

An Illustrative Visualization Framework for 3D Vector Fields

Cheng-Kai Chen¹, Shi Yan¹, Hongfeng Yu², Nelson Max¹, Kwan-Liu Ma¹,

¹Department of Computer Science, University of California, Davis

²Sandia National Laboratories

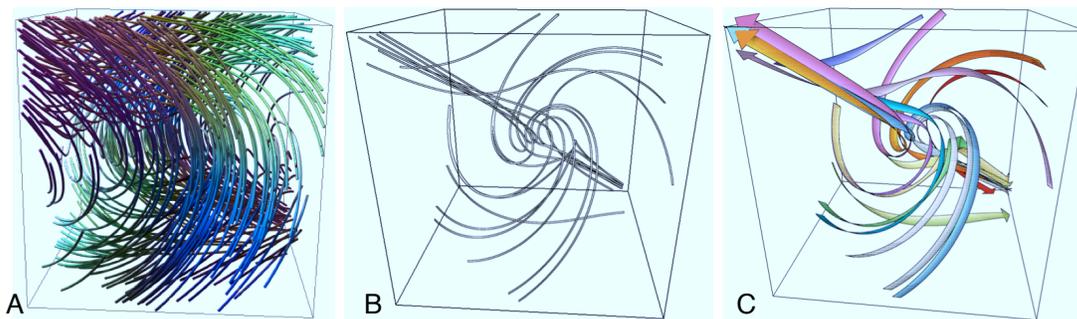


Figure 1: A) A direct streamline visualization of a 3D linear vector field exhibits severe visual clutter. B) A few intermediate representative streamlines generated by our approach capture the essential flow pattern. C) The final illustrative visualization using streamtapes enhances the depth cues for streamlines and succinctly conveys both directional and structural information.

Abstract

Most 3D vector field visualization techniques suffer from the problem of visual clutter, and it remains a challenging task to effectively convey both directional and structural information of 3D vector fields. In this paper, we present a novel visualization framework that combines the advantages of clustering methods and illustrative rendering techniques to generate a concise and informative depiction of complex flow structures. Given a 3D vector field, we first generate a number of streamlines covering the important regions based on an entropy measurement. Then we decompose the streamlines into different groups based on a categorization of vector information, wherein the streamline pattern in each group is ensured to be coherent or nearly coherent. For each group, we select a set of representative streamlines and render them in an illustrative fashion to enhance depth cues and succinctly show local flow characteristics. The results demonstrate that our approach can generate a visualization that is relatively free of visual clutter while facilitating perception of salient information of complex vector fields.

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

1. Introduction

With the rapid growth of computing power, scientists are able to simulate and study complex physical phenomena in unprecedented detail. However, given the sheer size and complexity of data, it is a non-trivial task for scientists to study the output from large simulations. In particular, exploration of 3D vector fields using visualization often results in visual clutter due to a potentially high density of visual

primitives presented in renderings. Therefore, it remains a challenge to effectively generate an informative picture from a complex 3D vector field.

A wealth of techniques have been developed for visualizing vector fields. Among them, streamline visualization is a simple but common way to convey the structure of a 3D vector field. By integrating the flow path, a set of streamlines can reveal flow patterns in an intuitive fashion. How-

ever, there are two main problems with this approach. First, if the seeds of streamlines are not carefully placed, the resulting streamlines can miss important features of the underlying field, thereby failing to convey the actual flow structures inherent in the data. Second, the streamlines generated from a 3D turbulent flow are typically intertwined and occlude one another, exhibiting high depth complexity. Thus, a rendering of 3D streamlines can easily exhibit visual clutter, hindering users from perceiving structural information.

To address these issues we propose an approach that is inspired by the principles of *abstraction* and *emphasis* used by artists and illustrators to efficiently and precisely convey information [SS02]. The idea behind abstraction and emphasis is to display abstractions that preserve certain precise qualities of objects while suppressing or omitting unnecessary details. The resulting images can direct the user's focus of attention to salient properties and characteristics of the rendering. The effectiveness of this illustrative rendering methodology has been shown by a wide class of applications from cartoon animation, stylish rendering, and technical illustration to scientific visualization.

In this work, we study the issue of 3D vector field visualization and present a novel framework to generate illustrative visualizations of 3D vector fields by applying the principles of abstraction and emphasis. Figure 1 shows a sample result with a synthesized 3D vector field, illustrating the effectiveness of our approach. Our illustrative visualization work makes the following contributions:

- We present a method to abstract a 3D vector field by a set of streamlines that are representative and cover the important regions of flow.
- We develop a new two-stage k -means clustering algorithm that can effectively and efficiently decompose a large number of streamlines into different groups, wherein each group has a coherent or nearly coherent flow structure, facilitating abstraction of complex structures.
- We choose to render streamlines using a streamribbon-like visual metaphor, the *streamtape*, that not only effectively enhances depth perception of streamlines, but also concisely depicts characteristics of vicinity flow along streamlines.

We demonstrate our framework with several 3D vector fields. We show that the resulting visualizations, composed of streamtapes, are illustrative and provide the informative depictions of the vector fields.

2. Related Work

Researchers have developed a large number of vector field visualization techniques, and an overview of flow visualization can be found in [WE05]. Here we discuss the references that are most relevant to our work.

2.1. Streamline Visualization

Among vector field visualization techniques, streamline visualization is one of the most popular methods. The effectiveness of streamline visualization relies on the appropriate placement of streamline seeds. Verma *et al.* [VKP00] introduced a method that segments a vector field into different regions based on detected critical points. In each region, a template is used to guide seed placement according to the types of critical points. Ye *et al.* [YKP05] presented a method for placing streamlines in a 3D flow field with the goal of presenting all significant flow patterns with maximal coverage and minimal clutter. Spencer *et al.* [SLCZ09] introduced an image-based seeding strategy that efficiently generates evenly-spaced streamlines for any general surface-based vector field. Li and Shen [LS07] noted that streamlines that are generally well organized in 3D may still produce a cluttered view when projected onto the screen. Therefore, they developed an image-based seeding algorithm that places seeds in the image plane in a clutter-free manner so that the depths and structures of streamlines can be clearly presented. Li *et al.* [LHS08] devised a method for illustrative streamline placement, where new streamlines are only created when they represent flow characteristics that have not been shown by neighboring streamlines. In this way, the display of extraneous repetitive flow patterns can be effectively avoided. Marchesin *et al.* [MCHM10] proposed a streamline removal and placement technique that combines both view-independent (i.e., data space) and view-dependent (i.e., screen space) information to construct an intelligible and uncluttered picture.

