Structure-accentuating Dense Flow Visualization

Sung W. Park\textsuperscript{1} Hongfeng Yu\textsuperscript{1} Ingrid Hotz\textsuperscript{1} Lars Linsen\textsuperscript{2} Bernd Hamann\textsuperscript{1}

\textsuperscript{1} Institute for Data Analysis and Visualization (IDAV) University of California, Davis Davis, CA 95616, U.S.A.
\textsuperscript{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 brings out 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 analyzed the results of our method for isolated first-order singularities and real data sets.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism

1. Introduction

Vector field visualization is a classical 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 LIC-type visualizations in regions with no structural information. In Section 3, we describe in detail the idea behind 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 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 given in Section 4.

The second step is the evaluation of the density field using texture-based rendering methods. Using texture-based methods for rendering allows for 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 are able to 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, SML’04]. Streamlines naturally lead to a sparse representation of the vector field, such that seeding strategies become a critical issue [Lar’02, 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 line integral convolution (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 has been introduced by Heman and Hesselink [HH91], which is 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. Theisel et al. [TWS03] 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 providing 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 the rendering. The density is generated by counting the hits of cells of 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,
reported by to a high density. Such regions are, for example, sources, sinks, and attracting/repelling separatrices. This approach results in a dense representation of the overall flow with an accentuation of the field skeleton. 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 provides an 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. The density of accumulated material points used 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, what means finite realizations, both densities are in general different. Thus, the particle density displayed in our approach will show some structure, but normalizing the vector field leads to an even stronger pronounced structure.

For two-dimensional slices of three-dimensional flow fields, sources and sinks are smoothly distributed since they are recognized as the negative partial derivative in the direction normal to the two-dimensional plane of the velocity component in this direction. These singularities show clearly up as high densities in our particle accumulation.

### 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 are center points, critical points where the Jacobian has only imaginary eigenvalues representing a pure rotational behavior.

The field structure is basically determined by its criticalities and the flow behavior in their neighborhood. In the following, we give an overview of density changes in the neighborhood of isolated field singularities of first order.

**2D Structure.** Given a source, all particles or streamlines move away from the source. The vector field is symmetric around the source and depends only on the radius \( r \). For initialization, we use a continuous seeding with density \( d \) on a circular domain with radius \( R \) around the source. Then the density distributions for forward advection \( D_F(r) \) and backward advection \( D_B(r) \) for \( 0 < r < R \) converge to

\[
D_F(r) = d \frac{r^2}{2} \quad \text{and} \quad D_B(r) = d \frac{R^2 - r^2}{2}.
\]

The density field for the forward advection does not allude to the position of the critical point. However, when performing a backward propagation all particles move toward the critical point. The density increases the closer we get to the critical point. Combining both directions the density distribution \( D(r) \) is dominated by the effects of the backward advection

\[
D(r) = d \left( \frac{R^2}{2} \right).
\]

These observations are also true for spiral sinks, where the field is rotationally symmetric around the source, i.e., the two eigenvalues of the Jacobian are equal. The resulting density reflects this symmetry. Its gradient vector field is a focus sink without rotational component. The only difference between spiraling and not spiraling sources or sinks may be the gradient magnitude.

A discretized domain breaks the rotational symmetry of the density and we obtain an image similar to a LIC image when performing a forward propagation. The image is shown in Figure 1(a). In case of a backward propagation the generated image clearly indicates, where the critical point is located, see Figure 1(b). For a sink we observe the exact opposite, i.e., for forward propagation we would obtain an image like the one in Figure 1(b) and for backward propagation we would obtain an image like the one in Figure 1(a).

The same effect is visible for the spiraling source. Figure 1(c) visualizes the density field for a spiraling source when blending forward and backward propagation results. The brighter lines we observe going into the source in Figures 1(a) and (c) are due to the fact that we are seeding our particles on a rectilinear grid while the influence of the source is radial.

As soon as there is an asymmetry in the neighborhood of the singularity, i.e., if the two eigenvalues of the Jacobian are different in the two eigendirections, it shows up in the density visualization.

For an isolated center, the divergence of the field is zero everywhere. Thus, the density does not change in the neighborhood of the center. Figure 1(d) shows the constant density field. Even the discretization does not influence the result very much, and almost no structural information shows up.

Similar considerations can be made for other criticalities. In a saddle, attracting and repelling separatrices meet. The forward propagation leads to high density values in the neighborhood of the attracting separatrices, while the backward propagation leads to high density values in the neighborhood of the repelling separatrices. When blending the two results the separatrices meet in the critical point, as shown in Figure 1(e).
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.

3D Structure. The density distribution in the neighborhood of three-dimensional critical points is similar to the two-dimensional case. The particle advection enhances attracting and repelling lines and surfaces, sources and sinks. Since rotations do not change the density they do not show up in the density distribution. Vortex cores are visible if attracting or repelling.

3.3. Overall Flow Behavior

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

Small irregularities caused, for example, by the discretization or noise 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 extremes, as for the structure skeleton. Thus, they can easily be blended in and out in the rendered image by changing the transfer function.

This part is similar to image-based rendering methods, with the difference that the texture is not automatically spread uniformly over the entire domain, but it depends on the chosen transfer function. This is especially useful for three-dimensional visualizations, allowing 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 \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \) or \( f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \) 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 \( f \) is defined as

\[
p(t) = p(0) + \int_0^t f(p(x))dx,
\]

where \( p(t) \) is the position of a particle at time \( t \). The position \( p(0) \) 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.

Another critical issue for the success of our method is the seeding strategy. In order for all the regions to receive attention, the seeding strategy needs to cover the entire domain of the flow field. Different seeding strategies can be adopted to achieve 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 equal spacing.

