State-of-the-Art Report 2002 in Flow Visualization

Helwig Hauser, Robert S. Laramee, and Helmut Doleisch

{Hauser, Laramee, Doleisch}@VRVis.at
VRVis Research Center, Austria
http://www.VRVis.at/

Abstract

Flow visualization has been a very attractive field within visualization research for a long time already. Usually huge datasets need to be processed, which often consist of multi-variate data with a really large number of sample locations, often arranged in multiple time-steps. Recently, the ever increasing performance of computers again has become a driving factor for a new boom in flow visualization (FlowViz), especially in FlowViz based on additional computation such as feature extraction, vector field clustering, and topology extraction. In this state-of-the-art report, an attempt was made to (1) provide a useful categorization of FlowViz solutions, (2) give a survey-like overview about existing solutions, and (3) focus on recent work, especially in the field of FlowViz based on derived data. We give careful consideration as to how these topics are best organized for such a presentation. In separate sections we describe (a) direct FlowViz techniques such as using arrows, (b) FlowViz using integral object such as stream lines, (c) space-filling FlowViz, including, spot noise or line integral convolution, and (d) FlowViz based on derived data such as flow topology. Within those sections, the discussion of FlowViz literature is sub-structured accoring to the dimensionality of the flow data (from 2D to 3D).

Categories and Subject Descriptors (according to ACM CCS): I.3 [Computer Graphics]: visualization, flow visualization, computational flow visualization

---

1. Introduction

Surely, the invention of computers was a major step in the history of mankind – nowadays, in all aspects of society – science, business and economics, telecommunications etc., computers are used to acquire, store, process, and communicate data (not at the least to the user). *Visualization*, as a separate field of research and development in computer science, addresses exactly the bridge between data and user: visualization solutions help users to explore, analyze, and present their data.

In *flow visualization* (FlowViz) – one of the traditional sub-fields of visualization – a rich variety of application fields is given, including automotive industry, aerodynamics, turbomachinery design, weather simulation and meteorology, medical applications, etc., with significantly different characteristics relating to the data and user goals. Consequently, the spectrum of FlowViz solutions is very rich, spanning multiple dimensions of technical aspects, e.g., 2D vs. 3D solutions, techniques for steady and time-dependent data, etc.

Bringing many of those solutions in a linear order (as necessary for a text like this one), is not at all easy or intuitive. Several options of subdividing this broad field of literature are possible. Hesselink, Post, and van Wijk, for example, addressed the difficult problem of how to categorize FlowViz solutions in their 1994 overview about (at this time) recent research issues[49]. In the following paragraphs several of those aspects are discussed on higher level, before literature is addressed directly.

- **Direct flow visualization vs. integration-based FlowViz vs. FlowViz based on derived data.**

  According to the different needs of the users there are different approaches to flow visualization. One is to facilitate an as direct as possible translation of the flow data into visualization cues, such as drawing arrows. FlowViz solutions of this kind allow immediate investigation of the flow data, without a lot of associated translation effort.

  For better communication of long-term behavior induced by flow dynamics, integration-based approaches first inte-

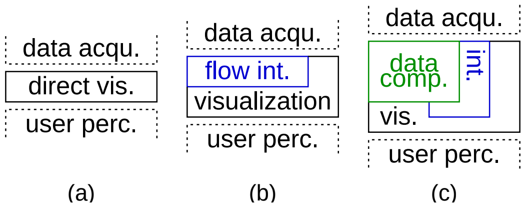

Figure 1: *Direct flow visualization (a) vs. FlowViz based on flow integration (b) vs. FlowViz based on derived data such as flow features or flow topology (c). This classification relates to the first-level structure of this report.*

grate the flow data and use resulting integral objects as basis for visualization, e.g., using stream lines for visualization.

Another step in complexity is to perform yet more computation and derive topological information or flow features before actually doing the visualization mapping. With this kind of FlowViz solutions, a significant amount of computation is spent during visualization to help the user with the interpretation of the flow data. This is especially useful (or even necessary) in cases where very large datasets are given, for example, many time-steps of unsteady 3D flow data.