In our work, we propose an entropy-based seeding technique to prioritize different flow areas so that important flow features can be detected and emphasized.

2.2. Vector Field Simplification

Clustering is a widely used technique for simplifying vector fields. Conventional approaches include physics-based clustering, top-down clustering, and bottom-up clustering.

For physics-based clustering and visualization, Griebel *et al.* [GPR*04] used an algebraic multigrid algorithm to obtain a multilevel decomposition of flow structures. Diewald *et al.* [DPR00] proposed an approach to segment a vector field based on anisotropic nonlinear diffusion. Garcke *et al.* [GPR*01] introduced specific physical quantities to control the shape, orientation, and distribution of the clusters as a function of the underlying field.

Heckel *et al.* [HWHJ99] introduced a top-down hierarchical clustering method by splitting groups of voxels iteratively. Their approach allows varied resolutions in the hierarchy to be visualized simultaneously. Telea and Wijk [TvW99] proposed a bottom-up method which repeatedly merges neighboring groups of voxels with the highest similarity. Simplified visualization results can be obtained at

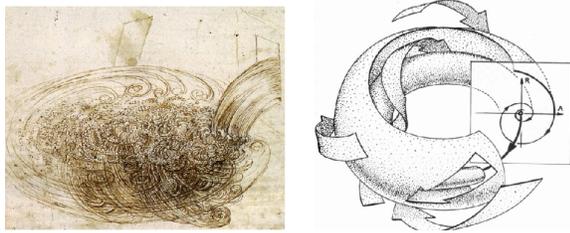


Figure 2: Left: Leonardo Da Vinci's hand-drawn water eddy. Right: Abraham and Shaw's streak arrow surface.

different levels of detail. Both approaches assume no particular structure of the field, nor do they require any topological connectivity information.

In our work, we develop a two-stage k -means algorithm to cluster streamlines based on their geometrical properties and entropy metrics, and to efficiently extract representative streamlines to depict the essence of flow fields.

2.3. Illustrative Visualization

Illustrative rendering has proven useful in showing 3D structures due to its ability to hide irrelevant details while emphasizing important features. This rendering style has traditionally been used by scientists to display pictures in textbooks. Two hand-drawn illustrations related to our work are Leonardo Da Vinci's study of water eddies, and Abraham and Shaw's streak arrow surface [AS92], both shown in Figure 2.

Illustrative rendering has been a hot topic of research in the computer graphics community [GG01, SS02]. In particular, problems arising from shading, outline inking, and the application of various artistic styles are currently under debate [VB99, CM02]. It has been argued that illustrative rendering styles are better able to present object structures. Sousa *et al.* [SFWS03] introduced a way of rendering 3D objects using a precise ink style, which mimics textbook illustrations. Other examples include the methods by Saito *et al.* [ST90], Gooch *et al.* [GGSC98] and Schlechtweg *et al.* [SS00].

In the area of visualization, non-photorealistic rendering has received much attention, especially in volume rendering. For example, Bruckner *et al.* [BG07] reproduced the anatomical illustration style found in medical books. However, the usefulness of illustrative approaches in vector field visualizations has not been studied as thoroughly as in other areas. To visualize 2D vector fields, Kirby *et al.* [KML99] presented an illustrative visualization inspired by paint strokes. Most illustrative visualizations of 3D vector fields have been designed with respect to streamlines or integral surfaces. Illustrative methods for streamlines typically contain no depth information and often result

in high visual complexity. Some researches have applied halos, lighting, and colors to lines in order to increase their visual salience [DHL09, EBRI09]. However, even with these methods, high visual complexity in 2D projections of 3D streamlines still persists.

Illustrative methods for integral surfaces can be geometrically complex. Born *et al.* [BWF*10] recently published their method of illustrative rendering of stream surfaces, which combines integral surfaces and cross sections of integral surfaces, as well as streamlines sampled on integral surfaces. Hummel *et al.* [HGH*10] also provided an illustrative rendering framework for integral surfaces. Their method sets transparency based on the view-normal angle and the normal variation to the integral surface, and also assigns a texture pattern to the surface, which counters surface stretching. However, in order to generate a proper integral surface, the initial parameters of the integral surface (such as the start curves) need to be carefully determined. Automatic placement of start curves remains a difficult problem, and conventional integral surface methods generally require intensive manual work to place start curves for complex flows. Moreover, displaying multiple integral surfaces simultaneously can easily lead to high visual complexity.

3. Illustrative Visualization Framework

By applying the principles of abstraction and emphasis for illustrative rendering, we address the issue of effectively visualizing 3D vector fields based on streamlines. We aim to derive an abstraction that captures essential flow features while suppressing unimportant details, with the goal of minimizing visual clutter.

3.1. Overview

The proposed approach consists of four steps: entropy-based seeding, two-stage streamline clustering, streamtape generation, and illustrative rendering. Figure 3 illustrates an overview of the framework.

In the first step, we select streamlines to capture the intrinsic features of a flow. We adopt Shannon's entropy from information theory as a guide to place the seeds, where the local densities of streamline seeds are determined according to local information measurements. In this way, we can generate a set of streamlines whose spatial locations reflect the local importance of flow in a quantitative manner, thereby capturing the critical features of interest.

The number of streamlines generated in the first step can be large. Thus in the second step, streamline clustering, the object is to simplify the streamline set by identifying and extracting the representative streamlines. We develop a two-stage k -means clustering method for decomposing a large number of streamlines into groups, where each group contains coherent or nearly coherent streamline bundles. Based

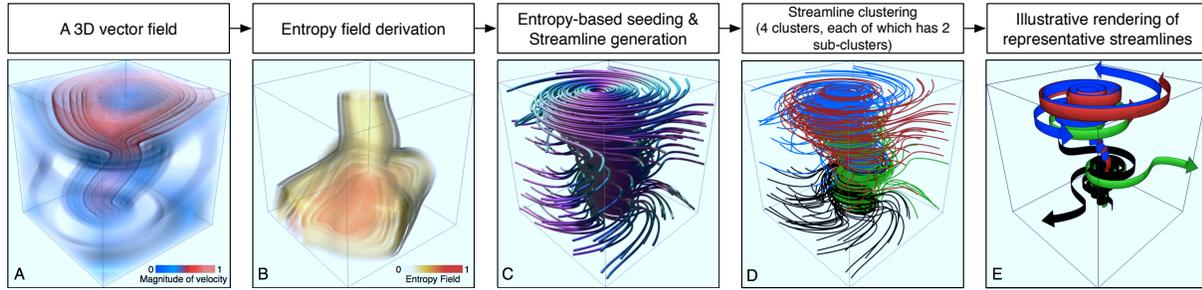


Figure 3: An overview of our illustrative visualization framework for 3D vector fields. A) The volume rendering of velocity magnitude for a tornado dataset where the red color indicates the voxels with higher velocities. B) The entropy volume shows that the bottom-center regions have higher entropy values. C) The generated streamlines based on our entropy-based seeding strategy. D) Streamline clustering results where eight representative streamlines are generated (four clusters, each of which has 2 sub-clusters). E) The illustrative rendering of representative streamlines in each bundle.

on user preference, one or multiple representative streamlines are selected from each bundle, reducing the overall number of streamlines while retaining the essential directional and structural information of each bundle.

