Eurographics/ IEEE-VGTC Symposium on Visualization (2006)
Thomas Ertl, Ken Joy, and Beatriz Santos (Editors)

Structure-accentuating Dense Flow Visualization

Sung W. Park[1] Hongfeng Yu[1] Ingrid Hotz[1] Oliver Kreylos[1] Lars Linsen[2] Bernd Hamann[1]

[1] Institute for Data Analysis and Visualization (IDAV)
University of California, Davis
Davis, CA 95616, U.S.A.

[2] Department of Mathematics and Computer Science
Ernst-Moritz-Arndt-Universität Greifswald
Greifswald, Germany

Abstract

Vector field visualization approaches can broadly be categorized into approaches that directly visualize local or integrated flow and approaches that analyze the topological structure and visualize extracted features. Our goal was to come up with a method that falls into the first category, yet reveals structural information. We have developed a dense flow visualization method that shows the overall flow behavior while accentuating structural information without performing a topological analysis. Our method is based on a geometry-based flow integration step and a texture-based visual exploration step. The flow integration step generates a density field, which is written into a texture. The density field is generated by tracing particles under the influence of the underlying vector field. When using a quasi-random seeding strategy for initialization, the resulting density is high in attracting regions and low in repelling regions. Density is measured by the number of particles per region accumulated over time. We generate one density field using forward and one using backward propagation. The density fields are explored using texture-based rendering techniques. We generate the two output images separately and blend the results, which allows us to distinguish between inflow and outflow regions. We obtained dense flow visualizations that display the overall flow behavior, emphasize critical and separating regions, and indicate flow direction in the neighborhood of these regions. We have test our method for isolated first-order singularities and real data sets.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Vector Field Visualization

1. Introduction

Vector field visualization is a central topic in scientific visualization. There are two main streams to be reported. On one hand, there are flow visualization techniques, which display the local or integrated flow behavior for the entire domain. On the other hand, there are topological approaches, which examine the vector field to detect singularities or criticalities and retrieve topological informations like topological skeletons or vortices. Our approach belongs to the first category. However, we have developed a method that besides showing the overall flow behavior also accentuates the topological structure of the vector field.

The main idea behind our approach is based on the observation that streamlines cluster together in attracting regions like sinks or attracting separatrices, while they thin out in repelling regions like sources or repelling separatrices. Many approaches exist to overcome this problem. We, instead, would like to make use of this phenomenon by counting the number of streamlines that cross each cell of the underlying grid structure. This accumulation step leads to a scalar density field. The density is high for attracting regions and low for repelling regions when we use forward propagation, and vice versa when we use backward propagation. Rendering (and possibly blending) these two density fields leads to a visualization that highlights the structural information of the underlying flow field. Using an appropriate seeding mechanism for the streamlines, the visualization also exhibits the overall flow behavior leading to line integral convolution(LIC)-type visualizations in regions with no structural information. In Section 3, we describe in the idea underlying our approach and analyze its behavior for various topological cases.

The first step of our algorithm is the flow integration step. We use a geometric approach and higher-order integration methods to precisely trace the path of a (mass-less) particle moving under the influence of the underlying flow field. The integration method computes a streamline. Instead of storing the geometry of the streamline, we increase the counter of each cell the streamline passes through. Starting with all counters initially being zero and running an animation with many streamlines moving over time leads to an accumulated intensity value for each cell. We store the result in a texture representing the density field mentioned above. Details about the accumulation step are provided in Section 4.

The second step is the evaluation of the density field using texture-based rendering methods. Using texture-based methods for rendering supports a dense visualization, which is important, if we want to capture the entire vector field. We use transfer functions to extract attracting and repelling regions and explore their interplay by blending the two density fields for forward and backward propagation. We describe the rendering step in Section 5.

In Section 6, we provide results for two- and three-dimensional vector fields. We evaluate our approach and discuss its advantages and drawbacks. The results document that we capture both overall flow behavior and structural information.

2. Related Work

Vector field visualization approaches can be categorized into rendering approaches such as direct, geometric, and texture-based flow visualization, and analytical topology-extracting approaches, also called feature-based approaches [LHD*04].

Early attempts such as arrow and hedgehog plots or color coding fall into the category of direct flow visualization [PLV*02]. They provide an intuitive image of local flow properties. For a better understanding of global flow dynamics with respect to "long-term" behavior, integration-based approaches have been introduced. These integrate flow data leading to trajectories of no-mass particles moving over time. Geometric flow visualization approaches render the integrated flow using geometric objects such as lines, tubes, ribbons, or surfaces [PLV*02].

In particular, streamlines are widely used and have been integrated into various flow visualization systems [BMP*90, SML04]. Streamlines naturally lead to a sparse representation of the vector field, such that seeding strategies become a critical issue [Lar02,TB96]. Dense representations are desirable, as they provide information concerning overall flow behavior and serve as a context for chosen visualization methods. Park et al. [PBL*05] presented a dense geometric flow visualization approach by using a high number of randomly-seeded streamlines with short life times, generated via particle advection in texture space.