In this literature overview we use separate chapters for the above mentioned classes of approaches: direct FlowViz is discussed in Sect. 2, integration-based FlowViz then in Sects. 3 and 4, and FlowViz based on derived data is described in Sect. 5. Fig. 1 illustrates the difference between the above mentioned classes – note the increasing amount of computation spent within the visualization step when changing from direct FlowViz (a) to FlowViz based on derived data (c).

- **Spatial dimensions vs. time.**

In flow visualization, available solutions significantly differ with respect to the given dimensionality of the flow data. Techniques which are useful for 2D data, like color coding or arrow plots, for example, sometimes lack similar advantages in 3D. Also, the question, whether the flow data is steady or time-dependent, usually makes a big difference with respect to the FlowViz solution of choice. Fig. 2 illustrates these differences with respect to data dimensionality.

In this state-of-the-art report, we sub-structure the sections about different classes of FlowViz solutions into subsections with respect to different spatial dimensions involved. The sections start with a sub-section on 2D techniques (Sects. *n*.1), i.e., FlowViz solutions which focus on 2D flow data (in 2D domains).

A second sub-section (Sects. *n*.2) discusses FlowViz solutions for boundary flows or sub-sets of 3D flows, for example, flow data on sectional slices. This sub-section therefore deals with (rather) 2D flow data, but embedded within 3D space. Whereas boundary flows often are primarily interesting to the user anyway (for example in aerospace design), the visualization of sectional sub-sets of 3D flow usually needs special care (not at the least because of the usually missing third flow component).

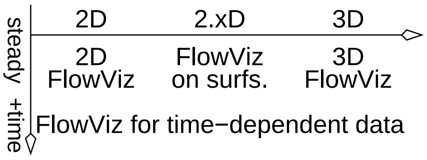

Figure 2: *FlowViz spans different spatial dimensions, and also time (1D & nD omitted here). This categorization corresponds to the second-level structure of this report.*

Finally, a third sub-section (Sects. *n*.3) discusses 3D FlowViz, i.e., visualization techniques, which apply to true 3D flow data. With true 3D FlowViz, rendering becomes a central issue – in many cases compromises are needed, trading visibility for completeness. Solutions range from clipping and opacity modulations to feature-based selections.

Despite of spatial dimensions as addressed above, also dimensionality with respect to time matters a lot in flow visualization. First of all, flow data themselves incorporate a notion of time – flows often are interpreted as differential data with respect to time, i.e., when integrating the data, paths of moving entities are obtained.

Additionally, the flow itself can change over time (like in turbulent flows, for example), resulting in time-dependent or unsteady data. In this case, two dimensions of time are present and the visualization must carefully distinguish between both in order to prevent the user from being confused. This is especially true, when animation should be used for flow visualization.

Although the distinction between steady and unsteady flows could open another dimension when sorting FlowViz literature, in this report solutions for time-dependent data are put aside to related techniques for steady data.

- **Placement and interaction.**

Many FlowViz solutions build on the use of individual visualization objects, for example, stream lines. For at least three reasons, the placement of those visualization cues is an issue within FlowViz literature: (1) when using integral objects such as stream lines, an even distribution of seed locations usually does not result in an even distribution of integral objects – separate algorithms need to be employed; (2) when dealing with 3D flow data, occlusion and/or visualization complexity raises special challenges – dense placement often results in severe cluttering within rendered images; (3) when using feature-based strategies, placement needs to be coupled (and aligned) with the feature extraction parts of the visualization.

In addition to placement, user interaction plays an important role, especially in case of flow analysis. Users require systems which allow (1) navigation, including zooming, panning, etc., (2) interactive placement of visualization cues, for example, using an interactive rake for stream line seeding, as well as other means to influence the visualization, or even (3) options of interacting with the flow data, for example, through steering.

In this report we interleave the discussion of placement and interaction issues with the above mentioned order.

