamics. The measured or simulated flow field can be static or time varying.

Early attempts to visualize flow field data include hedgehog plots, particle tracing, and streamlines. Streamlines have been elaborated to stream ribbons and stream tubes for steady fields [1] and pathlines, timelines, and streaklines for unsteady fields. Texture-based approaches have gained the most popularity recently. This is mostly due to the tremendous progress in graphics hardware, which is now highly amenable to texture-based approaches. The new hardware supports storage and processing of 2D and 3D textures of steadily increasing size as well as programmability, which makes possible the extension of the graphics processing unit to general purpose computation. In Section 3, we review 3D flow visualization techniques and describe the approaches we integrated with to our new feature-extraction method.

Volume rendering approaches for 3D scalar fields have made use of 3D textures in graphics hardware even before hardware-assisted flow visualization methods emerged. In these scalar field volume rendering approaches, color and opacity are assigned to scalar values using transfer functions. The color and opacity assignments can be further refined and improved by using multi-dimensional transfer functions (MDTFs), which allow for selection and extraction of very distinct features without having them occluded by features with similar, yet slightly different characteristics.

We present MDTFs for 3D flow visualization. The transfer functions are based on vector field properties derived from vector field calculus. The dimensions of

* {sunpark|bcbudge|llinsen}@ucdavis.edu, {hamann|joy}@cs.ucdavis.edu
† http://graphics.cs.ucdavis.edu

our transfer functions are given by velocity magnitude, velocity gradient tensor determinant, curl magnitude, helicity, and divergence of 3D flow fields [2]. In Section 4, we explain and discuss the use of these vector field properties, and in Section 5, we combine the derived properties to form an MDTF. The MDTF approach can be combined with many 3D flow visualization techniques. In Section 6, we describe how our transfer function can be applied to GPU-based 3D texture advection flow visualization methods.

Using our MDTF, we can interactively explore 3D flow fields and extract/select features of well-defined behavior. Our approach helps to overcome the occlusion problems that texture-based flow visualization techniques generally suffer from. In Section 7, we show results of our approach for steady as well as unsteady 3D flow fields. In particular, we show how significant features in an unsteady data set can be tracked and how they evolve over time.

2. Related Work

Our work utilizes transfer functions to aid in feature extraction. Transfer functions are crucial components of volume visualization techniques [3, 4, 5]. In recent years, a large number of improvements have been made to make transfer functions produce higher quality images as well as making them easier to use.

Bergman et al. [6] described making colormap selection an interactive process. The colormaps provided to the user are decided by the rule-based approach depending on the type of data being visualized. He et al. [7] and Kindlmann and Durkin [8] introduced improved methods for generating transfer functions. He's method generates transfer functions by use of a stochastic optimization process. The process relies on a user defined fitness function, or user input in order to decide the quality of transfer functions. Kindlmann's method automatically attempts to search out isovalues which might describe material boundaries.

Specifically, our work makes use of multi-dimensional transfer functions, or MDTFs. MDTFs can be a very powerful tool, yet can be tricky to deal with. Kniss et al. [9] described a widget based method for dealing with multi-dimensional transfer functions. Their method supports interactive exploration of multivariate volumes. Morris and Ebert [10] used multi-dimensional transfer functions combined with photographic data to generate intuitive colors for volume rendering of medical data. Hadwiger et al. [11] applied separate transfer functions to each material on a per-fragment basis in order to obtain color values for each material.

Our main contribution is the combination of vector field properties with MDTFs to provide useful feature extraction. Feature extraction of scalar datasets has been of interest to the visualization community for a long time. More recently, work has been done on feature extraction of vector field data. The approach of Verma et al. [12] finds critical points in order to intelligently place seeds for streamline generation in 2D vector fields. The method of Mann and Rockwood [13] calculates singularities in 3D vector fields. These singularities can be useful in limited situations for classifying flow fields. Wischgoll and Scheuermann [14] described an algorithm for locating closed streamlines in 3D vector fields, and Mahrous et al. [15] presented a method for topological segmentation of 3D steady flow fields.

While these methods describe geometric separations, it is also possible to extract features of certain properties. Suzuki et al. [16] derived an "S-map" to assign significance to portions of the volume. The S-map is generated by approximating critical points and using them to assign importance. Post et al. [17] discussed this technique further and mentioned many possible values to extract including vorticity and helicity. A method by Gray et al. [18] maps curl and divergence to colors of an isosurface extracted from velocity magnitude. Our technique extracts features in the same vein, and we use similar quantities to generate MDTFs that allow useful feature extraction.