In texture-based flow visualization, a texture is used for a dense representation of a flow field. The texture is filtered according to the local flow vectors leading to a notion of overall flow direction [LHD*04]. The most prominent approaches are LIC [CL93] and texture advection [MBC93]. The LIC primitive is a noise texture, which is convolved in the direction of the flow using filter kernels. In texture advection, the primitive is a "moving" texel, while the motion is directed by the flow field [JL97]. One major drawback of texture-based methods for volume data is occlusion, which can be alleviated by the application of multi-dimensional transfer functions (MDTF) [PBL*04].

Our approach is a hybrid one, since we are using both geometric and texture-based methods. The geometric step is used for a precise flow integration, while the texture-based step is used for a dense visualization. Recently, Weiskopf et al. [WSEE05] also presented a hybrid approach, where they used forward particle flow integration for temporal coherence in time-dependent vector fields. Schussman and Ma [SM04] introduced a rendering method for dense thin lines, which is related to our work, as it samples line structures in a 3D texture space, which is used for rendering.

Feature-based flow visualization is concerned with the extraction of specific patterns of interest, or features. Various features such as vortices, shock waves, or separatrices have been considered. Each of them has specific physical properties, which can be used to extract the desired feature. Once a feature has been extracted, standard visualization techniques are used for rendering [PVH*03]. Topological analysis of two-dimensional vector fields was introduced to the visualization community by Helman and Hesselink [HH91], based on detection and classification of critical points. Scheuermann et al. [SHK*97] generalized the approach to higher-order critical points. Wischgoll and Scheuermann [WS02] generated closed separating lines or separatrices in 2D vector fields. Theisel et al. [TWHS03] connected saddles to visualize the topological skeleton of 3D vector fields, which was extended to higher-order critical points by Weinkauf et al. [WTS*05]. We do not compute the precise locations of critical points or separating structures, but provide a flow visualization that supports the highlighting of such topological structures.

3. Structure-accentuating Dense Flow Visualization

3.1. Overview

Our method is based on a very simple principle which automatically generates an image showing the basic field structure and provides an overview of the overall behavior of the flow at the same time. After seeding the entire domain with particles we trace these particles moving with the flow. Instead of displaying the particle traces themselves, we use them to generate a density field, which serves as basis for rendering. The density is generated by counting how often a cell is hit in a discretized domain. A forward and backward advection of the particles leads to an accumulation of particles in attracting and repelling points or lines, respectively, resulting in a high density. Such regions are, for example, sources, sinks, and attracting/repelling separatrices. This ap-

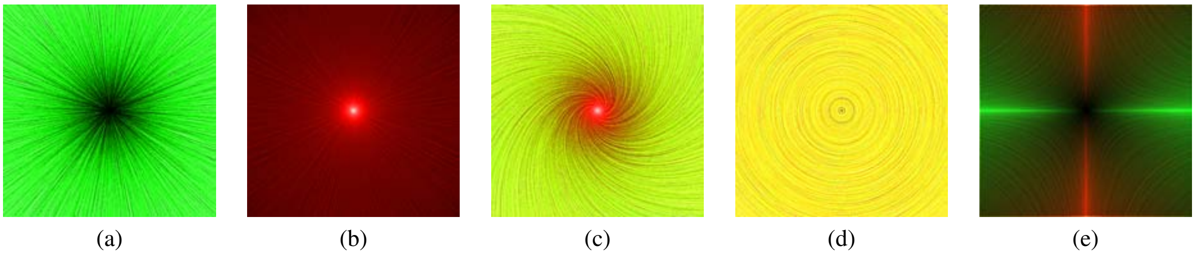


Figure 1: *Density change in neighborhood of isolated singularities of first order: (a) Source with forward propagation; (b) source with backward propagation; (c) spiraling source; (d) center; (e) saddle with blended forward and backward propagation.*

proach results in a dense representation of overall flow behavior with an accentuation of the "field skeleton“, its separatrix structure. Choosing different transfer functions allows us to focus on either the basic field structure or a dense texture showing particle paths. Our system supports interactive exploration of the data. Attractive and repelling regions can be distinguished by looking at forward and backward advection separately and blending these images together using, for example, different colors. The blending of inflow/outflow also support an improved understanding of flow direction.

Vector Field Magnitude. Many simulations, especially of three-dimensional fluid flows, are built on the assumption of dealing with an incompressible fluid, with a constant physical density without sources and sinks. If we use the vector field with its original magnitude for the advection, the density of accumulated material points in our approach should approach the same limit as the physical density as the number of seed particles goes to infinity and the fluid volume goes to zero. In practice, i.e., in finite realizations, both densities are similar but in general different. But rather than visualizing the physical density, we want to enhance field structure, which is independent from the vector field magnitude. Therefore, we use the normalized vector field, leading to a stronger pronounced structure.