From the second step, we obtain a minimal set of streamlines that represent an abstraction of the underlying flow field. We note that directly visualizing these streamlines as lines or cylindrical shapes may not necessarily lead to an effective visualization. First, this small set of streamlines is typically distributed sparsely in space, and the resulting image may not provide sufficient depth cues for users to correctly perceive the relative depth of structures. Second, a single streamline provides primarily directional information and is generally not effective at providing visual information on other important characteristics of flow. Although researchers have developed different methods (such as halos [EBRI09] shading [MPSS05], and volume rendering [SM04]) to enhance depth perception, these methods are mainly designed for dense line datasets. Moreover, there is still a lack of sophisticated methods incorporating additional flow information in streamline rendering, with the exception of some simple practices such as coloring streamlines with certain quantities (like velocity magnitude).

In our work, we choose to use a streamribbon-like visual metaphor, called streamtape, to visualize streamlines. A streamtape is a surface that is constructed along a streamline, and its thickness and orientation are computed with respect to the local curvature and torsion information of the streamline. The advantages of the streamtape are threefold. First, by representing a streamline as a surface, we can provide more depth cues and enhance user perception of the depth between different structures. Second, the directional information can be enhanced by the surface of a streamtape that is orientated along the skeletal streamline. Third, apart from the directional information, streamtapes also convey curvature and torsion, two of the most important characteristics

of flow, allowing users to observe the local flow structural information along the streamlines.

In the final step, we use an illustrative rendering method to visualize the streamtapes. We use silhouettes as the visual cue to simplify the depiction of streamtapes. The twists of streamtape surfaces are revealed by half-tone shadow. In this way we can effectively emphasize the intrinsic directional and structural information of a streamtape while omitting unnecessary details to generate our final illustrative visualization results.

3.2. Entropy-based Seeding

Given a flow field, we first place seeds for streamline generation. The primary goal of our seeding strategy is to capture the essence of the flow data. To achieve cost-effective seeding without an extensive knowledge of the dataset, based on the concept of Shannon's entropy in information theory we first calculate an entropy scalar field [XLS10] to measure the information content in each local region of the vector field. Shannon's entropy is defined as:

$$H(x) = - \sum_{x_i \in X} p(x_i) \log_2 p(x_i), \quad (1)$$

where H is the entropy of a discrete random variable X with a sequence of possible outcomes values $\{x_1, \dots, x_n\}$, and p is the probability mass function of X . In the flow field data, the function p can be approximated by creating a histogram of vectors. By following the techniques proposed by Leopardi [Leo06], this task can be done by using vector quantization, in which the vectors are partitioned into a number of bins x_i . Hence, the probability function p is defined as follows:

$$p(x_i) = \frac{C(x_i)}{\sum_{i=1}^n C(x_i)}, \quad (2)$$

where $C(x_i)$ is the number of vectors in the bin x_i .

The entropy field is a scalar field which is constructed by calculating the entropy around the local neighborhood of each voxel. We set the number of bins in the histogram and the size of the local neighborhood as suggested by Xu *et al.* [XLS10]. For each voxel, we define its probability of being seeded to be proportional to its scalar value in the entropy volume. We then place a sufficient number of seeds in the field based on the probability associated with each voxel so as to cover as many important areas as we can. This allows us to emphasize important regions with high entropy values. Finally, we use the fourth order Runge-Kutta integration to trace the streamlines from the seeds.

Figure 3 (B) shows a volume rendering of the entropy field of a synthesized tornado dataset. We can see that the tornado core has higher entropy values, corresponding to the nature of turbulent flow present in that area. Figure 3 (C) shows the streamlines generated based on our entropy-based seeding strategy. We can see that more streamlines are placed to cover the tornado core area, ensuring that the important flow features are well captured.

3.3. Streamline Clustering

After obtaining a number of streamlines based on our entropy-based seeding strategy, we then need to consider how to effectively draw them. Visualizing 3D streamlines is non-trivial: projecting too many lines to a 2D screen can result in unreadable pictures, while too few lines cannot be guaranteed to convey complete vector field information. Hence, to generate a concise but informative picture, it is necessary to have an efficient simplification method to keep a small set of streamlines which capture the essence of the underlying field.

We achieve the simplification of streamlines through clustering. Moberts *et al.* [MVvW05] evaluated several clustering methods for line data. They concluded that the hierarchical single-link clustering with the mean of closest-point distances gives the best result. However, this clustering method has a quadratic computational complexity due to pairwise similarity evaluation, which can be inefficient for a large line dataset in practice. In our framework, we instead use a k -means algorithm to classify the initial set of streamlines into several bundles, and select one or multiple representative streamlines from each bundle using an entropy-based metric. In this way, we can efficiently identify and isolate the representative streamlines of interest and prune a large number of streamlines with high similarities.

The k -means algorithm generally uses an iterative refinement heuristic called *Lloyd's algorithm*. Lloyd's algorithm initially partitions the input data points/vectors into k sets, either randomly or heuristically. Then, the algorithm iteratively computes the centroid of each set and re-associates each point/vector with its closest centroid to construct a new partition. The iteration stops when some convergence condition is met.

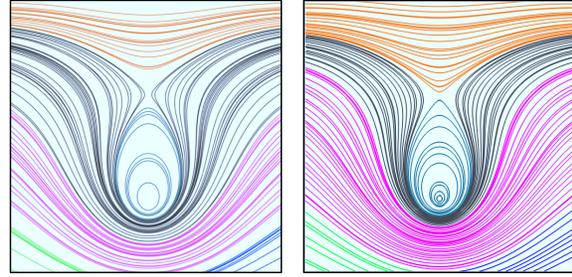


Figure 4: Comparison of our method (left) with the clustering method in [MVvW05] (right), using a synthesized dataset. Different color represents different streamline cluster.