3. 3D Flow Visualization

Our method permits incorporation of 3D flow visualization algorithms in order to enhance visualization of the volume. There are many techniques for visualizing 3D flow, with the simplest being direct flow visualization. Some common examples are the hedgehog plot, and mapping RGB colors to vectors in the field. These approaches tend not to be very helpful in 3D because of occlusion issues. Another class of algorithms for flow visualization is made up of the geometric approaches. These are integration approaches such as streamlines, streaklines, timelines, and pathlines. They are interesting from the standpoint that long term information is presented, and that the information can be sparse and directed at areas of importance (such as through seeding strategies [12]). Recently, texture-based methods have become very popular, especially for 2D fields, mainly because of their ability to show global information. The most common examples are line integral convolution (LIC) [19, 20] and texture advection methods, including image based flow visualization (IBFV),

along with their many variants. Feature-based flow visualization involves indirectly visualizing flow. Usually this involves calculating quantities based upon the vector field data. This paper's foundation is an example of feature-based flow visualization. For more information on these methods, see the survey paper by Laramee et al. [21].

Our technique makes possible the combination of feature-based flow visualization with the other types of flow visualization. In Section 6, we discuss the integration of our method with the GPU-based 3D texture advection method based on the method by Weiskopf and Ertl [22]. By varying different parameters, we show 3D LIC-type images of flow volumes. We also combine our methods with the 3D IBFV approach described by Telea and van Wijk [23] to produce 3D IBFV-type images.

4. Vector Field Calculus

A 3D flow field is defined by a trivariate function. Let $\mathbf{F} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a function such that

$$\mathbf{F}(x,y,z) = \begin{pmatrix} F_1(x,y,z) \\ F_2(x,y,z) \\ F_3(x,y,z) \end{pmatrix} ,$$

where F_1, F_2, and F_3 represent the three components in the directions of the three coordinates x, y, and z. Typically, the coordinates are Cartesian. In many practical settings, the discrete version of a flow field is obtained by measuring or simulating the flow components at equidistant integer sample points (i,j,k) in each of the three coordinate directions, forming a structured rectilinear grid. In the following, we derive properties from vector field calculus as described in [2].

4.1. Velocity Magnitude

A straightforward property of a flow field that has been used by other visualization approaches is velocity magnitude. The velocity magnitude is calculated at a discrete position (i,j,k) by the Euclidean norm $\|\cdot\|_2$, i. e.,

$$|\mathbf{F}(i,j,k)| = \left\| \begin{pmatrix} F_1(i,j,k) \\ F_2(i,j,k) \\ F_3(i,j,k) \end{pmatrix} \right\|_2 . \quad (1)$$

The velocity magnitude can be used by our MDTF to distinguish between regions of high and low flow/velocity.

4.2. Gradient

To detect sudden changes in the flow, one is required to consider the velocity gradient of the flow. The gradient field of function $\mathbf{F}$ is

$$\nabla\mathbf{F}(x,y,z) = \left(\frac{\partial\mathbf{F}(x,y,z)}{\partial x}, \frac{\partial\mathbf{F}(x,y,z)}{\partial y}, \frac{\partial\mathbf{F}(x,y,z)}{\partial z} \right) , \quad (2)$$

which defines a tensor represented by a 3×3-matrix.

At a discrete position (i,j,k) we can approximate the nine entries of the matrix by central differencing, for example,

$$\frac{\partial F_1(i,j,k)}{\partial x} = \frac{1}{2}\left(F_1(i+1,j,k) - F_1(i-1,j,k)\right) . \quad (3)$$

Gradient Magnitude

The tensor can be investigated by performing an eigenanalysis of the matrix. Each eigenvalue of the matrix tells us the stretch factor of the gradient tensor in the direction of its eigenvector, see Figure 1. For the generation of our MDTF, it would be useful to exploit this property, e. g., to extract features with high gradients. A full eigenanalysis for each discrete vertex location in our grid is not practical however, especially when dealing with unsteady fields, where all data processing needs to be done in real time as the data is streaming in. Instead the product of the three eigenvalues of a matrix can be computed by taking the determinant of the matrix. Thus, $|\nabla\mathbf{F}(x,y,z)|$ can be used to classify regions due to the dimension of the gradients. Gradient magnitude is given by the determinant

$$|\nabla\mathbf{F}(x,y,z)| = \begin{vmatrix} \frac{\partial F_1(x,y,z)}{\partial x} & \frac{\partial F_1(x,y,z)}{\partial y} & \frac{\partial F_1(x,y,z)}{\partial z} \\ \frac{\partial F_2(x,y,z)}{\partial x} & \frac{\partial F_2(x,y,z)}{\partial y} & \frac{\partial F_2(x,y,z)}{\partial z} \\ \frac{\partial F_3(x,y,z)}{\partial x} & \frac{\partial F_3(x,y,z)}{\partial y} & \frac{\partial F_3(x,y,z)}{\partial z} \end{vmatrix} . \quad (4)$$