3.2. Structure Accentuation

Mass balance states that the density of a material point is changed by the divergence of velocity. Points where density has an extremum are governed by mass balance with zero convection. For particle advection this fact leads to an accumulation of particles in sources and sinks, and the accumulation is also observable for saddle points. The only type of critical point with no change of density in its neighborhood is the center point, i.e. a critical point where the Jacobian has only imaginary eigenvalues representing pure rotational behavior. The field structure is basically determined by its critical points and the flow behavior in their neighborhood. We provide an overview of density changes in the neighborhood of isolated field singularities of first order.

2D Structure. The density is determined by the number of particles accumulating in a bucket. This number is proportional to the area of the bucket, the area covered by all entering streamlines of a certain length, which depends on the number of advection iterations. Given a simple sink, all particles or streamlines move radially towards it, resulting in a rotational-symmetric vector field. For initialization we assume a continuous seeding on an infinite domain. When performing a forward propagation all particles move toward the critical point. The density increases the closer we get to the critical point. A simple computation of the bucket area results in a density decreasing with $1/r$, where r is the distance from the critical point. The density field for the backward advection does not allude to the position of the sink. The reversed result is obtained by considering a source. Figure 1(a) shows forward propagation for a source. Only in case of a backward propagation the generated image clearly indicates where the critical point is located, see Figure 1(b). In both cases, the gradient vector field of the density is a source without rotational component. For a sink we observe the exact opposite, i. e., for forward propagation we would obtain an image like the one shown in Figure 1(b) and for backward propagation an image like the one in Figure 1(a).

The discretization of the domain and sparse seeding breaks the rotational symmetry of the density, and we obtain in addition to the enhancement of the critical points an image similar to a LIC image, resulting in different images for focus sources and spiral sources. Figure 1(c) visualizes the density field for a spiraling source when blending forward and backward propagation results. Similar considerations can be done for attracting and repelling separatrices, where forward advection generates a high density along attracting lines, and backward advection high density along repelling lines, see Figure 1(e).

Some examples illustrating the emerging structure in the neighborhood of critical points are shown in Figure 2 togehter with their topology. Figure 2(a) is a combination of four saddles symmetrically arranged around a center point. In case of backward propagation, the generated image clearly shows the attracting separatrices. The forward propagation accentuates the attracting separatrices. The saddle point itself is located where the attracting and repelling separatrices meet. In the neighborhood of the center point the divergence of the field is zero, leading to an almost constant density, see Figure 1(e). Figure 2(b) shows a spiral source in the center and four saddle points. Similar to the previous example the saddles and the separatrices swirling into the spiral source are expressed well. The third example, shown in Figure 2(c), is a combination of a source, a sink, and two saddles. Backward propagation clearly indicates where the source is located, while forward propagation generates the complementary part of the topology, enhancing the sink and the attracting separatrices.

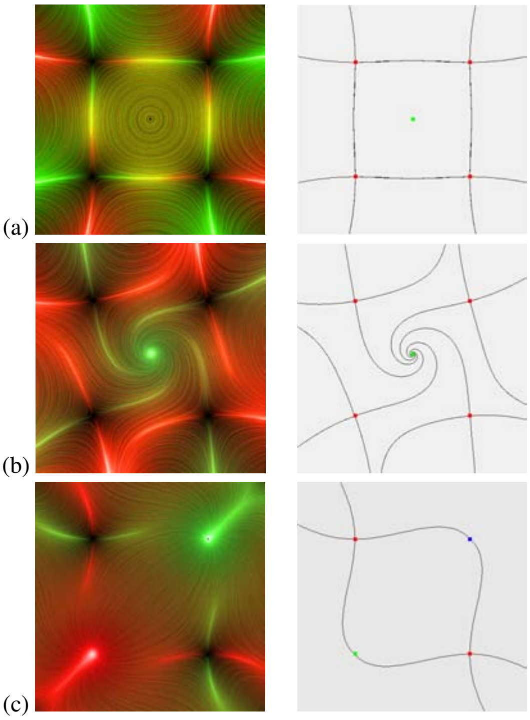


Figure 2: *Density change in neighborhood of simple combinations of singularities of first order: (a) Four saddles arranged around a center point; (b) four saddles arranged around a spiral source; (b) two saddles, a source, and a sink.*

In all examples the topological structure of the fields is pronounced, matching the results from a topological analysis. Using different colors for the forward and backward integration allows us to distinguish the attracting from the repelling features.

3.3. Overall Flow Behavior

The density representation is appropriate to show basic field structure except rotations. It does not extract single particle traces and does not visualize any rotations. To illustrate the context in which the field structure is embedded it is important to make also particle traces visible.

