Conference contribution :
Peng, Z. & Laramee, R. (2009). *Higher dimensional vector field visualization: A survey.*(pp. 149-163). IEEE.

EG UK Theory and Practice of Computer Graphics (2009)
Wen Tang, John Collomosse (Editors)

Higher Dimensional Vector Field Visualization: A Survey

Zhenmin Peng and Robert S. Laramee

Visual and Interactive Computing Group
Department of Computer Science
Swansea University, Wales UK
Email: {cszp, r.s.laramee}@swansea.ac.uk

Abstract

Vector field visualization research has evolved very rapidly over the last two decades. There is growing consensus amongst the research community that the challenge of two-dimensional vector field visualization is virtually solved as a result of the tremendous amount of effort put into this problem. Two-dimensional flow, both steady and unsteady can be visualized in real-time, with complete coverage of the flow without much difficulty. However, the same cannot be said of flow in higher-spatial dimensions, e.g. surfaces in 3D (2.5D) or volumetric flow (3D). We present a survey of higher-spatial dimensional flow visualization techniques based on the presumption that little work remains for the case of two-dimensional flow whereas many challenges still remain for the cases of 2.5D and 3D domains. This survey provides the most up-to-date review of the state-of-the-art of flow visualization in higher dimensions. The reader is provided with a high-level overview of research in the field highlighting both solved and unsolved problems in this rapidly evolving direction of research.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Over the last two decades, *vector field visualization* has developed very rapidly. Its applications range widely from the automotive industry to medicine [LEG*08]. As a fascinating sub-branch of scientific visualization, vector field visualization provides solutions which enable users to investigate and analyze some critical features and characteristics of flow phenomena. Although the challenge of two-dimensional flow visualization is deemed virtually solved as a result of the effort invested in this problem, the higher-spatial dimensional flow visualization, (e.g. the visualization on surfaces in 3D (2.5D) and for volumetric flow (3D)), still has many challenges which need to be addressed. We present a survey of higher-spatial dimensional flow visualization techniques in order to provide an up-to-date overview of the most recent developments in flow visualization in higher dimensions highlighting both solved and unsolved problems in this rapidly evolving direction of research.

1.1. Dimensions

A general solution to vector field visualization is not possible to find a common solution to visualize all kinds flow data because different users have different interests, data characteristics vary across datasets, etc.

According to data characteristics, flow visualization techniques can vary considerably: two-dimensional vector field visualization techniques focus on the flow in planar 2D domains; 2.5D flow visualization techniques are for boundary flows in 3D; and 3D flow visualization techniques are solutions for volumetric 3D flow. Although 3D techniques can offer true 3D flow visualization, the computational cost for rendering is high. Hence, compromises are usually needed to trade off visibility for completeness.

In addition to the spatial dimensions described above, temporal dimensions are of great importance in categorizing flow visualization. Steady-state vector field visualization visualizes the instantaneous or static flow representing a single time step while unsteady vector field visualization visualizes the transient or unsteady flow represented by several time steps.

		Direct	Vector-field Clustering	Texture-based	Geometric
2.5D Flow Data	Steady	[PL08] [Lar03]		[LvW95] [vW91] [LPPW95] [LvW95] [BSH97] [SH95]	[vW93a] [SvW91] [SLCZ09] [JL97] [PLCZ09]
	Unsteady			[LJH03] [vW02] [vW03] [vW02] [LvWJH04] [LJH03] [vW03] [LSH04] [LJH03] [LGSH06] [GTS*04] [LvWJH04]	
3D Flow Data	Steady	[Dov95] [CM92]	[TvW99] [HWHJ99] [GPR*01]	[RSHTE99] [CL93] [GEO02] [JEH01]	[SVL91] [Hul92] [MBC93] [ST90] [vW93b] [BHR*94] [WMW86] [LPSW96] [TK93] [FG98] [JL97] [WG97] [SBH*01] [Hul92] [Gel01] [vW93b] [KM96] [XZC04] [vW93b] [VP04] [LPSW96] [YKP05] [VKP00] [CCK07] [VKP00] [JL97] [LS07] [vW02] [JL97] [UKSE08] [MCG94]
	Unsteady		[GPR*04] [UA01]	[HA04] [CL93] [WE04] [TvW03] [WSE05] [MB95] [HE06] [Sun03] [WEE03] [WSE07] [WSE05]	[BLM95] [MBC93] [STWE07] [ABCO*01] [CL93] [STW*08] [DGH03] [GKT*08] [vFWTS08]

Table 1: *An overview and classification of higher dimensional flow visualization. The survey is grouped based on the classification of flow visualization techniques and the dimensionality of the flow data domain. Each group is then subdivided by steady or unsteady flow solutions. The entries of each sub-group are placed in chronological order. References in sub-script are 2D predecessors*

1.2. Classification

Vector field visualization techniques can be classified into four general categories: direct, geometric, texture-based and feature-based ([PVH*02] [PVH*03] [LHD*04]). We give brief descriptions of these four categories:

- *Direct:* Direct techniques are the simplest solutions to intuitively visualize flow fields. Examples include placing arrow glyphs at each sample point to depict the underlying vector field or mapping color to velocity magnitude. Direct techniques are able to make flow visualization universally and intuitively understandable with less computational cost. However, direct techniques can suffer from a lack of visual coherence. When applied to 3D flow data, they generally suffer from visual complexity and occlusion.
- *Feature-based:* these techniques extract subsets of data with features deemed interesting by users. This process is performed before visualization, hence the visualization is then based on these extracted subsets rather than the whole dataset. This may make the visualization more efficient and effective. However, the complexity and computational cost of feature extraction may be high.
- *Texture-based:* A known texture is distorted according to the local properties of the vector field and then rendered to visualize the vector field. This category of techniques provides a dense and coherent visualization with lots of detail, even in areas of complicated flow. Although texture-based algorithms can provide an efficient visualization of the flow, they can also suffer from visual complexity and occlusion when implemented for 3D flow data.
- *Geometric:* In order to gain a coherent representation of the vector field, integration-based geometric techniques are applied. A typical example involves defining a set of seeding points, then computing trajectories (e.g. streamlines) from these points through the flow, and finally, rendering resulting geometric objects from these trajectories. However, visual clutter and occlusion can be induced if a poor seeding strategy is employed.

1.3. Overview

We organize the literature of this review based on four categories, which can be seen in Table 1. The literature is subdivided into direct, geometric, texture-based and vector field clustering visualization according to the criteria of four general categories described above. The spatial dimensionality of the data domain is applied to classify literature at the second level. Since we focus on higher spatial dimensional domain, only 2.5D and 3D are taken into account. It is worth pointing out that we concentrate on vector field clustering visualization rather than the whole feature-based domain.

Temporal dimensionality is also used to classify literature. All literature in each sub-category is presented in chronological order. Related previous work in 2D is indicated by sub-scripts.

The contributions of this paper are:

- A novel overview of the most recent developments in vector field visualization in higher spatial dimensions.
- The first survey which focuses on flow visualization in higher dimensions (2.5D, 3D and 4D).
- The first survey to include vector-field clustering approaches for vector field visualization.
- Our table-based overview highlights both mature areas and immature areas in vector field visualization.

We point out that this survey is not simply a list of research papers. Related research is compared. Relative advantages and disadvantages are discussed. If a higher dimensional algorithm builds on a 2D algorithm, this is indicated using a reference in subscripts in Table 1.

2. Direct Vector Field Visualization

We start by describing direct vector field visualization methods according to the dimensionality of vector fields.

2.1. Direct Visualization of 2.5D Vector Fields

The following method(s) focus on surface-based (2.5D) vector fields.

Vector Glyphs for Surfaces: A Fast and Simple Glyph Placement Algorithm for Adaptive Resolution Meshes Peng and Laramee [PL08] present a fast and simple image-based glyph placement algorithm in order to investigate and visualize the vector fields on unstructured, adaptive resolution boundary meshes from CFD. Conceptually, this placement algorithm is an extension of the 2D method of Laramee [Lar03] to surfaces.

In the first stage, the vector field is projected onto the image plane. A color-coded velocity image is rendered to encode the vector field. Secondly, a user-defined, evenly-spaced resampling grid is used to define the spatial resolution of the glyph placement. A glyph is generated at the center of each cell. Then various representations of the flow are rendered by reconstruction using filters, optimized for either speed or accuracy. Geometric discontinuities are handled using edge detection in the depth buffer. The next step is to reconstruct each glyph according the resampled vector field value of each resampling cell and project each glyph onto image plane. Finally, all optional enhancements such as a shaded image of the geometry are rendered. In addition to other user interactions such as zooming, translating, and rotating, a multi-resolution technique is applied to highlight details in areas deemed interesting by the user. See Figure 1.