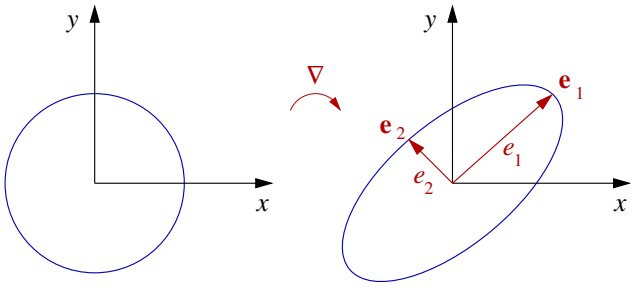

Figure 1. Influence of velocity gradient tensor determined by eigenanalysis: Unit circle gets stretched by eigenvalues e_1 and e_2 along the direction of the eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$.

Divergence

Another matrix property that is of interest is the trace

of the velocity gradient tensor, i. e., its associated matrix, since it is the sum of the eigenvalues. Indeed, the trace of matrix $\mathbf{trace}(\nabla \mathbf{F}(x,y,z))$ turns out to be the divergence of the field. The divergence of a vector field is a scalar value defined by

$$\mathbf{div}\ \mathbf{F}(x,y,z) = \frac{\partial F_1(x,y,z)}{\partial x} + \frac{\partial F_2(x,y,z)}{\partial y} + \frac{\partial F_3(x,y,z)}{\partial z}\ . \tag{5}$$

The divergence measures the rate of expansion per volume unit, i. e., the difference in inflow and outflow per unit, see Figure 2. Divergence is positive for expanding and negative for compressing flow fields.

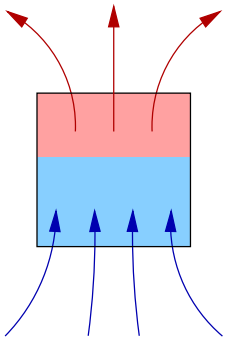

Figure 2. Divergence measures the difference in inflow and outflow per unit.

Curl Magnitude

The curl of a vector field is a measure for the vector field's vorticity (degree of turbulent behavior). It is defined by the vector

$$\mathbf{curl}\ \mathbf{F}(x,y,z) = \begin{pmatrix} \frac{\partial F_3(x,y,z)}{\partial y} - \frac{\partial F_2(x,y,z)}{\partial z} \\ \frac{\partial F_1(x,y,z)}{\partial z} - \frac{\partial F_3(x,y,z)}{\partial x} \\ \frac{\partial F_2(x,y,z)}{\partial x} - \frac{\partial F_1(x,y,z)}{\partial y} \end{pmatrix}\ . \tag{6}$$

We can exploit the magnitude of the curl $|\mathbf{curl}\ \mathbf{F}(i,j,k)|$ at each discrete grid location for our multi-dimensional transfer function to extract swirling features.

Helicity

Also of interest is the curl in the direction of the velocity of a flow field, which is called the helicity, illustrated in Figure 3. The helicity is a scalar function and can be computed as the dot product of the curl and the velocity:

$$\mathbf{heli}\ \mathbf{F}(x,y,z) = \mathbf{curl}\ \mathbf{F}(x,y,z) \cdot \mathbf{F}(x,y,z)\ . \tag{7}$$

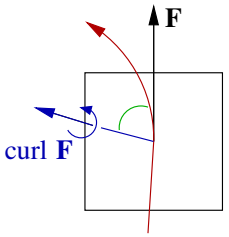

Figure 3. Helicity is curl in direction of velocity F, while curl measures vorticity.

5. MDTFs

Transfer functions provide a means to selectively visualize different aspects of a volume by defining a function $\mathbf{T} : \mathbb{R}^N \rightarrow \mathbb{R}^M$, where usually $N = 1$ (the data is scalar), and $M = 4$ (the function is usually RGBA valued, where RGB is color and A is opacity). In the vector field context, vector/velocity magnitude has been most commonly used for the definition of a one-dimensional transfer function. To enhance the capability to extract more subtle features, different properties of a vector field can be incorporated generating an N-dimensional transfer function to represent each of the N properties. Characteristic properties of a vector field are those mentioned in the previous section, i. e., velocity magnitude, gradient tensor determinant, curl magnitude, helicity and divergence. The combination of these five scalar values leads to a five-dimensional transfer function.