Small irregularities caused, by the discretization or noise for example, lead to a local accumulation of particles and a slightly higher density. These density fluctuations are propagated by the flow leaving traces of the particle movement. Using a non-uniform sparser seeding, e. g., using a quasi-random seeding, we can enhance this effect. These density fluctuations do not lead to density extremas, as for the structure skeleton. Thus, they can easily be blended in and out in the rendered image by changing the transfer function.

This method is similar to image-based rendering methods, with the difference that the texture is not automatically spread uniformly over the entire domain; it depends on the chosen transfer function. This is especially useful for three-dimensional visualizations, allowing one to focus on the basic structure without being occluded by a dense texture. No intelligent dye injection or explicit computation of the critical points is necessary.

4. Geometry-based Accumulation

4.1. Flow Integration

Let $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ or $\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be the two- or three-dimensional vector field, respectively. We integrate the flow by tracing particles over time. The path of the particle under the influence of vector field $\mathbf{f}$ is defined as

$$\mathbf{p}(t) = \mathbf{p}(0) + \int_0^t \mathbf{f}\big(\mathbf{p}(x)\big)dx \, ,$$

where $\mathbf{p}(t)$ is the position of a particle at time t. The position $\mathbf{p}(0)$ at time $t = 0$ denotes the seed location for that particle. Since precise flow integration is important for the quality of our method, we solve the integral equation using a fourth-order Runge-Kutta method keeping the integration error low.

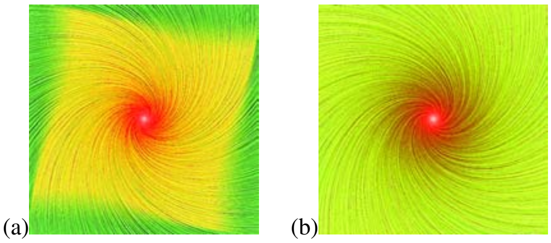


Figure 3: *(a) Density field influenced by domain shape; (b) domain shape influence treated by seeding strategy.*

4.2. Seeding

A critical issue for the success of our method is the seeding strategy. In order for all regions to be treated, the seeding strategy must cover the entire domain. The seeding strategy also needs to address the domain shape of the vector field biasing the density field.

Different seeding strategies can be adopted to achieve a more or less dense representation of the flow field in different regions. However, since typically no a priori knowledge about the underlying vector field is given, the seed locations should be distributed in a more or less uniform fashion. A simple approach to achieve a dense representation of the field is to cover the entire domain by placing a particle in each grid cell. However, such a repetitive seeding strategy may lead to aliasing artifacts. To avoid potential aliasing artifacts, each particle can be jittered slightly.

A more elegant way of seeding points to lead to a uniform coverage over the domain is to use quasi-random numbers. One reason to prefer quasi-random numbers over pseudorandom generators is that they tend to provide a more well-distributed set of samples [Nie92]. We decided to use a *Halton* sequence for quasi-random seed generation as adopted by Park et al. [PBL*05]. As discussed above, it may sometimes be beneficial to use less dense seeding strategies in order to bring out the characteristics of overall flow behavior. When using quasi-random numbers, a less dense representation can easily be generated by simply computing less seed points.

A domain shape-biasing density field results when seeds are just placed within the vector field domain once. Figure 3(a) shows an example of a density field for a spiraling source being influenced by the square-shaped domain. One way to address this problem is to continuously inject particles along the domain boundaries to simulate infinite particles flowing into the domain. Different approaches can be adopted to simulate continuous particle flow. One simple approach that is often adopted in many computational simulations is to seed particles over a much larger domain that covers the entire vector field and mirror the vector field in the undefined regions or propagate the boundary values to the undefined regions for advection. Another approach is to compute the inflow and outflow of the vectors along the domain border and inject particles into the domain where there is inflow. Figure 3(b) shows domain influence addressed by seeding particles over a larger domain and propagating the boundary values.

4.3. Accumulated Texture Generation

We initialize flow integration with a dense quasi-random seeding strategy to simultaneously start tracing particles. We capture the traces by evaluating the particles' positions after each flow integration step. Thus, for each point in time t we store where all the particles are located, position $\mathbf{p}(t)$.

We use a grid structure to partition the domain of the vector field $\mathbf{f}$ into cells. The position $\mathbf{p}(t)$ of the particles at each point in time t is bucketed into the respective cell. The most intuitive grid structure to use is a uniform rectilinear grid. It is not only easy to bucket positions into a uniform rectilinear grid, but it also maps naturally to 2D and 3D textures (depending on the dimension of the flow field's domain). Thus, it allows for efficient rendering in commodity graphics hardware using texture-based visualization techniques.