We note that the complexity of the k -means algorithm is mainly determined by the number of points/vectors and their lengths. Most existing algorithms for line clustering represent a line as a vector of the original line points [MVvW05], and thus the complexity of algorithm can be high increased with a large number of long streamlines. For this reason, we designed a new two-stage k -means clustering algorithm, in which each streamline is represented as a short 1D feature vector that contains the spatial and shape properties of the streamline. Our k -means algorithm operates in two stages with respect to these two properties.

First of all, we want to obtain an overview of the flow patterns. In the first stage, we partition the streamlines into p bundles, where the input to the k -means algorithm is an additional vector of shape properties. For a streamline, the vector of spatial properties comprises its start point, middle point, and end point. In this way we obtain an initial partition containing p bundles, where each bundle has a roughly coherent structure in space.

Then, in the second stage, for each bundle generated in the first stage, we further partition it into q_i sub-bundles, where $i = 1 \dots p$ and the input to the k -means algorithm is the vectors of shape properties. For a streamline, the vector of shape properties comprise its linear and angular entropy values. Proposed by Furuya and Itoh [FI08], the linear entropy E_L quantifies the amount of information in the flow magnitude along the streamline, which is define as follows:

$$E_L = -\frac{1}{\log_2(m+1)} \sum_{j=0}^m \frac{D_j}{L_S} \log_2 \frac{D_j}{L_S} \quad (3)$$

where m is the number of streamline segments, D_j is the length of the j -th segment and L_S is the total length of the streamline (the sum of the D_j values).

Introduced by Marchesin *et al.* [MCHM10], the angular entropy E_A quantifies the amount of angular variation along

a streamline:

$$E_A = -\frac{1}{\log_2(m)} \sum_{j=0}^{m-1} \frac{A_j}{L_A} \log_2 \frac{A_j}{L_A} \quad (4)$$

where m is the number of streamline segments, A_j is the absolute value of the angle at the j -th streamline joint and L_A is the total angular variation along the streamline (the sum of the absolute values of the A_j). The combination of linear and angular entropy values can be used to discriminate streamline properties and detect flow phenomena. To this end, we further decompose each bundle into a set of refined sub-bundles, where each sub-bundle has coherent structures in terms of both the spatial and shape properties. Within each sub-bundle, we sort all streamlines based on their distances to the centroid, and select one or multiple streamlines that are closest to the centroid as the representative streamlines.

The total number of clusters c equals $\sum_{i=1}^p q_i$. At run time, either a user can specify the values of p and q_i , or our system can use the classic method of elbow criterion [KS96] to suggest the optimal number of clusters that gives the best inter-cluster separation. The elbow criterion determines the optimal number of clusters based on the *percentage of variance explained* which is the ratio of the within-cluster variance to the total variance. The within-cluster variance indicates the information carried on each cluster, and the total variance denotes the amount of information contained in the data. This ratio represents how much data difference in terms of variance is captured in the clustering. During the optimization process, the system incrementally increases the number of clusters and uses the elbow criterion to pick the number of clusters which has an abrupt change of the *percentage of variance explained ratio*. In this work, the within-cluster variance is obtained by calculating the squared distance between each streamline and the respective centroid for each cluster.

Figure 4 demonstrates that our method can generate a result similar to the hierarchical single-link clustering method with the mean of closest point distances. Since each vector for a streamline contains only five components, our algorithm is more efficient than the conventional k -means algorithms for clustering a large number of lines. Figure 3 (D) shows our streamline clustering results of the synthesized tornado dataset. We can clearly see the flow field is well partitioned into four clusters, and each cluster contains two sub-clusters, exhibiting several coherent flow patterns.

3.4. Streamtape Generation

To render the representative streamlines, we choose to use a streamribbon-like visual metaphor, the streamtape, by leveraging the advantages of both streamlines and integral surfaces. The streamribbon has been extensively studied in previous research. A streamribbon is constructed by connecting two edges. The first edge is a calculated streamline, while

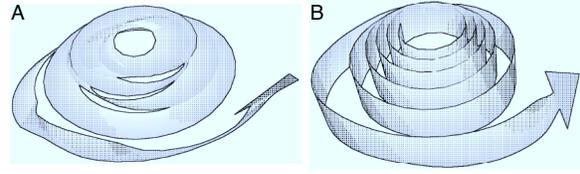


Figure 5: The comparison of a streamribbon and a streamtape generated from a same streamline. A) shows the visual flaw of a streamribbon if the first normal vector is not carefully chosen. B) shows the streamtape generated from the same streamline.

the second edge is generated based on the first edge and usually maintains a constant distance away from the first edge. The second edge can be calculated by rotating a normal vector about the first edge, and each rotation angle of the normal vector is corresponding to the local curl evaluated on the first edge [DH92, MS93, USM96]. Therefore, the twists of a streamribbon can faithfully reflect the vorticity of the local flow. However, the orientation of the first normal vector needs to be carefully chosen to avoid visual flaws such as self-overlap. This visual flaw is demonstrated in Figure 5 (A). Finding a good orientation for the first normal vector is non-trivial and requires much effort. The conventional method may not guarantee a good streamribbon. In particular, for a long and complex streamline, when a visual flaw at certain part of the streamribbon is resolved by rotating the first normal vector, other flaws may appear on other parts of this streamribbon.

We use streamtapes to generate both meaningful and visually pleasing visualization results. A streamtape is constructed in the following steps. First, we sample a series of points at an equal arc length along each streamline. And then we generate an orthogonal line segment at each sample point to the streamline tangent. The direction and length of a line segment are determined by the local torsion of the flow. By connecting the successive line segments, we obtain the surface of a twisted tape with a varying width. Finally, we put an arrow at the end of a streamtape to indicate its direction.

In contrast to streamribbons, our streamtape is more flexible in managing visual clutter and conveying important flow characteristics. The twist and width of a streamtape are determined according to the local flow torsion with specified parameters, which allows us to not only easily control the shape of a streamtape to reduce visual complexity, but also convey important characteristics of the local flow, as displayed in Figure 5 (B).