In order to define an MDTF, we map each of the N scalar magnitudes to color channels R, G, and B and opacity values α by using one-dimensional transfer functions $\mathbf{T}_i : \mathbb{R} \rightarrow \mathbb{R}^4$, $i = 1, \ldots, N$. We combine the one-dimensional transfer functions $\mathbf{T}_i$ to an N-dimensional transfer function $\mathbf{T} : \mathbb{R}^N \rightarrow \mathbb{R}^4$, according to the equations

$$\mathbf{T}_{RGB}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{T}_{i,RGB}(x_i) \cdot \mathbf{T}_{i,\alpha}(x_i)$$

and

$$\mathbf{T}_{\alpha}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{T}_{i,\alpha}(x_i)\ ,$$

where $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N$. The color values $\mathbf{T}_{i,RGB}$ assigned by the individual one-dimensional transfer functions are averaged in a weighted fashion to define the color $\mathbf{T}_{RGB}$ assigned by the MDTF, where the weights are given by the individual opacities $\mathbf{T}_{i,\alpha}$. The opacity values $\mathbf{T}_{i,\alpha}$ assigned by the individual one-dimensional transfer functions are averaged as well, to

define the opacity $\mathbf{T}_\alpha$ assigned by the MDTF. $\mathbf{T}$ allows one to visualize each component distinctly, but also makes possible blending of the values in overlapping regions. If extracted features appear to be too dark, one can increase the intensity by multiplying the opacity with a constant factor.

The user interface to manipulate our MDTF is shown in Figure 4(a). For each of the five magnitudes from Section 4, we provide the user with a bar that allows the user to pick and color the interesting parts of each property. For example in the third bar of Figure 4(a), we select regions of high curl magnitude and assign to them an orange/red color, while in the fifth bar of Figure 4(a) we distinguish between regions of high negative and high positive divergence by assigning to them the colors blue and white, respectively. The colors and opacities selected using the individual bars are combined to an MDTF as described above.

6. Implementation

We incorporated MDTFs into a GPU-based 3D texture advection application in C++ using NVIDIA's GeForce FX GPU. The GPU-based advection and rendering system we used for steady and unsteady flow is similar to the one presented by Weiskopf and Ertl [22].

The 3D texture advection visualization goes through three basic steps: texture advection, texture blending, and volume rendering.

For texture advection, 3D volume is advected a slice at a time, using the first-order Eulerian scheme

$$\mathbf{x}(t - \Delta t) = \mathbf{x}(t) - \Delta t \mathbf{F}(\mathbf{x}(t), t) \ ,$$

where $\mathbf{F}(\mathbf{x}(t), t)$ is our vector field evaluated at position $\mathbf{x}(t)$ at time t. The integration step size is Δt.

For texture injection and blending, we use the extended blending equation

$$\mathsf{T}_t = W_t \cdot \mathsf{T}_{t-\Delta t} + V_t \cdot \mathsf{I}_t \ ,$$

where the previously advected texture $\mathsf{T}_{t-\Delta t}$ and the injection texture I_t are multiplied by multi-component weights W_t and V_t to produce the newly advected texture T_t. In our implementation, we tested two different settings for noise texture blending. In one case we performed an affine combination of values from the previously advected texture to produce a fully opaque LIC type volume. In the other case, we used space-variant scalar injection weights as described in 3D IBFV [23].

For each fragment rendered, properties of the field are found using Equation (1) and approximating Equations (4) – (7) by using central differencing as in Equation (3). These properties are then evaluated by the MDTF in Equations (5) and (5).

To render the final image, a direct 3D texture-based volume rendering is used, by rendering view-aligned slices.

7. Results

We tested our approach for steady and unsteady datasets. For steady flow data, we examined a tornado dataset [24] of size 128^3. For the unsteady case, we examined the CFD simulation of five jets consisting of 2000 timesteps of 128^3 vector field data[1].

Figure 6 shows the components we examined: velocity magnitude, determinant of the gradient tensor, magnitude of curl, helicity, and divergence, respectively. Solid opaque surfaces are rendered without texture advection to better show the structure of the properties. Figure 6(a) shows regions of constant velocity magnitude. Figure 6(b) shows high absolute values for the gradient tensor determinant. Figure 6(c) shows a region of high curl magnitude, which yields the structure of the vortex core of the tornado. The helical motion of the tornado is shown in Figure 6(d). In Figure 6(e), the blue regions are regions of high negative divergence, where flow is converging, whereas the white regions are regions of high positive divergence, where flow is diverging. The last image shows the full MDTF, as described in Equations (5) and (5) with $N = 5$. The used color and opacity assignments are shown in Figure 4(a).