Initially, we set all grid cell counter values to zero. After each integration step, all particles get bucketed into the respective cells. The counter of each cell is incremented by the number of particles being in the cell at that respective moment. During flow integration the particles will move away from repelling regions and toward attracting regions, leading to an increasing density toward attracting regions. Thus, after a sufficiently large number of time steps, the grid represents a density field. The density field is mapped to a 2D or 3D texture, respectively.

The resolution of the chosen grid determines the accuracy of the density field. Thus, resolution governs the quality of our entire visualization method.

4.4. Implementation

For the flow integration step, we take advantage of commodity programmable graphics processing units (GPUs) to accelerate our implementation. Although Halton sequence can be implemented to run in the GPU, we compute them in the CPU since we only need to compute and send them to the GPU once.

After seeding, both the advection and bucketing process can be done entirely in the GPU. The generated seeds can be used as an input to the advection process, where the fourth-order Runge-Kutta integration can be performed in the fragment program. Once the newly advected particles are output, the new particle positions are used as input vertices to output to its proper location for bucketing. For separating forward and backward integration, we store one seed in the RG (Red-Green) components of texture and another seed in the BA (Blue-Alpha) components. The two seeds are advected separately and bucketed separately into different components of the output color.

The 2D approach maps nicely to GPU architecture. Unfortunately, the 3D process does not map as nicely. Although advection can still be done in the GPU, bucketing becomes more difficult as GPUs do not support outputting to arbitrary 3D volume location (not yet). Other ways to accelerate bucketing include laying out the 3D volume into 2D texture and using an address scheme to access different slice of the volume. However, this approach becomes difficult when volume dimensions become large. Also, there are other constraints such as memory limits.

To use texture-based rendering, we output to a regular grid. The input grid used for flow integration, however, can be of any structure.

5. Texture-based Exploration

To explore the two density fields generated by forward and backward propagation, we can investigate the textures, in which they are stored, separately. In the 2D case, the density fields can simply be rendered to an image by mapping the texture to a polygon and visualizing the polygon. In the 3D case, standard scalar field visualization tools can be applied. Since the density field is already stored as a 3D texture, we apply standard 3D texture-based direct volume rendering techniques. We use transfer functions that map density values to different color and opacity values to explore the desired features.

Blending Forward and Backward Accumulation. To eliminate the need to store and explore two density fields independently, we can blend the two density fields in a weighted fashion. Typically, a simple averaging step would be used. The blended density field can be explored using transfer functions for direct volume rendering. Based on our earlier analysis, we expect to see interesting features (regions of critical behavior) from the volume in regions of high density after particles have been traced both in the forward and backward direction.

Unfortunately, the blended density field provides just one scalar value at each grid point, such that no direction of flow is communicated. For example, we cannot distinguish between sources and sinks, and it would be hard to determine where flow is coming in or going out to form a saddle structure. One way to address this issue is to leave the individual density fields for forward and backward integration separate and blend them by applying multi-dimensional transfer function as described by Park et al. [PBL*04]. To each density field we apply a 1D transfer function to extract the desired features. The resulting images are blended. In this way, flow direction can be highlighted by applying one color spectrum to high-density regions in the forward-integrated density field and another color spectrum to high-density regions in the backward-integrated density field. The blended image distinguishes incoming and outgoing flow at critical regions and, thus, depicts the direction of flow. Transfer functions also can be applied to the gradient magnitude of the density field to accentuate flow structure.

6. Results and Discussion

We first tested our approach for simple, prototypical synthetic datasets, representing the basic isolated field singularities. The results show clearly the high density values in the neighborhood of critical points and accentuation of the surrounding structures, see Figures 1 and 2. Rotational patterns are recovered by advecting small fluctuations in the density function and can be seen in the texture as LIC-like structures.

To evaluate the method on real-world datasets, we used one 2D and two different 3D datasets. We applied it to 2D slices as well as to the entire 3D domain.

We used is the wind field simulation over New Orleans from the IEEE Visualization 2005 contest. The dataset consists of a model of New Orleans and airflow over this model. Figure 4 shows a slice. Here, orange is used for forward and white for backward advection. The visualization reveals several critical points as saddles and sources.

The influence of the special choice of the transfer function on the final image can be seen in Figure 5. The figure shows another slice of the New Orleans dataset of a region near the Superdome. Displaying only very high densities leads to structure extraction, as seen in Figure 5(a) and (c). Using a transfer function based on gradient magnitude also highlights the flow structure by emphasizing the region of high gradient change, as seen in Figure 5(b). Changing the transfer function allows adding context to this structure, providing an overall impression of the flow.

Figure 6 shows the results of a two-dimensional simulation of a swirling jet. Transfer functions are applied on the resulting density field in (a) and on the gradient magnitude of the density field in (b). Figure 6(c) shows the explicitly extracted vector field topology in comparison. It can be seen that much of the topological structure is recovered, but due to the resolution not all details are shown. Focusing on a special region rendering it in a higher resolution can resolve the structure in more detail.