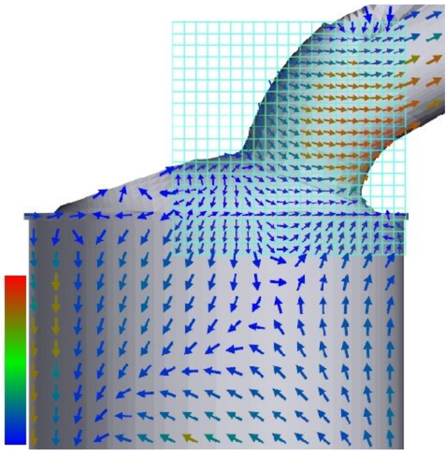

Figure 1: *A multi-resolution visualization of low resolution and high resolution glyphs used to visualize the flow at the surface of a gas engine simulation. [PL08]*

2.2. Direct Visualization of 3D Vector Fields

The following methods focus on 3D vector fields.

Vector Plots for Irregular Grids Dovery [Dov95] extends Crawfis and Max's method [CM92] from regular to curvilinear and unstructured grids. In order to visualize vector fields on unstructured grids, physical space and parameter space resampling methods are employed. During the physical space resampling, the vector field is linearly interpolated at each sample point, then the physical coordinates of the point are calculated, and lastly related oriented glyphs (plots) are projected from back to front. Although this ensures that sample points are uniformly distributed, physical space resampling is computationally expensive. To address this problem, it can be preferable to resample to parameter space instead. At first, random points are directly generated in parametric space with an area-weighted distribution. Then a relatively accurate and dense resampling can be approximated by mapping the parameterized coordinate to physical coordinate grid points. Vector field visualization on arbitrary 3D surfaces can be efficiently achieved with parameter space resampling.

3. Vector Field Clustering and Visualization

Although vector field clustering techniques are arguably feature-based, we include them here because (1) they have never been included in a vector field survey before and (2) the results resemble both direct and sometimes geometric vector field visualization techniques. Thus they can serve as a bridge between these two categories.

Simplified Representation of Vector Fields. Telea et al. [TvW99] present a hierarchical clustering based method that automatically places a limited number of glyphs in a suggestive manner to represent the vector field. It's the first algorithm of its kind and produces both global and detailed information in the same image.

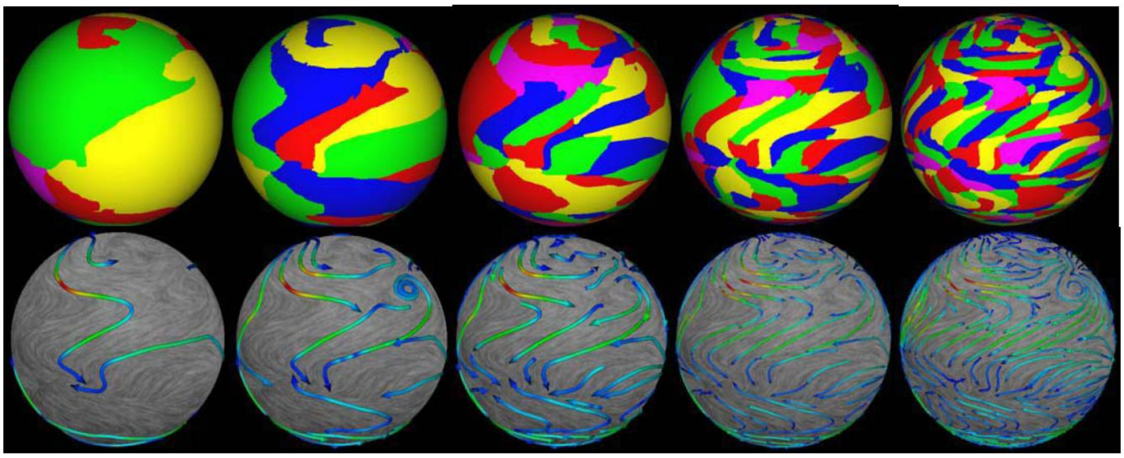

Figure 2: *Climate dataset decomposition, five coarsest levels (left to right). Cluster images (top row) and flow texture applied with streamline arrow icons (bottom row). Image Courtesy of Michael Griebel [GPR*04]*

In terms of the clustering algorithm, two candidate clusters are selected and merged together in a bottom-up fashion. During this process, a vector field similarity measure is used to evaluate which clusters should be merged. An error metric based on local vector magnitude and direction is introduced to define how a new cluster is merged from two existing ones. Although this algorithm automatically displays the simplified flow, a few clustering parameters are available to refine the result and satisfy different needs of the users, i.e. changing the detail level of the clustering.

Construction of Vector Field Hierarchies. Heckel et al. [HWHJ99] present a method to visualize discrete vector fields in a hierarchical fashion. This method uses a clustering approach to segment the original vector field into a series of disjoint clusters. The algorithm is applied in a top-down fashion. Firstly, points from the original vector field data are treated as a single cluster. Then a procedure for splitting clusters is applied using a weighted best-fit plane which partitions the space into convex regions (sub-clusters). Each region has an error measure which calculates the differences between the original discrete vector field and the simplified vector field. With the use of the error measure approach, the recursive procedure of splitting clusters can be terminated when a user-defined threshold value is met. It's also worthy of note that this method does not require a regular grid.

A Phase Field Model for Continuous Clustering on Vector Fields. Garcke et al. [GPR*01] present a new multiscale method for vector field clustering in 2.5D and 3D space. This continuous clustering method is inspired by the well-known physical phase separation clustering model - the Cahn Hilliard model [CH58].

To classify and enhance the correlation in the cluster sets effectively, this new phase-separation-based continuous clustering method formulates the clustering problem as a diffusion problem rather than a merging or a splitting problem. At the first stage, the set of clusters is implicitly handled by an evolution function with two important energy contributions taken into account: the nucleation of cluster sets and a successive coarsening of the clusters. In accordance with the underlying physical data and based on the evolution function, segments of the flow fields are extracted and classified depending on their location and orientation. Afterwards, a skeletonization approach is applied to highlight the essential features of the refined cluster sets. Finally, various geometric representations are adopted to render the highlighted skeleton in a intuitive way.

Flow Field Clustering via Algebraic Multigrid. Griebel et al. [GPR*04] present a vector field clustering method based on an algebraic multigrid [UA01]. Each sample in the flow field is represented by a tensor stiffness matrix that encodes the local properties of the flow field. The algebraic multigrid technique operates on these tensor matrices in order to construct a vector field hierarchy encoding the flow structure. The method is geometry-free and is demonstrated on 2D and 3D vector fields. See Figure 2.

Survey of Clustering Algorithms. Xu and Wunsch II [XW05] present a comprehensive and systematic survey focused on scalar clustering algorithms rooted in statistics, computer science, and machine learning. Firstly, they generalize the clustering analysis procedure using four steps: feature selection or extraction, clustering algorithm design or selection, cluster validation, and result interpretation. Secondly, they detail clustering algorithms in terms of the nature of clusters. Corresponding approaches for clustering different kinds of data sets are discussed and suggested. Lastly, in order to compare different clustering algorithms, applications to some benchmark data sets are presented.

Clustering algorithms can be divided in to two groups: either based on constructing a hierarchical structure or based on generating substes of similar data (known as parlilioning). Hierarchical clustering algorithms are mainly classified as agglomerative methods and divisive. Since the agglomerative methods can generate flexible clustering using binary trees and provide very informative descriptions, they are employed to develop some vector field clustering visualizations like [TvW99] and [HWHJ99]. However, the computational cost for hierarchical clustering can be expensive (e.g. $O(N^2)$) so some of the early hierarchical clustering methods are not capable of handling large-scale data sets. In order to address this problem, some improved hierarchical clustering methods are introduced like BIRCH [ZRL96] which utilizes the clustering feature tree to improve the robustness and reduce the computational complexity to $O(N)$. In terms of pre-specified numbered clustering methods, the K-means algorithm is representative and is effective in clustering large-scale data sets. A general solution to data clustering is a major aim of the AI community.

4. Geometric Vector Field Visualization

In this section, we describe geometric flow visualization methods.

4.1. 2.5D Vector Fields

The following methods focus on surface-based (2.5D) vector fields.

4.1.1. Steady State Vector Fields

Geometric flow visualization methods for surface-based (2.5D) steady-state vector fields are presented.

Flow Visualization with Surface Particles. Van Wijk [SvW91] extends his previous work with surface particles [vW93a]. The basic geometry of stream-surfaces and stream-ribbons is enhanced by adding massless particles with a small finite size (facets) to the surface and visualizing their behavior. Several features are shown including: an improved shading model used to decrease jagged edges and strobing artifacts, implementation of Gaussian filters to avoid spatial and temporal artifacts, and a scan-conversion algorithm to improve calculation efficiency. With this improved rendering method, high-quality images and animations are feasible at a reasonable computational cost.

Image-Guided Streamline Placement on Curvilinear Grid Surfaces. In order to generate evenly distributed streamlines on 3D curvilinear surfaces, Mao et al. present an image-guided streamline placement technique [MHHI98]. Essentially, this approach extends Turk and Banks' 2D image-guided streamline placement method [TB96] to 3D curvilinear grid surfaces.