For a streamtape, the orientation of each tape segment is defined by the binormal vector B at the sample point of the

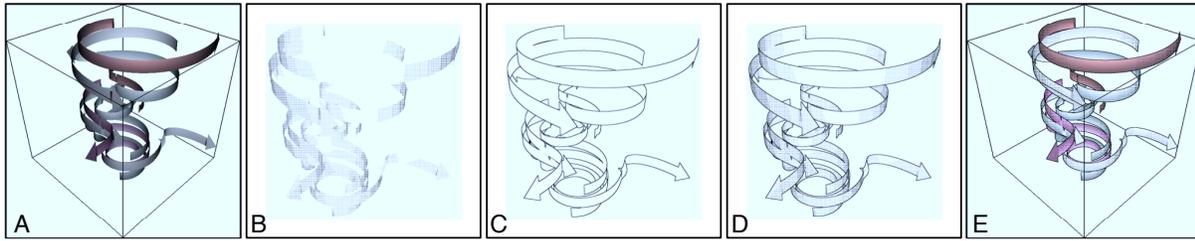


Figure 6: The work flow of our illustrative render. A) The Phong shaded streamtapes. B) The half-tone styled shadow. C) The outlines. D) The composed illustrative image. E) The final result.

streamline $r(s)$:

$$\begin{aligned} B &= T \times N \\ T &= dr/ds \\ N &= dT/ds \end{aligned} \quad (5)$$

where T and N are the tangent and normal vectors of the local field at a sample point, respectively. If we do not use the binormal direction as the line segment direction, the streamtape can be bent along the rotation direction of the streamline, incurring visually complex and aesthetically unpleasant results as in Figure 5 (A). The length of a line segment is determined by the torsion τ of the local field. Torsion is a measurement of how sharply the local field twists and is given by:

$$\tau = \frac{\det(r', r'', r''')}{\|r' \times r''\|^2} \quad (6)$$

where r', r'', r''' are, respectively, the first, second, and third derivatives of $r(s)$ with respect to the arclength parameter s . We want to reflect the torsion values by adjusting the streamtape width accordingly. However, when the local flow twists sharply, the surface of a streamtape can self-intersect. To address this issue, we normalize the torsion τ to a range of $[0, 1]$ and the streamtape width ω is given by $\omega = 1.0 - \tau$.

3.5. Illustrative Rendering

Figure 6 shows the process of the illustrative streamtape rendering. Starting from a standard Phong shading shown in Figure 6 (A), the luminance of the image is recorded and stored as a framebuffer object. To better display depth and rotation information, we implement illustrative shadow using half-tone hatching rendering styles to incorporate lighting. For an overview of half-tone rendering, we refer the reader to the work of Freudenberg *et al.* [FMS01]. We obtain the hatch shadow image shown in Figure 6 (B). Note that Hummel *et al.* [HGH*10] has introduced a procedural texture map which generates a line stripe pattern on an integral surface along the advection direction of the surface. The density of the stripe pattern is adaptive to the stretching degree of the integral surface, thereby giving a hint of how the integral surface is being stretched. However, this method is

not compatible with our goal, since the proposed streamtape is a narrow object on which a complex pattern will lead to visual complexity.

Once we get the hatched shadow, we render the same scene again to obtain the silhouettes of objects. During the rendering, the normalized normal direction of each fragment is written to the RGB channel of the output color, and the alpha channel records the depth information clamped to $[0, 1]$. Furthermore, a different shader program checks the discontinuities of the outputs in image space in the up, down, left and right directions, and marks pixels with large discontinuities as the pixels on the silhouettes, as shown in Figure 6 (C). In the last step, as shown in Figures 6 (D) and (E), both the shadow image and the silhouette image are composed with the Phong shaded image. We provide different ways of coloring streamtapes. A straightforward approach assigns different colors to different clusters based on a predefined color table. An alternative way is to assign these colors based on local features of tapes, for instance, the local curvature information. This coloring strategy helps to reveal some structural information for certain datasets.

4. Results and Discussion

In our framework, entropy field calculation and streamline generation are implemented using CUDA on the GPU, and k -means clustering and streamtape generation are implemented on the CPU. Table 1 shows the performance of our framework. These results were obtained on a Xeon 5450 machine with 16GB of memory and a Geforce GTX 480 graphics card with 1.5GB of video memory. As shown in Table 1, for the given flow datasets, our framework can quickly complete these calculations within a few seconds. The time for entropy calculation is proportional to the data size and inversely proportional to the number of bins and neighborhood size [XLS10]. Note that this calculation is a one-time operation and that the calculated entropy field can be reused. The processing time for clustering is proportional to the number of streamlines and clusters, and the processing time for streamtape generation is proportional to the number of representative streamlines (usually equal to the number of clusters). In the # of clusters column, we show the total number

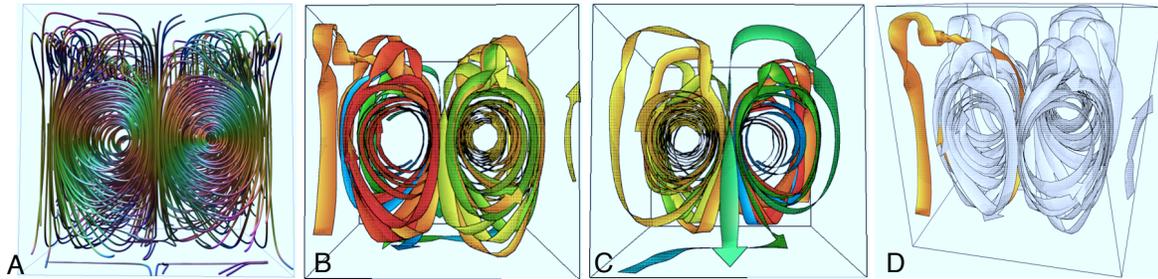


Figure 7: Visualization of a dataset with two swirls. A) shows a direct streamline visualization. B), C), and D) show the results obtained using our method. B) and C) show the dataset from the front and back, respectively, and D) highlights one of the most twisted tapes.