Figure 7 shows the results for two 3D datasets. Figure 7(a) shows a flow visualization of the tornado dataset [CM93] of size 128^3. The figure shows a visualization of the core of the 3D tornado dataset extracted using a proper transfer function. Figures 7(b) and (c) show the New Orleans dataset. In these images, the particles are propagated forward and backward at the same time. Thus, only one density function is generated. Figure 7(b) shows the surface representing a very high density, while Figure 7(c) uses a transfer function

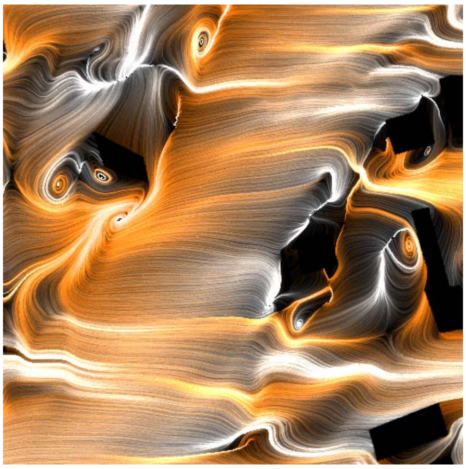


Figure 4: *Structure visualization of 2D slice of New Orleans airflow dataset. Critical points like saddle, sources and sinks and their connections with context are emphasized by the visualization.*

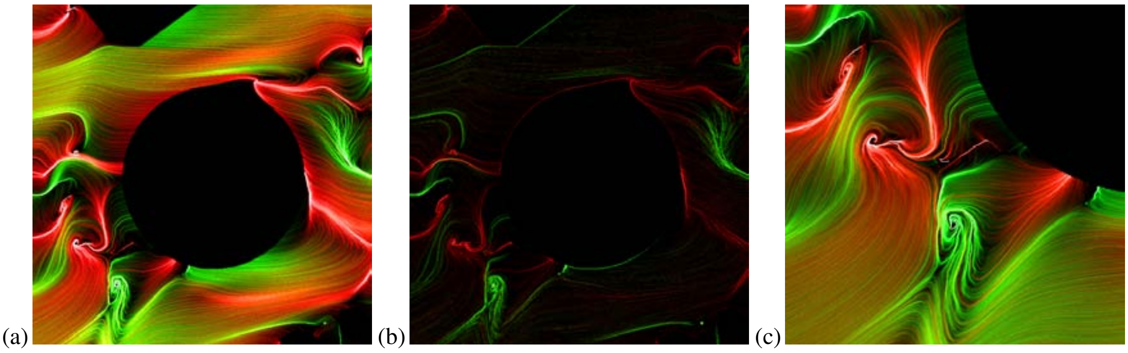


Figure 5: *2D slice of New Orleans airflow dataset showing the same density function rendered with two different transfer functions: (a) Overall flow behavior; (b) transfer function applied to gradient magnitude of density function to focus on the structure; (c) zoomed view on the density function.*

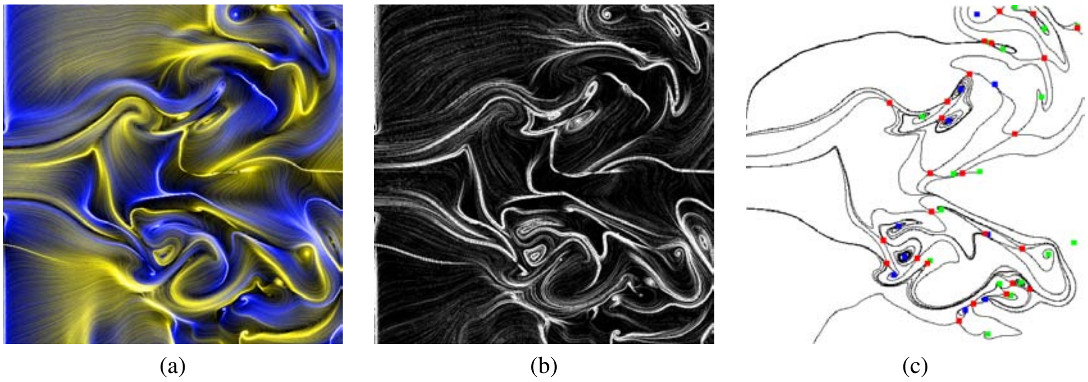


Figure 6: *2D simulation of a swirling jet, using two different transfer functions, (a) Overall flow behavior; (b) transfer function applied to gradient magnitude of density function to focus on the structure; (c) explicit flow field topology. (The explicit topology was generated with the software FAnToM, developed at the University of Kaiserslautern, Germany.)*

emphasizing several densities using different colors. Even though one obtains a rough impression of the field, one cannot see the nice structures as it is possible for two dimensions.