First, vectors on the 3D curvilinear surface are mapped into computational space. Second, streamlines are generated with desired density distribution by using the extended Turk and Banks' 2D algorithm. And last, streamlines generated in the computational space are mapped back to the 3D surface. Additionally, one of the most interesting parts of this paper is the new energy function which is applied with the poisson ellipse sampling technique. With help of this energy function, the density of streamlines in the computational space is locally adapted according to the physical space grid density in order to address the mapping distortion caused by non-uniform grid density in the curvilinear grid.

Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach. Spencer et al. present a novel, automatic streamline algorithm to visualize the vector field on surfaces in 3D space [SLCZ09]. This paper describes an image-based algorithm to generate evenly-spaced streamlines fast, simply, and efficiently for any general surface-based vector field. Conceptually, this algorithm can be viewed as an extension of Jobard and Lefer's 2D method [JL97] to 2.5D.

In the first stage, the vector field is projected onto the image plane. Based on this operation streamlines can be seeded and integrated. The next step divides the image with a user-defined grid and sequentially attempts to seed a streamline at the center of each cell. If the z-depth of the seed is non-zero and no streamlines lie closer than d_{sep} to the seed point. A new streamline is traced. A vector field-based seeding method is also used - whenever a seed point along the length of the grid-based streamline curve at regular interval and no streamlines lie closer than d_{sep} to the seed point, a new streamline is traced from it and pushed onto a queue. This process repeats until the queue is empty. The algorithm is enhanced by a proximity testing method which decides when to terminate the particle tracer and an edge detection which ensures that the streamline is seeded in a continuous, bounded area. See Figure 3 for more detail.

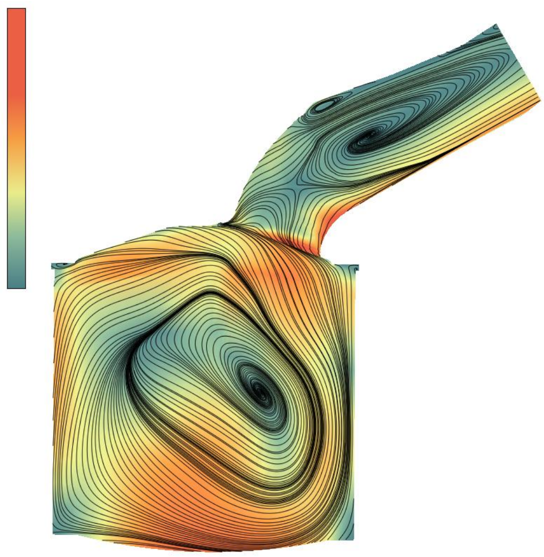

Figure 3: *Visualization of flow through the gas engine simulation. By setting d_{test} to $0.05 \times d_{sep}$, streamlines overlay highlighting loops within the flow field. [SLCZ09]*

4.2. Geometric Visualization of 3D Vector Fields

The following methods focus on 3D vector fields.

4.2.1. Steady Vector Fields

Geometric flow visualization methods for surface-based (3D) steady-state vector fields are described in this section.

The Stream Polygon: A Technique for 3D Vector Field Visualization. In order to display local deformation including normal and shear strain and rigid body rotation, Schroeder et al. [SVL91] present a 3D vector visualization technique - the *stream polygon* which is an extension of stream lines. Stream polygons are actually a set of regular, n-sided polygons perpendicular to the local vector. By tracing stream polygons along streamlines , stream tubes are generated to depict properties of the vector field including strain, translation and rotation by the corresponding shapes and radius of polygons.

Constructing Stream Surfaces in Steady 3D Vector Fields. Hultquist [Hul92] introduces a novel method to construct *stream surfaces* efficiently. Stream surfaces offer perceptual benefits over seeding many streamlines. Depth perception is enhanced using surface primitives. Visual complexity is also reduced. In this implementation, flow fields can be explored rapidly and effectively from improved sampling densities over the tangent surface, see Figure 5.

During stream surface construction, advancement of the stream surface front is controlled by adaptively adjusting the sampling density, so regions of high divergence can be sampled accurately. Surfaces can be constructed using a greedy triangular tiling between adjacent pairs of streamlines. Ribbon-splitting can be introduced to adaptively refine the polygonal approximation in this process - stream surfaces can be split apart by adding some seeding points into the advancing front of sparsely sampled regions in order to visualize highly divergent flow fields with an acceptable sampling density. On the other hand, ribbons can be merged in areas of convergent flow.

Flow Volumes for Interactive Vector Field Visualization. Max et al. [MBC93] introduce flow volumes, the 3D equivalent of streamlines, to visualize more information about the vector field by intuitive volumn density which 2D streamlines or ribbons can't offer. In order to render the flow volumes in an efficient way, the method of Shirley and Tuchmann [ST90] is adopted to volume render the flow using a set of semitransparent tetrahedra. These tetrahedra triangulate in intermediate layers between the layers of vertices in successive time steps. Additionally, a curvature based adaptive subdivision method is applied to handle the sampling fragment if the flow diverges or converges too much. Since only a small volume of the smoke is rendered, flow volumes are fast enough to make user interaction possible.

Implicit Stream Surfaces. In order to obtain an efficient method for the automatic placement of initial curves, van Wijk presents a new method for construction of stream surfaces [vW93b]. The main concept of his method is based on modeling a stream surface as an implicit surface $f(x) = C$. Streamlines are traced from every sample point in the vector field until they hit a boundary. Then a scalar field is derived such that all of the points common to a streamline are constant. Stream-surfaces are then computed using standard iso-surfacing techniques on the scalar field.

Streamball Techniques for Flow Visualization. Brill et al. [BHR*94] introduce a 3D flow visualization technique using objects called streamballs which are based on the metaballs of Wyvill et al. [WMW86]. Spheres split or merge automatically depending on the distance between their center points in order to visualize the divergence or convergence of arbitrary complex flow fields.

The premise for visualizing flow data with streamballs is to use the center positions of particles in the flow as skeletons for the construction of implicit surfaces. Spherical surfaces can then be blended with each other to form three-dimensional streamlines and stream surfaces. There are two kinds of streamballs. Using each of the discrete particles as a single skeleton leads to producing discrete streamballs. Grouping several discrete particles together produces a continuous skeleton (i.e. lines, curves etc.). Both forms can be enhanced by radius-mapping and color mapping to visualize local field parameters.

UFLOW: Visualizing Uncertainty in Fluid Flow. In order to visualize uncertainty resulting from the application of different numerical algorithms for particle tracing in a fluid flow, Lodha et al. [LPSW96] present a system named UFLOW (Uncertainty Flow) to visualize and investigate this uncertainty in fluid flow. This paper actually is a practical step to integrate presentation of data with uncertainty in scientific visualization [BBC91] [TK93].

First, the uncertainty in fluid flow is characterized by the possible sources of uncertainty. In this paper, focus is put on the use of different numerical algorithms to trace streamlines in fluid flow. Then with several visualization techniques including uncertainty glyphs, envelopes animations, priority sequencing, trace viewpoints and rakes applied, this system can intuitively highlight these uncertainties from application of different integration methods for users.

Real-Time Techniques For 3D Flow Visualization. Fuhrmann and Gröller [FG98] present a new method to effectively visualize 3D steady flow fields in order to address the problem of the occlusion and perception of distant details in 3D flow fields. Conceptually, this method draws upon the evenly-spaced streamline placement algorithm [JL97] and a texture based technique - FROLIC [WG97].

In order to solve the problem of preserving distance details when visualizing vector fields in a 3D domain, dashtubes - the animated, opacity-mapped streamlines, are automatically placed evenly in 3D space. A texture mapping technique is then applied to maintain the detail along the streamline at a constant level although the velocity of flow changes. Additionally, magic lenses and magic boxes are implemented as interactive tools for users to visualize detailed features of areas deemed interesting by users.

A Tetrahedral-based Stream Surface Algorithm. Scheuermann et al. [SBH*01] present an algorithm to precisely generate the stream surfaces in tetrahedral grids by exploiting the piecewise linear interpolation. This algorithm is closely related to Hultquist's parametric approach [Hul92]. The stream surface in each cell can be precisely described by using piecewise linear interpolation, since the whole surface is built upon the cellwise analytic description which enables easy access to the local error of stream surfaces and related information of the flow structure in the cells. By using this algorithm, the stream surfaces that approach vortices which can not be properly calculated by previous approaches can now be precisely described.

Stream Surface Generation for Fluid Flow Solutions on Curvilinear Grids. Gelder presents a method to generate stream surfaces in a semi-global fashion for 3D vector fields on a curvilinear grid [Gel01]. This method can solve constraints over a large region of space by integrals instead of locally propagating the solution of a differential equation. The basic idea of this method is similar to the method from van Wijk [vW93b] and the method from Knight and Mallinson [KM96], however, this presented method can address problems of no-slip surfaces and grids with degeneracies which the previous methods can not handle properly.