- **Data from simulation vs. measurements or models.**

As one major sub-field of visualization (and as the core topic of this survey), computational *flow visualization* deals with data that exhibit temporal dynamics such as results from flow simulation (e.g., the simulation of fluid flow through a turbine), flow measurements (possibly acquired through laser-based technology), or analytic models of flows (e.g., dynamical systems, given as set of differential equations).

In this report we mainly focus on flow visualization dealing with data from flow simulation, i.e., flow data given as a set of samples on some kind of grid, whereas solutions for data from flow measurements or flow modeling are only addressed in less detail.

- **Special challenges in flow visualization.**

When browsing through FlowViz literature, several challenges appear repeatedly, not to mention only those which are related to the handling of data with multiple dimensions and time-dependency. Stream line *seeding* deals with the problem of where to start multiple stream lines such that the flow domain is covered with stream lines according to a given spatial distribution (evenly or feature-based). Seeding of integral objects is a challenge in 2D but especially also in 3D.

Another issue in FlowViz is the treatment of data with special respect to the underlying *grid* involved. Techniques for visualizing flow data on unstructured grids are special challenges, involving separate strategies for volume rendering, flow integration, topology extraction, etc.

Yet another FlowViz issue is *accuracy*. On one hand, users need to know how well the flow simulation corresponds to reality – comparisons between computational flow visualization and experimental FlowViz are employed to answer such questions. On the other hand, the visualization itself needs to be validated. Components such as flow integration or feature extraction are potential sources of errors that should be checked carefully.

Technical issues frequently arise due to the combination of extremely large datasets and demanding user requirements such as interactive visualization of time-dependent data. Therefore, solutions in the field of parallel computing[11, 73, 131, 161], out-of-core rendering[143], and rendering of compressed data [63] are often discussed in the FlowViz literature.

Last but not least *human-computer interaction* challenges present themselves throughout flow visualization research, especially in the categories of perception in 3D, and interaction. For there is strong evidence that both 3D visualization[150] and interaction[51] are very important components for the user in understanding the data.

- **Compromises made.**

Naturally, this state-of-the-art report focuses on rather recent work to demonstrate what is currently possible in the field of flow visualization. Nevertheless, older but still well accepted solutions are used as a context for embedding newer achievements. Thereby, this overview also serves as a survey about what solutions currently are available in the broad field of flow visualization, given certain user goals and specific data characteristics. We have carefully chosen a selection of literature relating to flow visualization research while also considering the constraints imposed by the limited space available for this presentation.

This literature overview clearly focuses on computational flow visualization. There are many interesting solutions in the field of experimental as well as empirical FlowViz, e.g., based on optical techniques, which could not be addressed in this report. Interestingly, the reader might find a lot of analogies in the computational FlowViz domain, which relate to similar (and older) techniques of experimental flow visualization (e.g. streak lines, tufts, particle tracing, etc.).

Some topics, which also could be addressed in a state-of-the-art report like this, such as FlowViz in 1D or more than three dimensions or FlowViz with focus on flow models or measured flow data, could not be described in detail, mostly because of limited space.

- **Outline.**

In the following, four classes of approaches in the field of flow visualization are discussed (in the next four sections) – direct FlowViz is described in Sect. 2, FlowViz based on integral objects in Sect. 3, and dense, integration-based FlowViz in Sect. 4), as well as FlowViz based on derived data in Sect. 5. Work which is related to flow visualization is discussed in Sect. 6.

2. Direct flow visualization

Direct flow visualization techniques attempt to present the data in a straight forward manner with minimal computation between data acquisition and rendering. These techniques are perhaps the most intuitive visualization strategies as they present the data as is. Difficulties arise, when the long-term behavior induced by flow data is investigated, if direct FlowViz is used – this requires cognitive integration of visualization results.