Figure 5 shows still images from a real-time animation achieved from GPU-based texture advection methods using MDTFs to highlight features of interest. The parameters used create LIC-type visualizations. The first two images show the tornado data using different MDTFs. The third image shows the five-jet data at a fixed timestep. When animated with texture advection methods, these surfaces not only show interesting structures corresponding to the MDTF, but also show the direction and the movement of the vector field on the surfaces as well.

In comparison to the LIC-type visualization of volumes, we also generated animations combining 3D IBFV with MDTFs. Figure 4(b) shows the results when applied to the tornado dataset. We highlight helicity, magnitude, and divergence in the image.

We also show images from three different timesteps of the five-jet dataset with same MDTFs, highlighting properties of interest in Figure 7. The blue and white regions specify areas of high divergence with negative and positive sign, respectively. Red regions highlights

1 Dataset courtesy of Kwan-Liu Ma, IDAV, University of California, Davis

high curl, yellow regions highlight high gradient determinant, and green regions show the helical behavior.[2]

8. Conclusions and Future Work

Vector field visualization continues to be an important area of active research. It is important not only to see where a particular part of the field advects from one instant to the next, but also to visualize globally portions of the field that have certain behaviors.

We have used various vector calculus quantities that are important for understanding the flow, and we have discussed methods for visualizing these behaviors by using MDTFs. This is beneficial, because it allows us to extract and track features for 3D flow fields. The MDTFs alleviate part of the occlusion problem, which has been the major drawback of 3D flow visualization techniques.

In our current work, we have integrated our feature extraction MDTFs with GPU-based 3D texture advection methods to generate LIC-type images of volumes as well as 3D IBFV-type images of volumes. While they sometimes can add clarity to the visualization, they can sometimes distract from the feature extraction visualization. This is mainly due to still remaining occluding properties of dense vector field visualization techniques.

In future work we may explore integrating geometric techniques with our MDTF method with the hope that the occlusion problems can be avoided even further, while gaining valuable insight from them.

Acknowledgments

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, through the National Partnership for Advanced Computational Infrastructure (NPACI) and a large Information Technology Research (ITR) grant; the National Institutes of Health under contract P20 MH60975-06A2, funded by the National Institute of Mental Health and the National Science Foundation; by a United States Department of Education Government Assistance in Areas of National Need (DOE-GAANN) grant #P200A980307; and through a Hewlett-Packard contribution to a Graduate Student Fellowship. We thank the members of the Visualization and Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis.

2 The full animation is available as an AVI movie, see `http://graphics.cs.ucdavis.edu/~jahsik59/mdtf.avi`

References

[1] S. K. Ueng, K. Sikorski, and K.-L. Ma, "Efficient streamline, streamribbon, and streamtube constructions on unstructured grids," *IEEE Transactions on Visualization and Computer Graphics*, pp. 100–110, 1996.

[2] J. E. Marsden and A. Tromba, *Vector Calculus*. W.H. Freeman, New York, 4th edition, 1996.

[3] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, 1988.

[4] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," in *Computer Graphics (Proceedings of SIGGRAPH 88)*, vol. 22, pp. 65–74, 1988.

[5] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. Graph.*, vol. 9, no. 3, pp. 245–261, 1990.

[6] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish, "A rule-based tool for assisting colormap selection," in *Proceedings of the 6th conference on Visualization '95*, p. 118, IEEE Computer Society, 1995.

[7] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," in *Proceedings of the 7th conference on Visualization '96*, pp. 227–235, IEEE Computer Society Press, 1996.

[8] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *Proceedings of the 1998 IEEE Symposium on Volume Visualization*, pp. 79–86, ACM Press, 1998.

[9] J. Kniss, G. Kindelmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," in *Proceedings of the Conference on Visualization 2001* (T. Ertl, K. Joy, and A. Varshney, eds.), pp. 255–262, IEEE, IEEE Computer Society, 2001.

[10] C. J. Morris and D. Ebert, "Direct volume rendering of photographic volumes using multi-dimensional color-based transfer functions," in *Proceedings of the Symposium on Data Visualisation 2002*, pp. 115–124, Eurographics Association, 2002.

[11] M. Hadwiger and C. B. H. Hauser, "High-quality two-level volume rendering of segmented data sets on consumer graphics hardware," in *Proceedings of IEEE Conference on Visualization 2003*, IEEE Computer Society, 2003.