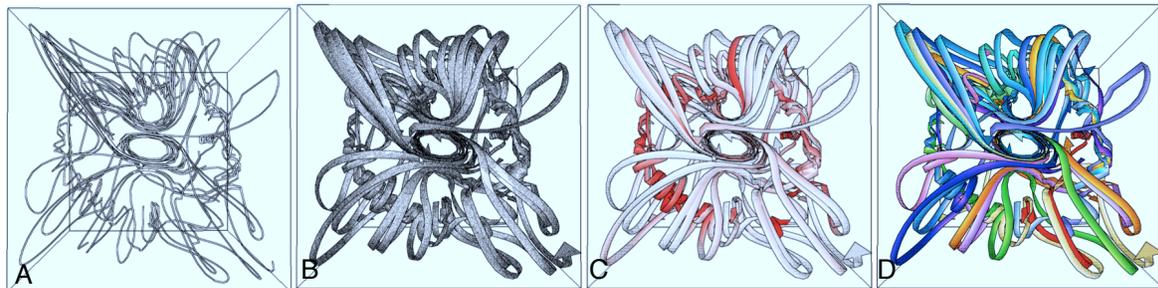


Figure 8: Comparison of different rendering results for a flame dataset. A) shows the representative streamlines. B) shows the representative streamtapes. In C), the regions of highest curvature and torsion are shaded in red to reveal internal structure. In D), different colors are used to illustrate different clusters.

of clusters c , followed by the exact values of clustering parameters p and q_i .

Figure 1 shows a comparison of three different vector field visualization techniques. In this dataset, the main flow is first attracted to the center, spiraling inward, and then escapes from the center to the two opposite corners of the volume. These patterns are difficult to discern using the streamline visualization shown in Figure 1 (A) because streamline occlusion hides most of the internal structures. In particular, the twisting and depth information located in the center of the dataset are not visible. Figure 1 (B) is a rendering of the representative streamlines selected by our approach. Figure 1 (C) shows the results obtained using our illustrative rendering approach, which we believe is better at revealing the structure of 3D vector fields. Note that in Figures 1, 7, 9, and 11 (A), we color the streamlines using Boy's surface immersion [DHL09] according to the local velocity direction.

Figure 7 shows a visualization of the "two swirls" dataset. Compared with the simple 200-streamline rendering shown in (A), our visualizations, shown in (B), (C), and (D), provide more concise and informative images; only 16 lines need to be drawn. With our technique, the two swirls with opposing rotations in the center can be easily perceived. Furthermore, our visualization provides more depth information

thanks to our shading and color strategies, and it is easier to see distinct layers in the data. Our result also provides a clear description of the flow directions. In Figure 7 (D) we highlight one of the twisted streamtapes captured by our simplification approach.

Figure 8 contrasts different rendering styles using a flame dataset. From this figure, we can clearly see that the dataset contains complex flow patterns organized into a main twisted ring structure and wrapped around by many small vortical structures. We can further tell that these small vortical structures converge in the central area to form a smaller, but denser, vortical structure. The arrows on the streamtape succinctly depict directional information. More specifically, Figures 8 (A) and (B) compare the representative streamline rendering with our streamtape rendering. The representative streamlines shown in Figures 8 (A) cannot clearly reveal the layered structures of the dataset, while the streamtapes shown in Figure 8 (B) clearly display the different flow patterns and the interplay between them. To better visualize the internal ring structure, we color the streamtapes by the local curvature and torsion values, where the areas with high curvature and torsion values are shaded in red. Because the internal structure of the flow is more turbulent, that structure appears to be more red in the image and thus becomes

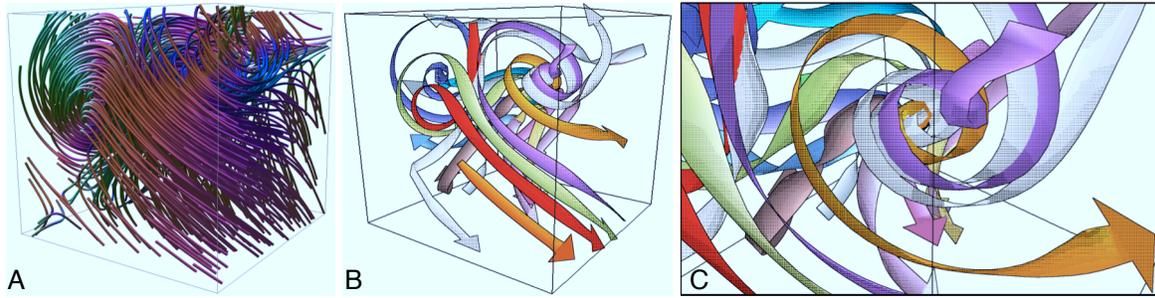


Figure 9: Visualization of dataset containing multiple vortices. A) shows a simple streamline visualization. B) depicts the representative streamtapes, which show two vortices. C) shows a closer look at the center of one of the vortices.

Dataset	Dimension	# of lines	# of clusters	Entropy	Clustering	Streamline	Streamtape
Hurricane (Fig. 10)	$200 \times 200 \times 40$	400	21 (8:3,2,4,2,3,2,3,2)	5.3	3.6	0.41	0.07
Plume (Fig. 11)	$126 \times 126 \times 512$	500	26 (8:4,3,3,5,4,2,2,3)	7.1	4.2	0.44	0.09
Linear (Fig. 1)	$100 \times 100 \times 100$	500	13 (4:3,3,3,4)	4.2	3.3	0.42	0.05
Tornado (Fig. 3)	$64 \times 64 \times 64$	200	8 (4:2,2,2,2)	1.8	0.53	0.23	0.01
Two swirls (Fig. 7)	$64 \times 64 \times 64$	200	16 (5:3,3,3,3,4)	1.7	0.66	0.2	0.047
Flame (Fig. 8)	$65 \times 65 \times 65$	200	20 (5:5,5,5,5,5)	2.1	0.82	0.21	0.058
Two vortices (Fig. 9)	$51 \times 51 \times 51$	200	17 (5:3,2,3,4,5)	1.5	0.745	0.19	0.06

Table 1: Performance results for our framework using seven different datasets, different initial streamline numbers, and different cluster numbers. Execution times for entropy field calculation, streamline clustering, and streamline and streamtape generation are shown in the table. Times are given in seconds.

more visible. Figure 8 (D) displays different clusters in different colors, with each cluster representing a coherent flow pattern.

Figure 9 shows a dataset with two vortices. Figure 9 (A) shows a streamline visualization of this dataset, where the internal structure is hard to perceive because of visual occlusion. Figure 9 (B) shows our streamtape visualization of this dataset, where two vortices can be clearly represented using just a few streamtapes. Figure 9 (C) shows a zoomed-in view, where local details such as torsion and curvature can be discerned.

As shown in Figure 10, this hurricane dataset contains two large, turbulent patterns, and our method is able to clearly represent the two main vortices using 21 streamtapes. Figure 11 shows another application of our method to a turbulent solar plume dataset. Compared with the streamline visualization shown in Figure 11 (A), our streamtape visualization more successfully depicts the twisted internal structures.