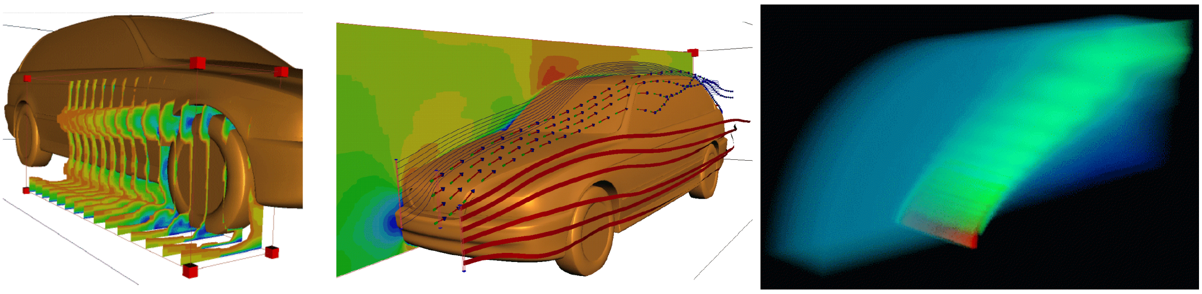

Figure 3: *examples of direct flow visualization – an interactive slicing probe with colored slices and scalar clipping (left); a colored slice, stream lines, 3D arrows along path lines, and stream ribbons (middle), both courtesy by Schulz et al.*[120]*; direct volume rendering based on resampling (right), image courtesy by Westermann*[156].

2.1. Direct FlowViz in 2D

In this subsection we shortly address widely distributed standard techniques for 2D FlowViz, i.e., coloring and contouring, as well as arrow plots (a.k.a. hedgehog visualization).

- **Color coding in 2D.**

A common direct flow visualization technique is to map flow attributes such as velocity, pressure, or temperature to color. Since color plots are widely distributed, this approach results in very intuitive depictions. Of course, the color scale which is used for mapping must be chosen carefully with respect to perceptual differentiation.

Color coding for 2D FlowViz very well extends to time-dependent data, resulting in moving color plots according to changes of the flow properties over time.

- **Contouring in 2D.**

Contouring is a natural extension to color coding in 2D. A contour is a boundary between two distinct regions. Often, the user is highly interested in transition areas in the vector field. In a color plot, transitions are shown by a change of color. With contouring, an explicit line or curve is drawn.

- **Arrow plots in 2D.**

A natural vector visualization technique is to map an line, arrow, or glyph to each sample point in the field (as in Fig. 10, left), which is oriented along the flow field. Usually a regular placement of arrows is used in 2D, for example, on an evenly-spaced Cartesian grid. Two variants of arrow plots are often used: (1) normalized arrows of unit length which visualize the direction of the flow only and (2) arrows of varying length that is proportional to the flow velocity. Klassen and Schroeder call this technique a *hedgehog visualization* (because of the bristly result)[71, 119].

2D hedgehog plots can be extended to time-dependent data, although bigger time-steps might result in jumping arrows, diminishing the quality of such a visualization.

- **Hybrid direct FlowViz in 2D.**

Kirby et al. propose simultaneous visualization of multiple values (of 2D flow data) by using a layering concept related to the painting process of artists[69]. Arrow plots are mixed with color coding to provide visualization results rich of information.

2.2. Direct FlowViz on slices or boundaries

When dealing with 3D flow data, visualization naturally faces additional challenges such as 3D rendering. Acting as a middle ground between 2D FlowViz and the visualization of truly 3D flow data is the restriction to sub-dimensional parts of the 3D domain, e.g., sectional slices or boundary surfaces. Thereby, techniques known from 2D FlowViz usually are applicable without major changes. When working with sectional slices, the treatment of flow components orthogonal to slices requires some special care.

- **Color coding and contouring on slices or boundaries.**

Color coding and contouring are also very effective for visualizing boundary flows or sectional sub-sets of 3D flow data. A good example is NASA's Field Encapsulation Library[92], which allows to easily use both techniques for various flow data.

Schulz, in the group of Ertl, also uses color coding of scalars on 2D slices through 3D automotive simulation data[120] as shown in Fig. 3 (left). They introduce an interactive slicing probe which maps the vector field data to hue.