Rendering Implicit Flow Volumes. Two different approaches are presented to better model and render the implicit flow volumes by Xue et al. [XZC04], a slice-based 3D texture mapping and an interval volume segmentation coupled with a tetrahedron projection-based renderer. Conceptually, the implicit flow volumes are the extension of implicit stream surfaces proposed by van Wijk [vW93b].

In the first method, the implicit flow field is rendered directly without the inflow mapping to a scalar field using a dynamic texture operation, thus high interactivity can be obtained. In order to visualize the flow on stream surfaces and time surfaces, the second approach extracts a geometric flow volume from the implicit flow using an iso-contouring or an interval volume routine, hence accurate and detailed result can be gained.

Comparative Flow Visualization. In order to address the need for comparative visualization tools to help analyze the differences on visualizing flow or vector data sets, Verman and Pang [VP04] present a method to compare individual streamlines and streamribbons as well as a dense field of streamlines. Since this method is an extension of UFLOW [LPSW96], it can visualize the difference between various integration methods of streamlines from the same data set and also the difference in streamlines generated

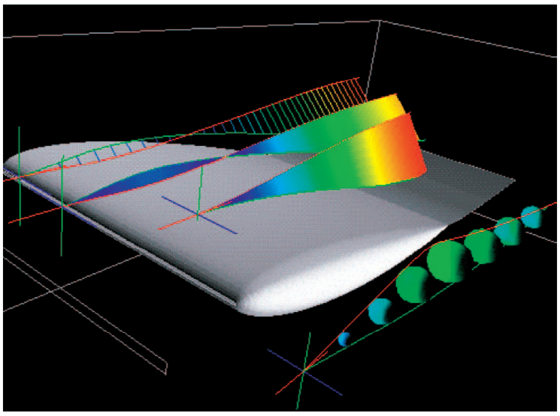

Figure 4: *Comparing streamlines of two datasets simulated using different turbulence models. The streamlines are compared using line glyphs, strip envelopes, and sphere glyphs to highlight the differences between them. Image Courtesy of Alex Pang [VP04]*

from the same seed point location using different integrators in pairwise fashion. Furthermore, it's able to compare the differences between dense fields of streamlines in order to present a intuitive global view of the differences between vector fields. Additionally, the differences of streamribbons can be visualized by using an overlay or a novel simplified representation according to users' different needs. See Figure 4 for an example.

Strategy for Seeding 3D Streamlines. Ye et al. [YKP05] present a strategy for streamlines placement in 3D flow fields. With this algorithm, intuitive visualization can be obtained from essential flow patterns captured in sufficient coverage of the field after reducing clutter. This approach is a 3D extension of the 2D flow guided approach [VKP00].

Firstly, critical points need to be extracted in order to distinguish the interesting areas with important flow patterns. Later, different seeding templates are applied around the vicinity of these critical points. Due to the variability of the flow pattern, these seeding templates can transform based on how far one critical point is to another. A unique template-based hybrid map is used to handle this transformation. Then, Poisson seeding is introduced to add streamlines into regions which have sparse coverage. Finally, in order to refine streamlines in 3D space, a filter is applied based on streamlines' geometric and spatial properties to filter out the streamlines which are too short, with small winding angles, similar to each other, or with very high winding angles.

Similarity-Guided Streamline Placement with Error Evaluation. Chen et al. [CCK07] present an adaptive method for streamline placement on steady vector fields in 2D and 3D. In order to achieve a good balance between feature-based and density-based streamline placement, this new approach draws upon both feature-based streamline placement methods like [VKP00] [YKP05], and density-based ones like [JL97] [MT*03] [LS07].

A similarity metric is applied to measure the similarity among streamlines. This metric consists of Euclidean distance and similarity of shape and direction. Guided by this metric, streamlines are traced to accentuate regions of geometric interest rather than explicitly enumerating the whole vector field. Additionally, an error metric for streamline representations is introduced to evaluate the generated result when compared to the original vector field. With this metric, comparison and evaluation of various streamline representations can be easiy presented in a quantitative way.

Image-Based Streamline Generation and Rendering. In order to better display 3D streamlines and reduce visual cluttering in result images, Li et al. [LS07] present an image-based method for streamline seeding and generation. This is the first algorithm of its kind to use an image based approach for the seeding of 3D streamlines.

First, 2D depth maps are generated. After, seeds are selected based on the 2D depth maps and then unprojected back to 3D object space before evenly-distributed streamline integrations are generated. By carefully separating streamlines in image space, visual cluttering after the projection from 3D to 2D image can be effectively reduced. Visual clarity can be also enhanced by controlling the density and rendering styles of streamlines according to different user criteria and needs. Additionally, this method provides level of detail rendering, depth peeling, and stylized rendering of streamlines to gain better perception of 3D flow fields.

GPU-Based Streamlines for Surface-Guided 3D Flow Visualization. In order to visualize 3D flow in the vicinity of boundary and feature surfaces without losing 3D information due to the inherent projection, Üffinger et al. present a combination of 2D dense texture-based and 3D streamlines visualization [UKSE08]. Conceptually, the idea of this implementation is based on techniques proposed by Max et al. [MCG94].

First, an image-based seeding strategy is applied to obtain an importance-driven distribution of streamline seed points on the surface in object space. Then, a new algorithm for generating streamlines using geometry shaders on the GPU is employed to efficiently trace streamlines based on seed points in object space, thus interactive exploration is enabled. The user can relocate seed points interactively to emphasize regions of interest.

4.2.2. Geometric Visualization of 3D Unsteady Vector Fields

Geometric visualization methods for 3D time-dependent vector fields are outlined in this section.

Unsteady Flow Volumes. Becker et al. [BLM95] present a method to generate the 3D analog of streaklines based on time-dependent flow fields. Conceptually, this method is an extension of 3D static flow based flow volumes [MBC93] to visualize unsteady flows in 3D space.

Similar to the steady flow volumes which consists of the collection of streamlines that are seeded from on a 2D generating polygon, flow volumes for unsteady flows are based on streaklines which also start on a generating polygon. A series of tetrahedra is constructed along the path of a streakline. Although they are based on static flow volumes, unsteady flow volumes have to take additional considerations into account in order to correctly depict unsteady flow. Firstly, the adaptive step-size integration presented by Lane [Lan94] is used to advect the vertices which are generated by the streakline integration at the given time steps. Additionally, in order to address the difficulty in extending the original subdivision scheme of Max et al. [MBC93] to unsteady flow, adaptive subdivision is applied to insert or delete particles corresponding to the diverging or converging flow. The adaptive subdivision in the downstream direction is also required to keep the distance between layers of particles roughly constant.

Point-based Stream Surfaces and Path Surfaces. For the purpose of creating and rendering stream surfaces and path surfaces which enable users to interactively manipulate seed curves, even for unsteady flow, Schafhitzel et al. [STWE07] present a point-based algorithm by exploiting GPU-based programming. This algorithm is based on point set surfaces [ABCO*01] and adopts line integral convolution [CL93] to incorporate texture-based flow visualization.

First, the streamlines and pathlines are generated by a GPU-based particle tracing algorithm. In order to deal with local flow divergence, a particle-density criterion is applied to obtain an even distribution of those particles. Second, based on the particle traces, the corresponding surfaces are generated and displayed by point set surfaces. Third, in order to compute corresponding flow textures for the texture-based flow visualization, a LIC method is applied on the vector field which is stored by the previous particle integration. Since the GPU-based implementation is efficient, user interaction for visualization and exploration on stream surfaces and path surfaces, and even for unsteady vector fields, is possible.

Path Line Attributes - an Information Visualization Approach to Analyzing the Dynamic Behavior of 3D Time-Dependent Flow Fields. In order to visually analyze the dynamic behavior of the path lines on 3D time-dependent flow fields, Shi et al. [STW*08] present an information visualization approach which enables the user to intuitively explore intricate 4D flow structures. Although SimVis [DGH03] is the most similar application, this approach puts focus on dynamic flow data rather than scalar data.

First, local and global properties of path lines are obtained at selected seeding positions in 4D space. Then, for the purpose of analyzing the multivariate data set from previous step, interactive brushing and focus+context visualiza-

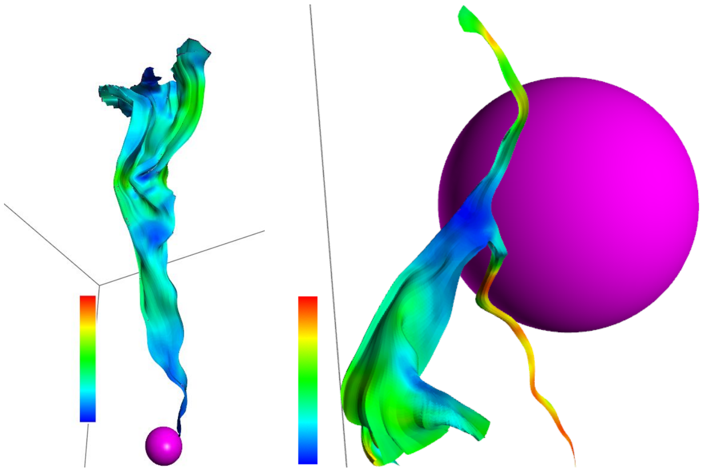

Figure 5: *Streamsurface visualization of the smoke (left) leaving from a sphere (right) and spliting when it intersects the sphere. [MLZ09]*

tion are utilized to illustrate how path lines can be used to describe and visualize the underlying flows.

Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields. A novel algorithm for the computation of integral surfaces is presented by Garth et al. [GKT*08] that generates accurate integral surfaces from unsteady vector fields. As opposed to previous work on surface computation techniques like the method proposed by Hultquist [Hul92], this approach introduces a novel separation of integral surface approximation from the generation of a graphical representation in order to address limitations of previous work. According to this decoupling, the first step approximates a series of timelines using iterative refinement and generates an integral surface skeleton. The second step computes a well-conditioned triangulated representation based on the skeleton. This method can be applied on large time-varying datasets.

Smoke Surfaces: An Interactive Flow Visualization Technique Inspired by Real-World Flow Experiments. Von Funck et al. [vFWTS08] present a new approach to visualize smoke surfaces by utilizing semi-transparent streak surfaces. This is the first time that semi-transparent streak surfaces are adopted to interactively visualize time-dependent flow fields.

To avoid the problem of expensive adaptive remeshing which prevented streak surfaces from being used in an interactive fashion, the method of coupling the opacity of the triangles to their area, shapes, and curvatures is applied to render smoke using a triangular mesh with a fixed topology and connectivity. Thus, an intuitive and interactive exploration is possible.

5. Texture-Based Vector Field Visualization

In this section, we survey texture-based visualization methods for vector fields in higher dimensions.

5.1. 2.5D Vector Fields

The following methods focus on surface-based (2.5D) vector fields.

5.1.1. Steady Vector Fields

Texture-Based visualization methods for surface-based (2.5D) time-independent vector fields are outlined first.

Enhanced Spot Noise for Vector Field Visualization. Leeuw and van Wijk [LvW95] present an enhanced spot noise method for surface vector field visualization as an extension of the original spot noise described by van Wijk [vW91]. This enhanced method offers several improvements over the original. To obtain a better visualization of vector fields with high curvature, spot blending is applied to deform the spot into a curved shape according to the local velocity field characteristics. The deformation of the spot is carried out based on a stream surface. Second, with spot filtering implemented, the problem of generating spot noise pictures with a coarse, low-frequency component can be solved, and more homogeneous textures are computed. Third, the spot noise images are rendered fast by graphics hardware, thus the visualization can be rendered at interactive speeds. This method is extended to visualize flow on curvilinear grids.

Visual Simulation of Experimental Oil-Flow Visualization by Spot Noise Images from Numerical Flow Simulation. Leeuw et al. [LPPW95] present a visual simulation of the oil-flow experimental patterns using an enhanced spot noise texture. This work is based on the enhanced spot noise method [LvW95].

Experimental flow visualization techniques visualize flow field on the surface with oil streaks. The numerical flow simulation can be used to obtain the wall-friction-vector data based upon which a spot noise texture is generated and blended to enhance the visualization of the flow field on the

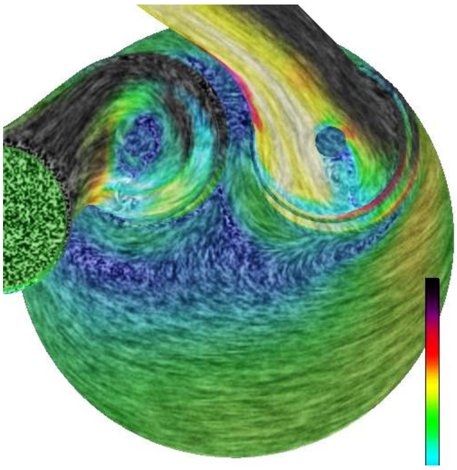

Figure 6: *Texture-based flow visualization is applied to the surface of an 221K polygonal intake port mesh. [LJH03]*

surface. Convergence data is used to scale the intensity distribution function and spot advection is used to simulate oil accumulation in high convergence areas, the final spot noise visualization renders images very comparable to the numerical and experimental visualizations.

Fast Line Integral Convolution for Arbitrary Surfaces in 3D. Battke et al. [BSH97] introduce a enhanced line integral convolution method for visualizing surface-based vector fields. This method is an extension of the fast LIC approach described in [SH95] and can be applied to 3D arbitrary surfaces whereas previous approaches are restricted to the characteristics of curvilinear surfaces. This method starts with a triangular approximation of the arbitrary surface, then for each triangle a local LIC texture is computed based on local euclidean coordinates, the final step is to generate a smooth textured surface by following stream lines across neighboring triangles.

5.1.2. Texture-Based Visualization for 2.5D Unsteady Vector Fields

Texture-Based visualization methods for surface-based (2.5D) time-dependent vector fields are presented here.

Image Space Based Visualization of Unsteady Flow on Surfaces Laramee et al. [LJH03] present a method to generate dense representations of unsteady flow on surfaces based on 2D Lagrangian-Eulerian Advection [JEH01] and Image Based Flow Visualization [vW02]. This paper addresses the problem of high computation time required to visualize unsteady flow on surfaces, especially on boundary surfaces of large and complex meshes from CFD data sets. See Figure 6.

This approach starts with vector field projection which projects the vector field of the surface to the image plane. A color-encoded velocity image which stores the projected vector field is generated. Secondly, the mesh which is applied to advect the textures similar to IBFV [vW02] is distorted according to the discretized Euler approximation of a pathline. Then a noise texture is distorted and attached according ot the distorted mesh. The next step is to inject and blend noise in image space. Geometric discontinuities are taken into account using edge detection. Finally, all optional images such as a shaded version of the geometray are rendered.

Image Based Flow Visualization for Curved Surfaces. Van Wijk [vW03] presents a new method to render dense, vector-field guided textures on 3D curved surfaces. As an extension of the original 2D algorithm - IBFV [vW02], this algorithm synthesizes textures on curved surfaces directly in image space and also inherits advantages of simplicity, speed, and efficiency in 3D.

Each vertex defining the triangular surface contains a position, normal vector and velocity vector. The mesh and its associated vector field is projected to image space. Each vertex position is distorted according to the velocity vector at that position. Advected coordinate positions are computed using an Euler approximation of a pathline. Synthesized noise textures are then mapped to the newly advected coordinate positions. After this, noise blending is used to update the visualization as noise texture is advected in the direction of the flow. Finally, a shaded version of the mesh geometry is rendered to enhance depth perception.

ISA and IBFVS: Image Space Based Visualization of Flow on Surfaces. Laramee et al. [LvWJH04] present a side-by-side comparison between two image space based methods for visualization of vector fields on surfaces: Image Space Advection (ISA) [LJH03] and Image Based Flow Visualization for Curved Surfaces (IBFVS) [vW03]. Firstly, they identify that both algorithms share several overlapping characteristics such as projection, texture-mapping and noise insertion, whereas the main difference between these two methods is the way the textures advect: ISA adopts the image-based mesh however the IBFVS is driven by the original object-based mesh. Furthermore, the relative advantages and disadvantages of each approach are also presented and suggestions are given regarding when and where they are most suitable.

Texture-Based Flow Visualization on Iso-surfaces from Computational Fluid Dynamics. Laramee et al. [LSH04] apply a method to extend dense, texture-based visualizations of flow onto iso-surfaces. In terms of the texture-based visualization techniques, the ISA method [LJH03] is adopted as the solution in this case.

Combined with the iso-surfacing, a normal mask is applied to address the normal component of the flow to the isosurface. This normal mask, stored in the alpha channel, adjusts the opacity of the image overlay according to the magnitude of cross-flow component to the isosurface. In the texture advection phase ISA [LJH03] is applied. The vector field is projected onto image space. Edge detection is incorporated. The image advects according to the advected texture coordinates. Noise is injected and blent, Finally, an image overlay is employed to enhance the result.

Texture Advection on Stream Surfaces: A Novel Hybrid Visualization Applied to CFD Simulation Results. Laramee et al. [LGSH06] present a hybrid visualization which combines the advantages from stream surfaces and texture advection techniques. The stream surface generation is based on the method of Garth et al. [GTS*04] while ISA [LvWJH04] is adopted as the texture advection algorithm. This approach conveys properties of the vector field that stream surfaces alone can not. They apply the visualization technique to various patterns of flow from CFD (important to automotive engine simulation including two patterns of in-cylinder flow (swirl and tumble) as well as flow through a cooling jacket). In addition, they explore multiple vector fields defined at the streamsurface such as flow, vorticity, and pressure gradient.