[12] V. Verma, D. Kao, and A. Pang, "A flow-guided streamline seeding strategy," in *Proceedings of IEEE Conference on Visualization 2000* (T. Ertl, B. Hamann, and A. Varshney, eds.), pp. 163–170, IEEE, IEEE Computer Society Press, 2000.

[13] S. Mann and A. Rockwood, "Computing singularities of 3d vector fields with geometric algebra," in *Proceedings of the 13th IEEE Visualization 2002 Conference*

(R. Moorhead, M. Gross, and K. I. Joy, eds.), pp. 283–290, IEEE Computer Society Press, 2002.

[14] T. Wischgoll and G. Scheuermann, "Locating closed streamlines in 3d vector fields," in *Proceedings of the Symposium on Data Visualisation 2002*, pp. 227–235, Eurographics Association, 2002.

[15] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. Joy, "Topological segmentation in three-dimensional vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 198–205, 2004.

[16] Y. Suzuki, I. Fujishiro, L. Chen, and H. Nakamura, "Case study: hardware-accelerated selective lic volume rendering," in *Proceedings of the 13th IEEE Visualization 2002 Conference* (R. Moorhead, M. Gross, and K. I. Joy, eds.), pp. 485–488, IEEE Computer Society Press, 2002.

[17] F. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualization: Feature extraction and tracking," *Computer Graphics Forum*, vol. 22, no. 4, 2003.

[18] J. Gray, L. Linsen, B. Hamann, and K. I. Joy, "Adaptive multi-valued volume data visualization using data-dependent error metrics," in *Proceedings of the Third IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2003)*, The International Association of Science and Technology for Development (IASTED), 2003.

[19] B. Cabral and L. Leedom, "Imaging vector fields using line integral convolution," in *Computer Graphics Proceedings*, pp. 263–269, 1993.

[20] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl, "Interactive exploration of volume line integral convolution based on 3d-texture mapping," in *Proceedings of the 1999 IEEE Conference on Visualization* (D. Ebert, M. Gross, and B. Hamann, eds.), pp. 233–240, IEEE, ACM Press, 1999.

[21] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf, "The state of the art in flow visualization: Dense and texture-based techniques," *Computer Graphics Forum*, vol. 23, 2004.

[22] D. Weiskopf and T. Ertl, "GPU-based 3d texture advection for the visualization of unsteady flow fields," in *Procceedings of WSCG 2004*, 2004.

[23] A. Telea and J. van Wijk, "3d IBFV: Hardware-accelerated 3d flow visualization," in *Proceedings of IEEE Conference on Visualization 2003*, pp. 233–240, 2003.

[24] R. A. Crawfis and N. Max, "Texture splats for 3d vector and scalar field visualization," in *Proceedings of IEEE Conference on Visualization 1993* (G. M. Nielson and D. Bergeron, eds.), pp. 261–266, IEEE, IEEE Computer Society Press, 1993.