The 2D datasets can be rendered at interactive frame rates during density accumulation. For 3D data sets, this performance cannot be achieved due to the problems described in the implementation section. All 2D images were generated at 512×512 resolution, and 3D volume images were rendered at 256^3 resolution.

7. Conclusions and Future Work

We have presented a flow visualization method based on a hybrid approach providing structural insight into flow field data. Interactive exploration of the flow makes it possible to focus on flow structure, a dense overall representation, or a combination of both. The method is easy to understand and implement. Even so the concept of accumulating particles is not able to extract rotational structures, they are made visible by the propagation of dense regions. Compared to other dense flow visualization methods, e. g., LIC, the structure of the vector field is much easier to recognize without distracting a viewer by non-relevant parts of the field's domain.

We have explored 2D structures in depth and have initial results extending our approach to 3D datasets. We plan to explore 3D structures in depth as well as develop more flexible transfer functions to make the field structure even more prominent.

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. We gratefully acknowledge the support of the W.M. Keck Foundation provided to the UC Davis Center for Active Visualization in the Earth Sciences (CAVES). We thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis. In particular, we thank Wolfgang Kollmann for valuable discussions.

References

[BMP*90] BANCROFT G. V., MERRITT F. J., PLESSEL T. C., KELAITA P. G., MCCABE R. K., GLOBUS A.: FAST: a multi-processed environment for visualization of computational fluid dynamics. In *VIS '90: Proceedings of IEEE conference on Visualization '90* (1990), IEEE Computer Society Press, pp. 14–27. 2

[CL93] CABRAL B., LEEDOM L.: Imaging vector fields using line integral convolution. In *Computer Graphics Proceedings* (1993), pp. 263–269. 2

[CM93] CRAWFIS R. A., MAX N.: Texture splats for 3d vector and scalar field visualization. In *Proceedings of IEEE Conference on Visualization 1993* (1993), Nielson G. M., Bergeron D., (Eds.), IEEE, IEEE Computer Society Press, pp. 261–266. 6

[HH91] HELMAN J., HESSELINK L.: Visualizing vector field topology in fluid flows. *IEEE CGA 11*, 3 (1991), 36–36. 2

[JL97] JOBARD B., LEFER W.: The motion map: efficient computation of steady flow animations. In *VIS '97: Proceedings of IEEE conference on Visualization '97* (1997), IEEE Computer Society Press, pp. 323–328. 2

[Lar02] LARAMEE R. S.: Interactive 3d flow visualization using a streamrunner. In *CHI 2002 Conference on Human Factors in Computing Systems, Extended Abstracts* (2002), pp. 804–805. 2

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum 23* (2004). 2

[MBC93] MAX N., BECKER B., CRAWFIS R.: Flow volumes for interactive vector field visualization. In *VIS '93: Proceedings of IEEE conference on Visualization '93* (1993), pp. 19–24. 2

[Nie92] NIEDERREITER H.: *Random Number Generation and quasi-Monte Carlo Methods*. SIAM, 1992. 5

[PBL*04] PARK S., BUDGE B., LINSEN L., HAMANN B., JOY K. I.: Multi-dimensional transfer functions for interactive 3d flow visualization. In *Proceedings of The 12th Pacific Conference on Computer Graphics and Applications - Pacific Graphics 2004* (2004), Cohen-Or D., Ko H.-S., Terzopoulos D., Warren J., (Eds.). 2, 6

[PBL*05] PARK S. W., BUDGE B., LINSEN L., HAMANN B., JOY K. I.: Dense geometric flow visualization. In *Eurographics / IEEE VGTC Symposium on Visualization - EuroVis 2005* (2005), Brodlie K., Duke D., Joy K. I., (Eds.), pp. 21–28, 318. 2, 5

[PLV*02] POST F. H., LARAMEE R. S., VROLIJK B., HAUSER H., DOLEISCH H.: Feature extraction and visualisation of flow fields. In *Eurographics 2002, State of the Art Reports* (2002), Fellner D., Scopigno R., (Eds.), The Eurographics Association, IEEE Computer, pp. 69–100. 2

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum 22*, 4 (2003), 775–792. 2

[SHK*97] SCHEUERMANN G., HAGEN H., KRÜGER H., MENZEL M., ROCKWOOD A.: Visualization of higher order singularities in vector fields. In *VIS '97: Proceedings of the 8th conference on Visualization '97* (1997), IEEE Computer Society Press, pp. 67–74. 2

[SM04] SCHUSSMAN G., MA K.-L.: Anisotropic volume rendering for extremely dense, thin line data. In *VIS '04: Proceedings of the conference on Visualization '04* (2004), IEEE Computer Society, pp. 107–114. 2

[SML04] SCHROEDER W., MARTIN K., LORENSEN B.: *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*, 3 ed. Kitware, Inc. publishers, 2004. 2