5.2. Texture-Based Visualization for 3D and 4D (Time-Dependent) Vector Fields

The following methods focus on 3D vector fields.

5.2.1. Steady-state vector Fields

Texture-Based flow visualization methods for 3D steady-state vector fields are described and compared to one another in this subsection.

Interactive Exploration of Volume Line Integral convolution Based on 3D-Texture Mapping. Rezk-Salama et al. [RSHTE99] present a direct volume rendering approach based on 3D-Texture mapping for visualizing vector fields in an interactive and animated fashion. Conceptually, this appraoch is a 3D extension to the original LIC [CL93].

With interactive methods including user-controlled transfer functions and volume clipping approachs applied, interactive exploration of the interior structures of the vector field is realized. In order to improve the perception of flow in 3D at a reasonable computational cost, an animated 3D-LIC method is presented via two approaches. The first one is to pre-compute a special 3D-LIC texture for the time-dependent color table animation. The second one is to clip the 3D-LIC volume interactively according to pre-computed user-defined volumes.

Case Study: Visualizing Ocean Flow Vertical Motions using Lagrangian-Eulerian Time Surfaces. In order to track the evolution of features caused by depth variations of thermoclines, Grant et al. [GEO02] present a new method named Lagrangian-Eulerian Time Surfaces (LETS) to visualize ocean flows including vertical motions in 2D and 3D space. This method is built upon the original 2D Lagrangian-Eulerian Advection (LEA) method proposed by Jobard et al. [JEH01]

Initially, LETS distributes a set of particles uniformly on a horizontal plane. Then a time surface is obtained by tracking the horizontal plane in time using a mixture of Eulerian and Lagrangian techniques during each time step. The vertical motion is taken into account in addition to the horizontal motion to displace the time surface according to the flow. Afterwards, a texture computed using texture advection from the dominant horizontal motion is projected onto the evolving surface to enhance the final visualization. This method extends LEA to handle 3D flows, however, the method only works for the case of weak vertical velocities rather than the general 3D case. Vector fields with vertical velocities comparable to the horizontal velocities will distort time surfaces too much.

5.2.2. Texture-Based Visualization for 3D Unsteady (4D) Vector Fields

Texture-Based vector field visualization methods for 3D time-dependent vector fields are described.

Visualization of Vector Fields Using Seed LIC and Volume Rendering. Helgeland and Andreassen [HA04] present a new method to more effectively visualize three-dimensional vector fields using LIC. Conceptually, this method is a extension of the original LIC method [CL93].As the first stage, a seeding-based LIC is employed to fast compute 3D textures using an appropriate input texture in a interactive fashion. The direction of a vector field can be effectively observed from generated images. Secondly, in order to reveal depth relations among the field lines traced by Seed LIC at a reasonable computational cost, an appropriate transfer function based on a limb darkening shading technique is applied. Finally, the combination of texture-based direct volume rendering and volume LIC is used to depict additional information of scalar quantities when visualizing 3D vector fields.

GPU-Baed 3D Texture Advection for the Visualization of Unsteady Flow Fields Weiskopf and Ertl present an interactive texture-based approach for the dense visualization on unsteady 3D flow fields [WE04]. Although the basic idea of this approach is from 3D IBVF [TvW03], this approach improves and extends 3D IBVF.

The first contribution of this paper is that this method can display a full range of velocity values with a fully tree-dimensional advection mechanism, thus the problem of velocities being restricted in 3D IBVF is addressed. Second, this GPU-based technique allows a slice of the 3D representation to be updated in a single rendering pass, which makes the interaction possible. Third, an enhanced blending scheme is applied with more flexible noise and dye injection. Moreover, the advection and rendering schemes are extended to transport and display different materials in order to gain more insight of flow fields.

Real-Time Advection and Volumetric Illumination for

the Visualization of 3D Unsteady Flow. With the goal of high efficiency and good visual perception, Weiskopf et al. [WSE05] present a dense texture-based technique to interactively visualize the unsteady 3D flow. Essentially, this technique is a 3D extension to the texture advection [MB95] by exploiting GPU-based programming.

First, a novel 3D GPU-based texture advection mechanism is applied, where the problem of 3D logical memory structure is implementated fast and efficiently by using 2D textures in 2D physical memory. Second, slice-based direct volume rendering is applied to compute volume illumination. Third, some perception-guided volume shading methods are introduced, such as halos with a volumetric importance detector in order to gain good visual perception.

High-Quality and Interactive Animations of 3D Time-Varying Vector Fields. In order to address perceptual problems arising from dense presentation like cluttering, Helgeland et al. [HE06] present a interactive texture-based method for visualizing 3D unsteady flow fields based on a sparse representation.

At the first stage, a set of particles are evenly distributed throughout the whole domain and then tracked along the time-dependent velocity field by calculating pathlines. Then a novel particle advection strategy inspired by the evenly-spaced streamlines in 2D [JL97] is applied in order to maintain the coherent particle density as time increases. At each time step, directional information is visualized in the output 3D texture. In this way, animation shows the advection of particles, while each frame shows the instantaneous vector field. Conceptually, idea of this hybrid solution is similar to the ones used in DLIC [Sun03] and UFAC [WEE03]. Additionally, by decoupling the rendering stage from the rest of the visualization pipeline (based on the method proposed by Li et al. [LBS03]), rendering performance is improved and interactive exploration of multiple fields is also possible.

Texture-Based Visualization of Unsteady 3D Flow by Real-Time Advection and Volumetric Illumination. Taking computational efficiency and visual perception into account, Weiskopf et al. [WSE07] present a technique to interactively visualize the unsteady 3D flow in a dense texture-based fasion. This technique is conceptually built upon their previous texture advection based work [WSE05].

Common to previous work, a 3D GPU-based texture advection mechanism and slice-based volume rendering are applied to achieve efficient 3D texture advection. Additionally, in order to incorporate volumetric illumination, two alternative methods are introduced: first, gradient-based illumination that makes use of GPU-based real-time computation of gradients for the sake of local illumination, and second, line-based illumination which adopts the idea of illuminated streamlines based on vector field directions [ZSH96]. Finally, the problems of clutter and occlusion are addressed by applying perception-guided rendering methods and the volumetric importance function.

6. Conclusion and Future Work

This paper provides the most up-to-date overview of the state-of-the-art in vector field visualization in higher dimensions. The most important literature is included and discussed in this survey. There is no single solution to visualize all kinds flow data or to handle all kinds of visualization problems. Hence the existence of various vector field visualization techniques provides users different choices to gain an intuitive exploration or a high quality presentation according to their different interests.

As clearly illustrated from Table 1, the group of geometric approaches is the most mature since a great amount of effort has been invested in this area. Reasons for the success of this group have to do with its coherent and insightful presentation which can help users to gain an intuitive picture of on the data. Major focus has been put on geometric methods for 3D steady flow fields whereas the cases of surface-based (2.5D) or 3D unsteady flow fields have received comparatively less attention. Table 1 highlights the absence of work that has been done using geometric techniques for unsteady flow on surfaces. The texture-based group is also a mature group which contributes solutions to visualize 2.5D or 3D flow fields densely and coherently. However, it still can suffer from visual complexity when implemented for 3D and 4D flow data.

On the other hand, direct approaches and vector-field clustering approaches have received relatively little attention. As we can see from Table 1, there was no direct method for surface-based flow field visualization until 2008. This is probably due to the difficulties in placing the glyphs on the complex boundary meshes and perceptual problems like visual complexity and occlusion. With respect to vector-field clustering methods, although some methods have been presented to visualize 3D vector fields, there are still few methods for visualization of 2.5D vector field because of the potential challenges added by clustering vector field on large, unstructured boundary surfaces. However, the size of the input data set continues to grow very fast. For this reason, simplification approaches are necessary, such as vector-field clustering.

In future work, there are some areas where extra work is needed: (a) visualizing time-dependent flow datasets, (b) addressing visual complexity and occlusion , (c) handling inaccuracy and uncertainty of visualization, (d) and most importantly, automatic or semi-auto selection and simplification approaches for visualization.

7. Acknowledgements

This work was supported by EPSRC research grant EP/F002335/1. The authors would like to thank Edward Grundy for his help in proofreading the manuscript.

References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point Set Surfaces. In *VIS '01: Proceedings of the conference on Visualization '01* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 21–28.

[BBC91] BEARD M., BUTTENFIELD B., CLAPHAM S.: *NCGIA Research Initiative 7: Visualization of Spatial Data Quality*. Technical paper 91-26, National Center for Geographic Information and Analysis, October 1991.

[BHR*94] BRILL M., HAGEN H., RODRIAN H.-C., DJATSCHIN W., KLIMENKO S. V.: Streamball Techniques for Flow Visualization. In *Proceedings IEEE Visualization '94* (Oct. 1994), pp. 225–231.