5. Conclusions and Future Work

In this paper, we have presented a new framework for visualizing 3D vector fields. Our framework combines an entropy-based seeding strategy, a clustering-based simplification approach, and illustrative rendering techniques to

generate concise and informative visualizations for complex flow structures. Starting from a 3D flow dataset, our framework efficiently generates an initial set of representative streamlines based on the entropy, or information content, in the field. We then employ a two-stage k -means clustering algorithm to further refine these streamlines into a subset of streamlines that captures essential flow features. Rendering the resulting streamlines as streamtapes using our illustrative rendering techniques allows us to successfully enhance depth cues and succinctly capture local flow characteristics. We believe that our framework provides domain experts a powerful visualization tool for displaying complex flow fields and allows efficient interpretation of their data.

There are a number of directions that we would like to pursue in order to further improve our framework. First of all, currently the representative streamlines of each individual cluster are selected based on their distances to the centroid of the cluster. We think that taking camera view angles into account could further reduce visual clutter due to streamline occlusion.

Second, we would like to apply our techniques to unsteady flow fields. One possible direction is to adopt a high-dimensional data representation that unifies vector field visualization methods for steady and unsteady flows [YWM07]. Another possible direction is to establish a correspondence

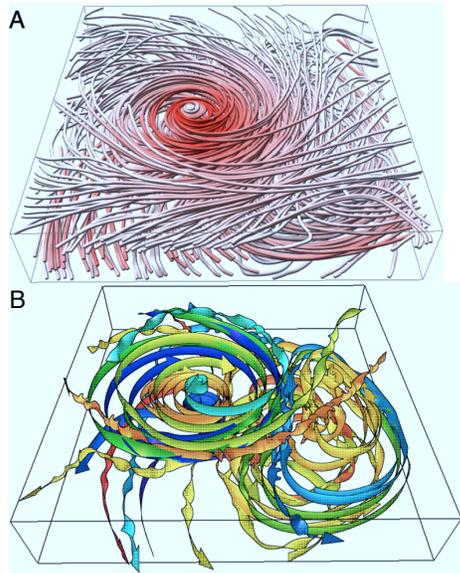


Figure 10: Visualization of a hurricane dataset. A) shows a direct streamline visualization, where the regions with high-magnitude vectors are shaded in red. B) shows 21 representative streamtapes.

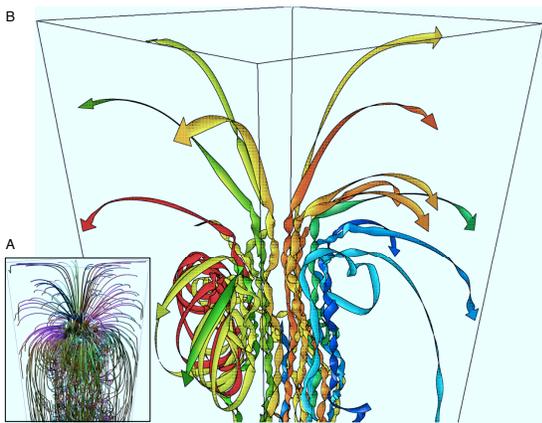


Figure 11: Comparison of A) a streamline visualization of a solar plume dataset and B) our concise streamtape visualization result of the same dataset.

between visualizations of successive frames, resembling the work of Jobard and Lefer [JL00]. We would like to investigate these possibilities to generate both spatially and temporally coherent results for unsteady flow fields.

Third, we would like to support level-of-detail (LOD) in our visualization results. Although our current method can generate concise visualizations for complex datasets, we have noticed that more clusters are required to ensure meaningful results for more turbulent flows, which may increase

visual complexity. In the future, we would like to explore the possibility of generating results at different LODs, from global coverage to fine details, from a given flow dataset.

Fourth, although entropy calculation and clustering are one-time preprocessing steps in our framework, we would like to develop parallel algorithms to carry out these operations using multiple GPUs and CPUs, making our approach truly scalable for large datasets.

Finally, our streamtapes generally lead to effective visualizations of local flow characteristics along streamlines. We would like to conduct a more detailed study of the effectiveness of sparse streamtape visualization. One possible approach may be based on Finkelstein *et al.*'s work [CSD*09], where an experiment was designed to study how well people interpret shapes from line drawings. Similarly, we would like to study user performance in locating certain types of flow features using our streamtape visualizations compared to conventional streamline visualizations, and gain more insights into how different parameters improve the effectiveness of streamtape visualizations.

6. Acknowledgments

This work has been sponsored in part by the U.S. Department of Energy through the SciDAC program with Agreement No. DE-FC02-06ER25777, and by the U.S. National Science Foundation through grants OCI-0749217, CCF-0811422, CCF-0850566, OCI-0749227, and OCI-0950008. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. The hurricane dataset was provided by Wei Wang, Cindy Bruyere, Bill Kuo, and others at NCAR. The solar plume dataset was provided by Mark Rast of University of Colorado at Boulder.

References

- [AS92] ABRAHAM R., SHAW C. D.: *Dynamics: The Geometry of Behavior (Studies in Nonlinearity)*. Addison-Wesley, 1992.
- [BG07] BRUCKNER S., GRÖLLER M. E.: Style transfer functions for illustrative volume rendering. *Computer Graphics Forum* 26, 3 (2007), 715–724.
- [BWF*10] BORN S., WIEBEL A., FRIEDRICH J., SCHEURMANN G., BARTZ D.: Illustrative stream surfaces. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (nov.-dec. 2010), 1329–1338.
- [CM02] CARD D., MITCHELL J. L.: Non-photorealistic rendering with pixel and vertex shaders. In *Direct3D ShaderX, Wordware* (2002), Wordware Publishing, Inc, pp. 319–333.
- [CSD*09] COLE F., SANIK K., DECARLO D., FINKELSTEIN A., RUSINKIEWICZ T. F. S., SINGH M.: How well do line drawings depict shape? In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (Aug. 2009), vol. 28(3).
- [DH92] DARMOFAL D., HAIMES R.: Visualization of 3-D vector fields: Variations on a stream. In *Proceedings AIAA 30th Aerospace Science Meeting and Exhibit* (1992), pp. 92–74.