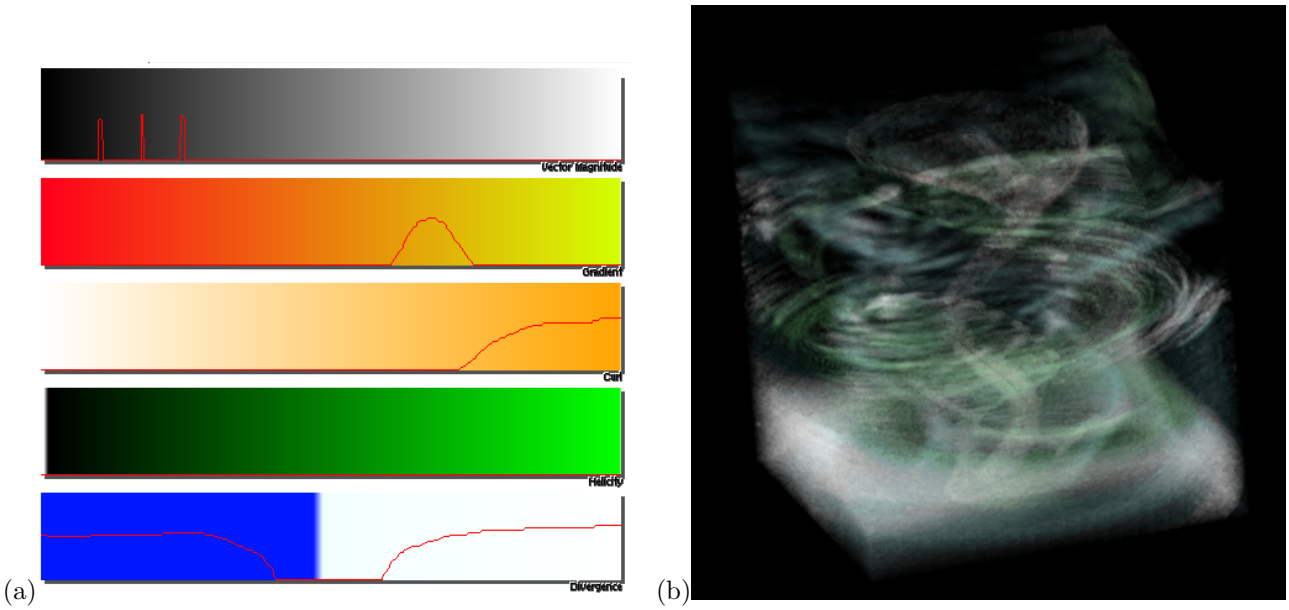

Figure 4. (a) User interface for an MDTF combining five components: velocity magnitude, determinant of velocity gradient tensor, curl magnitude, helicity, and divergence (top to bottom). (b) 3D IBFV method combined with MDTF, applied to tornado dataset.

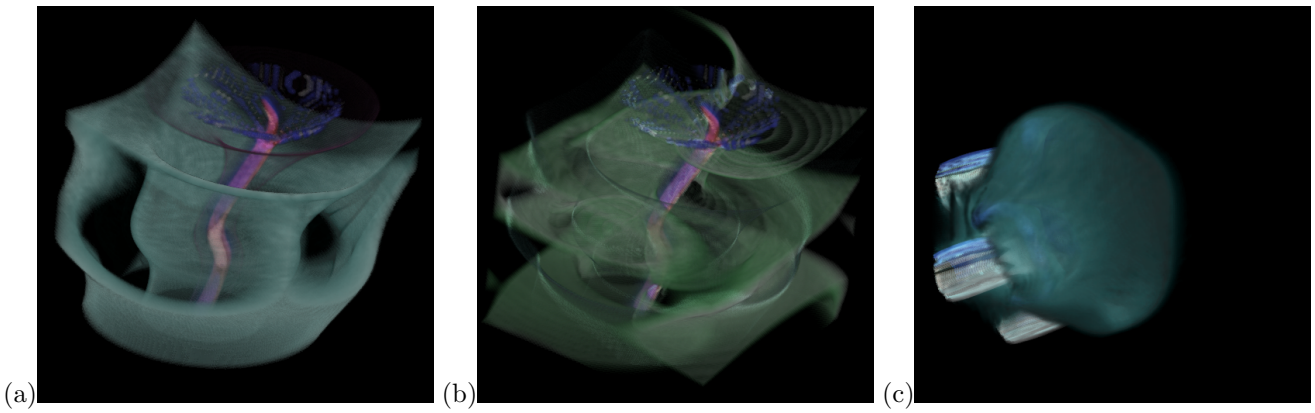

Figure 5. Texture advection method combined with MDTF: (a) highlighting velocity magnitude, divergence, and curl magnitude, applied to tornado dataset; (b) highlighting helicity, divergence, and curl magnitude, applied to tornado dataset; and (c) highlighting velocity magnitude, divergence, and curl magnitude, applied to timestep 1440 of five-jet dataset.