[BLM95] BECKER B. G., LANE D. A., MAX N. L.: Unsteady Flow Volumes. In *Proceedings IEEE Visualization '95* (1995).

[BSH97] BATTKE H., STALLING D., HEGE H.: Fast Line Integral Convolution for Arbitrary Surfaces in 3D. In *Visualization and Mathematics* (1997), Springer-Verlag, pp. 181–195.

[CCK07] CHEN Y., COHEN J., KROLIK J.: Similarity-Guided Streamline Placement with Error Evaluation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1448–1455.

[CH58] CAHN J., HILLIARD J.: *Free Energy of a Non-Uniform System I.Interfacial Free Energy*, vol. 28. J.Chemistry and Physics., 1958.

[CL93] CABRAL B., LEEDOM L. C.: Imaging Vector Fields Using Line Integral Convolution. In *Poceedings of ACM SIGGRAPH 1993* (1993), Annual Conference Series, pp. 263–272.

[CM92] CRAWFIS R., MAX N.: Direct Volume Visualization of Three-Dimensional Vector Fields. In *VVS '92: Proceedings of the 1992 workshop on Volume visualization* (New York, NY, USA, 1992), ACM, pp. 55–60.

[DGH03] DOLEISCH H., GASSER M., HAUSER H.: Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In *Data Visualization, Proceedings of the 5th Joint IEEE TCVG–EUROGRAPHICS Symposium on Visualization (VisSym 2003)* (May 2003), pp. 239–248.

[Dov95] DOVEY D.: Vector Plots for Irregular Grids. IEEE Visualization, pp. 248–253.

[FG98] FUHRMANN A. L., GRÖLLER M. E.: Real-Time Techniques for 3D Flow Visualization. In *Proceedings IEEE Visualization '98* (1998), IEEE, pp. 305–312.

[Gel01] GELDER A. V.: Stream Surface Generation for Fluid Flow Solutions on Curvilinear Grids. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '01)* (May 2001), pp. 95–106.

[GEO02] GRANT J., ERLEBACHER G., O'BRIEN J.: Case Study: Visualizing Ocean Flow Vertical Motions Using Lagrangian-Eulerian Time Surfaces. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 529–532.

[GKT*08] GARTH C., KRISHNAN H., TRICOCHE X., BOBACH T., JOY K. I.: Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (Nov.-Dec. 2008), 1404–1411.

[GPR*01] GARCKE H., PRUSSER 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 (July-September 2001), 230–241.

[GPR*04] GRIEBEL M., PREUSSER T., RUMPF M., SCHWEITZER M. A., TELEA A.: Flow field clustering via algebraic multigrid. 35–42.

[GTS*04] GARTH C., TRICOCHE X., SALZBRUNN T., BOBACH T., SCHEUERMANN G.: Surface Techniques for Vortex Visualization. In *Data Visualization, Proceedings of the 6th Joint IEEE TCVG–EUROGRAPHICS Symposium on Visualization (VisSym 2004)* (May 2004), pp. 155–164.

[HA04] HELGELAND A., ANDREASSEN O.: Visualization of Vector Fields Using Seed LIC and Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (2004), 673–682.

[HE06] HELGELAND A., ELBOTH T.: High-quality and interactive animations of 3d time-varying vector fields. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1535–1546. Member-Anders Helgeland.

[Hul92] HULTQUIST J. P. M.: Constructing Stream Surfaces in Steady 3D Vector Fields. In *Proceedings IEEE Visualization '92* (1992), pp. 171–178.

[HWHJ99] HECKEL B., WEBER G. H., HAMANN B., JOY K. I.: Construction of vector field hierarchies. In *Proceedings IEEE Visualization '99* (1999), pp. 19–26.

[JEH01] JOBARD B., ERLEBACHER G., HUSSAINI M. Y.: Lagrangian-Eulerian Advection for Unsteady Flow Visualization. In *Proceedings IEEE Visualization '01* (October 2001), IEEE Computer Society, pp. 53–60.

[JL97] JOBARD B., LEFER W.: Creating Evenly–Spaced Streamlines of Arbitrary Density. In *Proceedings of the Eurographics Workshop on Visualization in Scientific Computing '97* (1997), vol. 7, pp. 45–55.

[KM96] KNIGHT D., MALLINSON G.: Visualizing Unstructured Flow Data Using Dual Stream Functions. *IEEE Transactions on Visualization and Computer Graphics 2*, 4 (Dec. 1996), 355–363.

[Lan94] LANE D.: UFAT - a Particle Tracer for Time-Dependent Flow Fields. In *Proceedings of Visualization '94* (Los Alamitos, CA, 1994), IEEE Computer Society Press, pp. 257–264.

[Lar03] LARAMEE R. S.: FIRST: A Flexible and Interactive Resampling Tool for CFD Simulation Data. *Computers & Graphics 27*, 6 (2003), 905–916.

[LBS03] LI G. S., BORDOLOI U., SHEN H. W.: Chameleon: An Interactive Texture-based Framework for Visualizing Three-dimensional Vector Fields. In *Proceedings IEEE Visualization '03* (2003), IEEE Computer Society, pp. 241–248.

[LEG*08] LARAMEE R. S., ERLEBACHER G., GARTH C., THEISEL H., TRICOCHE X., WEINKAUF T., WEISKOPF D.: Applications of Texture-Based Flow Visualization. *Engineering Applications of Computational Fluid Mechanics (EACFM) 2*, 3 (Sept. 2008), 264–274.

[LGSH06] LARAMEE R. S., GARTH C., SCHNEIDER J., HAUSER H.: Texture-Advection on Stream Surfaces: A Novel

Hybrid Visualization Applied to CFD Results. In *Data Visualization, The Joint Eurographics-IEEE VGTC Symposium on Visualization (EuroVis 2006)* (2006), Eurographics Association, pp. 155–162,368.

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., POST F. H., VROLIJK B., WEISKOPF D.: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum 23*, 2 (June 2004), 203–221.

[LJH03] LARAMEE R. S., JOBARD B., HAUSER H.: Image Space Based Visualization of Unsteady Flow on Surfaces. In *Proceedings IEEE Visualization '03* (2003), IEEE Computer Society, pp. 131–138.

[LPPW95] LEEUW W., PAGENDARM H., POST F., WALTZER B.: Visual Simulation of Experimental Oil-Flow Visualization by Spot Noise from Numerical Flow Simulation. In *Visualization in Scientific Computing '95*. Springer-Verlag, May 1995, pp. 135–148.

[LPSW96] LODHA S. K., PANG A., SHEEHAN R. E., WITTENBRINK C. M.: UFLOW: Visualizing Uncertainty in Fluid Flow. In *Proceedings IEEE Visualization '96* (Oct. 27–Nov. 1 1996), pp. 249–254.

[LS07] LI L., SHEN H.-W.: Image-based streamline generation and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 630–640.

[LSH04] LARAMEE R. S., SCHNEIDER J., HAUSER H.: Texture-Based Flow Visualization on Isosurfaces from Computational Fluid Dynamics. In *Data Visualization, The Joint Eurographics-IEEE TVCG Symposium on Visualization (VisSym '04)* (2004), Eurographics Association, pp. 85–90,342.

[LvW95] LEEUW W., VAN WIJK J.: Enhanced Spot Noise for Vector Field Visualization. In *Proceedings IEEE Visualization '95* (Oct. 1995), IEEE Computer Society, pp. 233–239.

[LvWJH04] LARAMEE R. S., VAN WIJK J. J., JOBARD B., HAUSER H.: ISA and IBFVS: Image Space Based Visualization of Flow on Surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (Nov. 2004), 637–648.

[MB95] MAX N., BECKER B.: Flow Visualization Using Moving Textures. In *Proceedings of the ICASW/LaRC Symposium on Visualizing Time-Varying Data* (Sept. 1995), pp. 77–87.

[MBC93] MAX N., BECKER B., CRAWFIS R.: Flow Volumes for Interactive Vector Field Visualization. In *Proceedings IEEE Visualization '93* (Oct. 1993), IEEE Computer Society, pp. 19–24.

[MCG94] MAX N., CRAWFIS R., GRANT C.: Visualizing 3D Velocity Fields Near Contour Surfaces. In *VIS '94: Proceedings of the conference on Visualization '94* (Los Alamitos, CA, USA, 1994), IEEE Computer Society Press, pp. 248–255.

[MHHI98] MAO X., HATANAKA Y., HIGASHIDA H., IMAMIYA A.: Image-Guided Streamline Placement on Curvilinear Grid Surfaces. In *Proceedings IEEE Visualization '98* (1998), pp. 135–142.

[MLZ09] MCLOUGHLIN T., LARAMEE R., ZHANG E.: Easy Integral Surfaces: A Fast, Quad-based Stream and Path Surface Algorithm. In *Proceedings of Computer Graphics International (CGI 09)* (Victoria, Canada, May 26-29 2009). forthcoming.