One 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 get 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]. In this paper, we decided to use a Halton sequence for quasi-random seed generation as adopted by Park et al. [PBL05].

As discussed above, it may sometimes be beneficial to use less dense seeding strategies in order to bring out the characteristics of the overall flow behavior. A less dense representation can easily be generated by skipping a certain number of grid cells during uniform seeding, or, when using quasi-random numbers, by simply computing less seed points.

4.2. Accumulated Texture Generation

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

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 or 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 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.

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

4.3. Implementation

For the flow integration step, we take advantage of commodity programmable graphics processing units (GPUs) to accelerate our implementation. In the case of uniform seeding, we can operate entirely in the GPUs to accelerate the process. Uniform seeding over the entire domain can be done by simply rendering a quad over the output domain and outputting spatial locations of each fragment instead of color values. For quasi-random seeding, the values can be processed in GPUs, but we compute them in the CPU and store them in a texture, which needs to be send to the GPU only 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 Runge-Kutta 4th-order integration can be computed 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. Then, the two seeds are advected separately and bucketed separately into different components of the output color.

The two-dimensional approach maps nicely to GPU architecture. Unfortunately, the 3D process does not map as nicely. Although advection can be still done in the GPU, bucketing becomes tricky as GPUs do not support outputting to arbitrary 3D volume location (at least not yet). Other ways to accelerate bucketing include laying out the 3D volume into 2D texture and using an addressing scheme to access different slice of the volume. However, this approach becomes tricky 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 use 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 two-dimensional case, the density fields can simply be rendered to an image by mapping the texture to a polygon and visualizing the polygon. In the three-dimensional case, standard scalar field exploration 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 apply transfer functions to 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 criticalities) 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 covered. For example, we cannot distinguish between sources and sinks, and it will 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 one-dimensional transfer function to extract the desired features. The resulting 3D 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, draws out the direction of the flow.
Figure 2: Two-dimensional slice of the tornado dataset reflecting the rotational structure of the flow field for two different transfer functions. (a) Blue and red (b) green and blue indicate forward and backward propagation, respectively.

6. Results and Discussion

We first tested our approach on some 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 Figure 1. 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 two-dimensional and two different three-dimensional datasets. We applied it to two-dimensional slices as well as to the entire three-dimensional domain.

Figure 2 shows a flow visualization of a two-dimensional slice of the tornado dataset [CM93] of size $128^3$. Even so this is a purely rotational dataset the structure is expressed very well. The formation of the four “arms” is an effect of the discretization of the data, representing selected particle traces. The density resulting from the forward advection and backward advection are shown using different colors. Depending on the transfer function more or less individual particle traces are visible.

The second data set is the wind-field simulation over New Orleans from the IEEE Visualization 2005 contest. The data consists of a model of New Orleans and airflow over this model. Figure 3 shows a two-dimensional slice of the region near the Superdome. Here, green is use for forward and white for backward advection. The visualization shows several critical points as saddles (e.g. in the upper right corner) and sources. Figure 4 shows another slice of the same dataset in a different region. Figure 4(a) shows density only based on forward advection, Figure 4(b) density using only backward advection. Figure 4(c) shows how both structure complement each other, blending into the complete structure of the flow. In Figure 4(d), we compare our method to a LIC image.

Figure 3: Structure visualization of a 2D slice of the New Orleans airflow dataset in the area surrounding the Superdome. Critical points like saddle, sources and sinks and their connections with context are emphasized by the visualization.

Figure 8: Visualizing the core of the 3D tornado dataset.

The influence of the special choice of the transfer function on the final image is shown in Figure 5. Displaying only very high densities leads to structure extraction seen in (a). 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, rendered using different transfer functions and the explicit 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

submitted to Eurographics/IEEE-VGTC Symposium on Visualization (2006)
Figure 4: 2D slice through a model of the New Orleans and airflow over this model dataset showing the structure. The grey regions represent the buildings in the model. (a) shows the structure resulting from a forward advection (b) shows the complementary structure resulting from a backward advection. (c) shows a blending of both images with the complete structure. (d) LIC representation of the same slice.

Figure 5: 2D slice of the New Orleans airflow dataset showing the same density function rendered with two different transfer functions: (a) focus on flow structure; (b) exhibits the overall flow behavior.

Figure 6: Two-dimensional simulation of a swirling jet, our method for two different transfer functions in comparison with the explicit topology. (The explicit topology image is generated with the software FAnToM developed at the university of Kaiserslautern.)
shown. Focusing on a special region rendering it in a higher resolution can resolve the structure into more detail.

Figure 7 shows the result for a three-dimensional dataset. In these images the particles are propagated forward and backward at the same time. Thus only one density function is generated. Figure 7(a) shows the surface representing a very high density. Figure 7(b) uses a transfer function emphasizing several densities using different colors. Even though we get a rough impression of the field, we do not yet see the nice structures as for two dimensions. Figure 8 shows a visualization of the core of the 3D tornado dataset extracted using a proper transfer function.

The computation times for 2D data sets allowed for interactive rendering during density accumulation. For 3D data sets this was not possible due to the problems described in the implementation section. All the 2D images were generated at 512 × 512 output resolution and 3D images at 256 × 256.

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 to 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 being distracted by non-relevant parts of the field’s domain.

We plan on working on a reduction of artifacts resulting from the regular discretization and boundary. We will develop more flexible transfer functions to make the field structure even more prominent. Another issue is the acceleration of the three-dimensional visualization, where our aim is real-time interactive exploration also for three dimensions.

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 Oliver Kreylos and Wolfgang Kollmann for valuable discussions.

References


submitted to Eurographics / IEEE VGTC Symposium on Visualization (2006)