The use of scalar clipping, i.e., the transparent rendering of slice regions where the corresponding data value does not lie within a specific data range, allows to use multiple (colored) slices with reduced problems due to occlusion.

- **2D arrows on slices or boundary surfaces.**

Using 2D arrows on slices from 3D flow data is also an effective visualization technique[32]. However, results of such a visualization should be interpreted carefully, as flow components which are orthogonal to the slice are usually not depicted.

Above mentioned difficulties with 2D arrows and sectional slices through 3D flow are basically negligible, when talking about boundary surfaces, since in these cases, rarely cross-boundary flows are given. Therefore the use of arrows spread out over boundary surfaces usually is very effective, as used by Treinish for weather visualization[138].

2.3. Direct FlowViz in 3D

After discussing direct FlowViz on slices and boundary surfaces, direct FlowViz of real 3D flows is discussed in this subsection. In contrast to previously mentioned techniques, here rendering becomes the most critical issue. Occlusion and complexity make it difficult (if possible at all) to get an immediate overview about an entire flow dataset in 3D.

- **Volume rendering for 3D FlowViz.**

The natural extension of color coding in 2D (or on slices, etc.) is color coding in 3D. This, however, poses special requirements onto rendering due to occlusion problems and non-trivial complexity – volume rendering is needed (or iso-surfacing, which would relate to contouring in two dimensions). Volume rendering is well-known from another field of research (far beyond the scope of this text), i.e., volume visualization. However, those challenges, which closely correspond to flow visualization are shortly addressed here: (1) flow datasets often are significantly smoother than medical data – an absence of sharp and clear "object" boundaries (like organ boundaries) makes mapping to opacities more difficult (and less intuitive). (2) flow data often is given on non-Cartesian grids, e.g., on curvilinear grids – the complexity of volume rendering get significantly more tricky on those kinds of grids, starting with non-trivial solutions required for visibility sorting. (3) flow data also can be time-dependent, imposing additional loads on the rendering process.

Already in the early nineties, Crawfis, Max, and others[17], as well as Ebert et al.[31] applied volume rendering techniques to vector fields. Little later, Frühauf applied ray casting to vector fields[38]. Recently, Westermann, presented a relatively fast 3D volume rendering method using a resampling technique for vector field data from unstructured to Cartesian grids[156]. A result from this technique is illustrated in Fig. 3 (right).

Recently, Clyne and Dennis[16] as well as Glau[42] presented volume rendering for time-varying vector fields using algorithms which make special use of graphics hardware. Ono et al. use direct volume rendering to visualize thermal flows in the passenger compartment of an automobile[96] Their goal is to attain the ability to predict the thermal characteristics of the automotive cabin through simulation. Swan et al. apply direct volume rendering techniques in flow visualization in a system that supports computational steering[130]. Their visualization results are extended to the CAVE environment.

Recently, Ebert and Rheingans demonstrated the use of non-photorealistic volume rendering techniques for 3D flow data [30]. They apply, for example, silhouette enhancement or tone shading to improve renderings of 3D flows.

- **Iso-surfaces for 3D FlowViz.**

Extending contouring from 2D to 3D, results in the use of iso-surfaces for 3D flow visualization. Special care needs to be taken with iso-value selection, mostly because of the usually smooth nature of flow data – in cases of no sharp transitions within the data, any iso-value lacks (at least partially) intuitive interpretation. Nevertheless there are useful applications of iso-surfaces to flow data, e.g., in the visualization of shock waves[132] or burning fronts in simulated combustion data. Furthermore, when scalar clipping is used together with color coding of slices, this naturally combined with iso-surfaces as long as iso-value and clipping value coincide, of course.

Röttger, Kraus, and Ertl present a hardware accelerated volume rendering technique which allows to use multiple (semi-transparent) iso-surfaces for visualization[107]. Treinish applies iso-surfacing to visualize (unsteady) weather data[138]. Weber et al.[151] present crack-free iso-surface extraction for adaptive (multi-resolution) grids. Laramee and Bergeron provide iso-surfaces for super adaptive grids[75].