[MT*03] MATTAUSCH O., THEUSSL T., , HAUSER H., GROLLER E.: Strategies for Interactive Exploration of 3D Flow Using Evenly-Spaced Illuminated Streamlines. In *Proceedings of the 19th Spring Conference on Computer Graphics* (2003), pp. 213–222.

[PL08] PENG Z., LARAMEE R.: Vector Glyphs for Surfaces: A Fast and Simple Glyph Placement Algorithm for Adaptive Resolution Meshes. In *Proceedings of Vision, Modeling, and Visualization (VMV) 2008* (Constance, Germany, 8-10 October 2008), pp. 61–70.

[PLCZ09] PENG Z., LARAMEE R. S., CHEN G., ZHANG E.: Glyph and Streamline Placement Algorithms for CFD Simulation Data. In *NAFEMS World Congress Conference Proceedings (NWC)* (June 2009), NAFEMS–The International Association for the Engineering Analysis Community. forthcoming.

[PVH*02] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: Feature Extraction and Visualization of Flow Fields. In *Eurographics 2002 State-of-the-Art Reports* (2–6 September 2002), pp. 69–100.

[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 (Dec. 2003), 775–792.

[RSHTE99] REZK-SALAMA C., HASTREITER P., TEITZEL C., ERTL T.: Interactive Exploration of Volume Line Integral Convolution Based on 3D-Texture Mapping. In *Proceedings IEEE Visualization '99* (1999), pp. 233–240.

[SBH*01] SCHEUERMANN G., BOBACH T., HAGEN H., MAHROUS K., HAMANN B., JOY K. I., KOLLMANN W.: A Tetrahedral-based Stream Surface Algorithm. In *Proceedings IEEE Visualization 2001* (Oct. 2001), pp. 151–157.

[SH95] STALLING D., HEGE H.: Fast and Resolution Independent Line Integral Convolution. In *Proceedings of ACM SIGGRAPH 95* (1995), Annual Conference Series, ACM SIGGRAPH, ACM Press / ACM SIGGRAPH, pp. 249–256.

[SLCZ09] SPENCER B., LARAMEE R., CHEN G., ZHANG E.: Evenly-Spaced Streamlines for Surfaces. *Computer Graphics Forum* (Forthcoming 2009).

[ST90] SHIRLEY P., TUCHMAN A.: A Polygonal Approximation to Direct Scalar Volume Rendering. In *Computer Graphics (San Diego Workshop on Volume Visualization)* (Nov. 1990), vol. 24, pp. 63–70.

[STW*08] SHI K., THEISEL H., WEINKAUF T., HAUSER H., MATKOVIC K., HEGE H.-C., SEIDEL H.-P.: Path Line Attributes - an Information Visualization Approach to Analyzing the Dynamic Behavior of 3d Time-Dependent Flow Fields. In *Topo-In-Vis 2007* (Grimma, Germany, 2008), Springer series of Mathematics and Visualization, Springer, pp. 60–74.

[STWE07] SCHAFHITZEL T., TEJADA E., WEISKOPF D., ERTL T.: Point-based Stream Surfaces and Path Surfaces. In *GI '07: Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), ACM, pp. 289–296.

[Sun03] SUNDQUIST A.: Dynamic Line Iintegral Convolution for Visualizing Streamline Evolution. In *IEEE Transactions on Visualization and Computer Graphics* (2003), vol. 9(3), pp. 273–282.

[SVL91] SCHROEDER W., VOLPE C. R., LORENSEN W. E.: The Stream Polygon: A Technique for 3D Vector Field Visualization. In *Proceedings IEEE Visualization '91* (1991), pp. 126–132.

[SvW91] STOLK J., VAN WIJK J.: Surface Particles for 3d Flow Visualization,. In *Proceedings of the Second Eurographics Workshop on Visualization in Scientific Computing* (April, 1991), To be published by Springer Verlag, pp. 22–24.

[TB96] TURK G., BANKS D.: Image-Guided Streamline Placement. In *ACM SIGGRAPH 96 Conference Proceedings* (Aug. 1996), pp. 453–460.

[TK93] TAYLOR B., KUYATT C.: *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*. Technical report, National Institute of Standards and Technology Technical Note 1297, Gaithersburg, MD, January 1993.

[TvW99] TELEA A., VAN WIJK J.: Simplified Representation of Vector Fields. In *Proceedings IEEE Visualization '99* (1999), pp. 35–42.

[TvW03] TELEA A., VAN WIJK J. J.: 3D IBFV: Hardware-Accelerated 3D Flow Visualization. In *Proceedings IEEE Visualization '03* (2003), IEEE Computer Society, pp. 233–240.

[UA01] U.TROTTENBERG, A.SCHULLER: *Multigrid*. Academic Press, Inc., Orlando, FL, USA, 2001.

[UKSE08] ÜFFINGER M., KLEIN T., STRENGERT M., ERTL T.: Gpu-based Streamlines for Surface-Guided 3D Flow Visualization. In *Proceedings of Vision, Modeling, and Visualization (VMV) 2008* (Constance, Germany, 8-10 October 2008), pp. 100–106.

[vFWTS08] VON FUNCK W., WEINKAUF T., THEISEL H., SEIDEL H.: Smoke Surfaces: An Interactive Flow Visualization Technique Inspired by Real-World Flow Experiments. *Proceedings of IEEE Visualization '08* (Oct. 2008).

[VKP00] VERMA V., KAO D., PANG A.: A Flow-guided Streamline Seeding Strategy. In *Proceedings IEEE Visualization 2000* (2000), pp. 163–170.

[VP04] VERMA V., PANG A.: Comparative Flow Visualization. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (2004), 609–624. Senior Member-Alex Pang.

[vW91] VAN WIJK J. J.: Spot noise-Texture Synthesis for Data Visualization. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)* (1991), Sederberg T. W., (Ed.), vol. 25, pp. 309–318.

[vW93a] VAN WIJK J.: Flow Visualization with Surface Particles. *IEEE Computer Graphics and Applications 13*, 4 (July 1993), 18–24.

[vW93b] VAN WIJK J.: Implicit Stream Surfaces. In *Proceedings of the Visualization '93 Conference* (Oct. 1993), IEEE Computer Society, pp. 245–252.

[vW02] VAN WIJK J. J.: Image Based Flow Visualization. *ACM Transactions on Graphics 21*, 3 (2002), 745–754.

[vW03] VAN WIJK J. J.: Image Based Flow Visualization for Curved Surfaces. In *Proceedings IEEE Visualization '03* (2003), IEEE Computer Society, pp. 123–130.

[WE04] WEISKOPF D., ERTL T.: GPU-Based 3D Texture Advection for the Visualization of Unsteady Flow Fields. In *WSCG 2004 Conference Proceedings, Short Papers* (February 2004), pp. 259–266.

[WEE03] WEISKOPF D., ERLEBACHER G., ERTL T.: A Texture-Based Framework for Spacetime-Coherent Visualization of Time-Dependent Vector Fields. In *Proceedings IEEE Visualization '03* (2003), pp. 107–114.

[WG97] WEGENKITTL R., GRÖLLER M. E.: Fast Oriented Line Integral Convolution for Vector Field Visualization via the Internet. In *Proceedings IEEE Visualization '97* (Oct. 19–24 1997), pp. 309–316.

[WMW86] WYVILL G., MCPHEETERS C., WYVILL B.: Data Structure for Soft Objects. *The Visual Computer 2*, 4 (February 1986), 227–234.

[WSE05] WEISKOPF D., SCHAFHITZEL T., ERTL T.: Real-Time Advection and Volumetric Illumination for the Visualization of 3D Unsteady Flow. In *Data Visualization, Proceedings of the 7th Joint EUROGRAPHICS–IEEE VGTG Symposium on Visualization (EuroVis 2005)* (May 2005), pp. 13–20.

[WSE07] WEISKOPF D., SCHAFHITZEL T., ERTL T.: Texture-Based Visualization of Unsteady 3d Flow by Real-Time Advection and Volumetric Illumination. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 569–582.

[XW05] XU R., WUNSCH D.: Survey of Clustering Algorithms. *Neural Networks, IEEE Transactions on 16*, 3 (2005), 645–678.

[XZC04] XUE D., ZHANG C., CRAWFIS R.: Rendering Implicit Flow Volumes. In *VIS '04: Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 99–106.

[YKP05] YE X., KAO D., PANG A.: Strategy for Seeding 3D Streamlines. In *Proceedings IEEE Visualization 2005* (2005), pp. 471–476.

[ZRL96] ZHANG T., RAMAKRISHNAN R., LIVNY M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. *SIGMOD Rec. 25*, 2 (1996), 103–114.

[ZSH96] ZÖCKLER M., STALLING D., HEGE H.-C.: Interactive Visualization of 3D-vector Fields Using Illuminated Stream Lines. In *VIS '96: Proceedings of the 7th conference on Visualization '96* (Los Alamitos, CA, USA, 1996), IEEE Computer Society Press, pp. 107–ff.