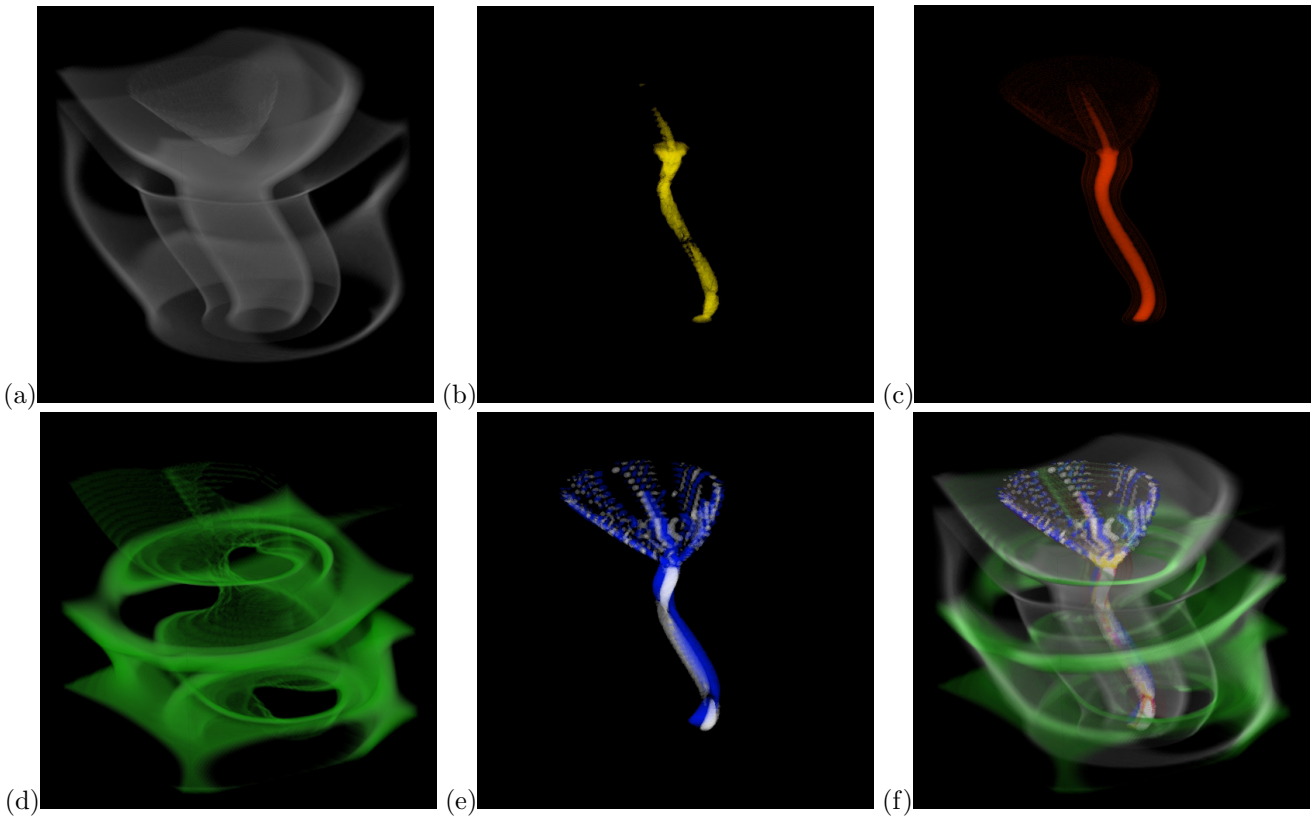

Figure 6. Feature extraction from tornado dataset selecting regions of (a) constant velocity, (b) high gradient tensor determinant, (c) high curl magnitude, (d) distinct helicity, (e) high negative (blue) and high positive (white) divergence, and (f) their combination using MDTF.

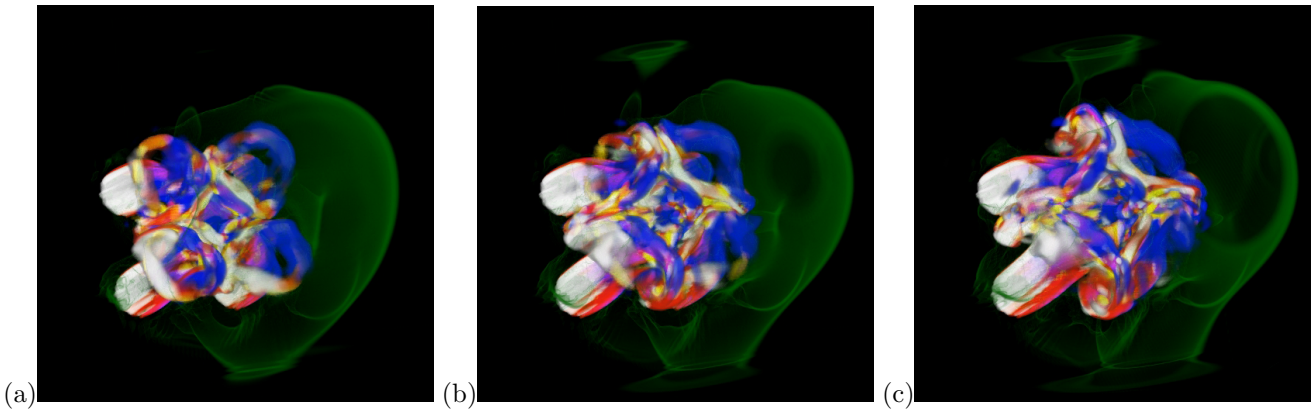

Figure 7. Rendering of five-jet dataset at timesteps (a) 1440, (b) 1760, and (c) 1960, with MDTF showing high curl magnitude (red), high determinant of velocity gradient tensor (yellow), high positive divergence (white) and high negative divergence (blue), and a feature of helicity (green).