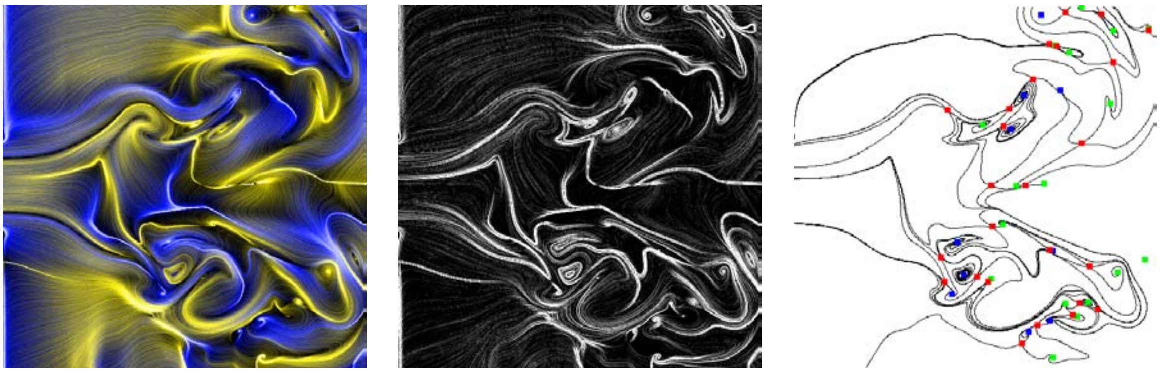
[TB96] TURK G., BANKS D.: Image-guided streamline placement. In *International Conference on Computer Graphics and Interactive Techniques* (1996), pp. 453–460. 2

[TWHS03] THEISEL H., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Saddle connectors - an approach to visualizing the topological skeleton of complex 3d vector fields. In *VIS '03: Proceedings of the 14th IEEE Visualization '03* (2003), IEEE Computer Society, pp. 30–37. 2

[WS02] WISCHGOLL T., SCHEUERMANN G.: Locating closed streamlines in 3d vector fields. In *Proceedings of the Symposium on Data Visualisation 2002* (2002), Eurographics Association, pp. 227–235. 2

[WSEE05] WEISKOPF D., SCHRAMM F., ERLEBACHER G., ERTL T.: Particle and texture based spatiotemporal visualization of time-dependent vector fields. In *VIS '05: Proceedings of the 8th conference on Visualization '05* (2005), IEEE Computer Society Press, pp. 647–654. 2

[WTS*05] WEINKAUF T., THEISEL H., SHI K., HEGE H.-C., SEIDEL H.-P.: Extracting higher order critical points and topological simplification of 3d vector fields. In *VIS '05: Proceedings of the 8th conference on Visualization '05* (2005), IEEE Computer Society Press, pp. 559–566. 2



(a) (b) (c)

Figure 6: *2D simulation of a swirling jet, using two different transfer functions, (a) Overall flow behavior; (b) transfer function applied to gradient magnitude of density function to focus on the structure; (c) explicit flow field topology. (The explicit topology was generated with the software FAnToM, developed at the University of Kaiserslautern, Germany.)*

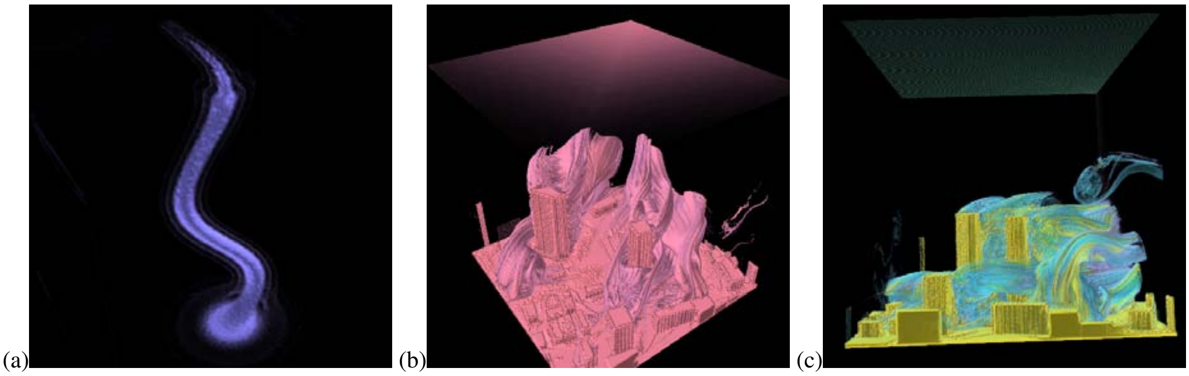


Figure 7: *(a) Visualizing core of 3D tornado dataset; (b),(c) full 3D New Orleans airflow dataset, using two different transfer functions for volume rendering.*