- [DHL09] DEMIRALP C., HUGHES J., LAIDLAW D.: Coloring 3D line fields using boy's real projective plane immersion. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (nov.-dec. 2009), 1457–1464.
- [DPR00] DIEWALD U., PREUSSER T., RUMPF M.: Anisotropic diffusion in vector field visualization on Euclidean domains and surfaces. *IEEE Transactions on Visualization and Computer Graphics* 6, 2 (2000), 139–149.
- [EBRI09] EVERTS M. H., BEKKER H., ROERDINK J. B., ISENBERG T.: Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1299–1306.
- [FI08] FURUYA S., ITOH T.: A streamline selection technique for integrated scalar and vector visualization. In *IEEE Visualization, Poster Session* (2008).
- [FMS01] FREUDENBERG B., MASUCH M., STROTHOTTE T.: Walk-through illustrations: Frame-coherent pen-and-ink style in a game engine. In *Proceedings of Eurographics* (2001), pp. 184–191.
- [GG01] GOOCH B., GOOCH A.: *Non-photorealistic Rendering*. A K Peters/CRC Press, 2001.
- [GGSC98] GOOCH A., GOOCH B., SHIRLEY P., COHEN E.: A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 447–452.
- [GPR*01] GARCKE H., PREUSSER T., RUMPF M., TELEA A. C., WEIKARD U., VAN WIJK J. J.: A phase field model for continuous clustering on vector fields. *IEEE Transactions on Visualization and Computer Graphics* 7, 3 (2001), 230–241.
- [GPR*04] GRIEBEL M., PREUSSER T., RUMPF M., SCHWEITZER M. A., TELEA A.: Flow field clustering via algebraic multigrid. In *Proceedings of IEEE Visualization Conference* (2004), pp. 35–42.
- [HGH*10] HUMMEL M., GARTH C., HAMANN B., HAGEN H., JOY K.: Iris: Illustrative rendering for integral surfaces. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (nov.-dec. 2010), 1319–1328.
- [HWHJ99] HECKEL B., WEBER G., HAMANN B., JOY K. I.: Construction of vector field hierarchies. In *Proceedings of IEEE Visualization Conference* (1999), pp. 19–25.
- [JL00] JOBARD B., LEFER W.: Unsteady flow visualization by animating evenly-spaced streamlines. *Computer Graphics Forum* 19 (2000), 31–39.
- [KML99] KIRBY R. M., MARMANIS H., LAIDLAW D. H.: Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings of IEEE Visualization Conference: celebrating ten years* (Los Alamitos, CA, USA, 1999), VIS '99, IEEE Computer Society Press, pp. 333–340.
- [KS96] KETCHEN D. J., SHOOK C. L.: The application of cluster analysis in strategic management research: An analysis and critique. *Strategic Management Journal* 17 (1996), 441–458.
- [Leo06] LEOPARDI P.: A partition of the unit sphere into regions of equal area and small diameter. *Electronic Transactions on Numerical Analysis* 25 (2006).
- [LHS08] LI L., HSIEH H.-H., SHEN H.-W.: Illustrative streamline placement and visualization. In *Proceedings of the IEEE Pacific Visualization Symposium* (2008), pp. 79–86.
- [LS07] LI L., SHEN H.-W.: Image-based streamline generation and rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 3 (2007), 630–640.
- [MCHM10] MARCHESIN S., CHEN C.-K., HO C., MA K.-L.: View-dependent streamlines for 3D vector fields. *IEEE Transactions on Visualization and Computer Graphics* 16 (November 2010), 1578–1586.
- [MPSS05] MALLO O., PEIKERT R., SIGG C., SADLO F.: Illuminated lines revisited. In *Proceedings of IEEE Visualization Conference* (2005), pp. 19–25.
- [MS93] MA K.-L., SMITH P.: Cloud tracing in convection-diffusion systems. In *Proceedings of IEEE Visualization Conference* (1993), pp. 253–260.
- [MVvW05] MOBERTS B., VILANOVA A., VAN WIJK J. J.: Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proceedings of IEEE Visualization Conference* (2005), pp. 65–72.
- [SFWS03] SOUSA M. C., FOSTER K., WYVILL B., SAMAVATI F.: Precise ink drawing of 3D models. *Computer Graphics Forum* 22, 3 (2003), 369–379.
- [SLCZ09] SPENCER B., LARAMEE R. S., CHEN G., ZHANG E.: Evenly spaced streamlines for surfaces: An image-based approach. *Computer Graphics Forum* 28, 6 (2009), 1618–1631.
- [SM04] SCHUSSMAN G., MA K.-L.: Anisotropic volume rendering for extremely dense, thin line data. In *Proceedings of IEEE Visualization Conference* (2004), pp. 107–114.
- [SS00] SCHLECHTWEG S., STROTHOTTE T.: Generating scientific illustrations in digital books. In *Smart Graphics, AAAI Spring Symposium* (2000), AAAI Press, Menlo Park, pp. 8–15.
- [SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-photorealistic Computer Graphics: Modeling, Rendering, and Animation (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 2002.
- [ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. *Computer Graphics* 24, 4 (August 1990), 197–206.
- [TvW99] TELEA A., VAN WIJK J. J.: Simplified representation of vector fields. In *Proceedings of IEEE Visualization Conference* (1999), pp. 35–42.
- [USM96] UENG S.-K., SIKORSKI C., MA K.-L.: Efficient streamline, streamribbon, and streamtube constructions on unstructured grids. *IEEE Transactions on Visualization and Computer Graphics* 2, 2 (jun 1996), 100–110.
- [VB99] VERYOVKA O., BUCHANAN J. W.: Comprehensive halftoning of 3D scenes. *Computer Graphics Forum* 18 (1999), 13–22.
- [VKP00] VERMA V., KAO D., PANG A.: A flow-guided streamline seeding strategy. In *Proceedings of IEEE Visualization Conference* (2000), pp. 163–170.
- [WE05] WEISKOPF D., ERLEBACHER G.: Overview of flow visualization. In *The Visualization Handbook*, Hansen C. D., Johnson C. R., (Eds.). Elsevier Academic Press, 2005, ch. 12, pp. 261–278.
- [XLS10] XU L., LEE T.-Y., SHEN H.-W.: An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1216–1224.
- [YKP05] YE X., KAO D., PANG A.: Strategy for seeding 3D streamlines. In *Proceedings of IEEE Visualization Conference* (2005), pp. 471–478.
- [YWM07] YU H., WANG C., MA K.-L.: Parallel hierarchical visualization of large 3D time-varying vector fields. In *Proceedings of the Supercomputing Conference* (2007).