- **Arrow plots in 3D.**

The use of arrows for direct 3D FlowViz poses at least two problems: (1) the position and orientation of a vector is often difficult to understand because of its projection onto a 2D screen – using 3D representations of arrows (like a cylinder plus a cone) decreases these problems with perception and (2) glyphs occluding one another become a problem – careful seeding is required (in contrast to the default of dense distributions).

In actual applications, arrow plots usually are based on selective seeding, for example, all arrows starting from one out of a few sectional slices through the 3D flow. Sometimes 3D arrows along certain integral curves are used (see Fig. 3, middle).

Boring and Pang address the problem of clutter in 3D direct FlowViz by highlighting those parts of a 3D arrow plot, which point in a similar direction compared to a user-defined direction[8]. Their methodology reduces the amount of data being displayed thus results in less clutter. Their methods can be combined with other techniques that use glyph representations and flow geometries such as stream lines for FlowViz. They apply the methods to both analytic and simulation datasets to highlight flow reversals.

Figure 4: *three images from an interactive exploration of a vector filed using the MR viewer, image courtesy of Jobard and Lefer*[62]. *A suitable level of resolution can be chosen while maintaining a roughly constant stream line density.*

3. FlowViz using integral objects

Direct flow visualization using hedgehog plots focuses on individual points in the flow field. However, more elaborate schemes are introduced when these points, or similar objects, are moved over a small time step. A hedgehog glyph approximates the motion of a point for the time period indicated by the glyph itself. A logical extension of this technique is to depict the motion of a point over more than one time step. The resulting path mathematically is expressed as an integral. This instantaneous path may be depicted as a stream line (in the case steady flow fields are considered).

3.1. 2D FlowViz using integral objects

In this subsection we shortly discuss 2D FlowViz techniques based on integral objects such as streamlets, stream lines, and their relatives within unsteady flows. Also, the seeding problem is addressed, which requires a solution in order to realize better distributions of integral objects.

- **Streamlets in 2D.**

If flow vectors are integrated for a very short time, *streamlets* are generated. Even though short, streamlets already communicate temporal evolution along the flow. Fig. 10 (middle) shows an example, where several streamlets are used to visualize a 2D flow field.

- **Stream lines in 2D.**

If longer integration is performed (as compared to streamlets), *stream lines* are gained. They are a natural extension of glyph-based techniques and offer intuitive semantics: users easily understand that flows evolve along integral objects.

- **Streak lines, time lines, and path lines.**

When unsteady flow data are investigated, several distinct integral objects are used for flow visualization. A *path line* or *particle trace* is the trajectory that a particle follows in fluid flow[119]. A *time line* joins the positions of particles released at the same instant in time from different insertion points, i.e., joins points at a constant time t[94]. A *streak line* is traced by a set of particles that have previously passed through a unique point in the domain[119]. Streak lines relate to continuous injection of foreign material in real flow.

- **Stream line seeding in 2D.**

One problem with stream lines, or integral curves, when used for visualizing continuous vector fields is the best choice of initial conditions. Since, in general, evenly distributed seed points do not result in evenly spaced stream lines, special algorithms need to be employed. Turk and Banks[141] as well as Jobard and Lefer[59] developed a techniques for automatically placing seed points to achieve a uniform distribution of stream lines on a 2D vector field.

Stream line seeding strategies may also be *topology-based*. Verma, Kao, and Pang[149] presents a seed placement strategy for stream lines based on flow features in the dataset. Their goal is to capture flow patterns near critical points in the flow field.

Building on their previous work, Jobard and Lefer presented a multi-resolution (MR) method for visualizing large, 2D, steady-state vector fields[62]. The MR hierarchy supports enrichment and zooming. The user is able to interactively set the density of stream lines while zooming in and out of the vector field (Fig. 4). The density of stream lines can be computed automatically as a function of velocity or vorticity.

Seeding of integral objects becomes a special challenge when dealing with time-dependent data. Jobard and Lefer presented an unsteady FlowViz algorithm by correlating instantaneous visualizations of the vector field at the stream

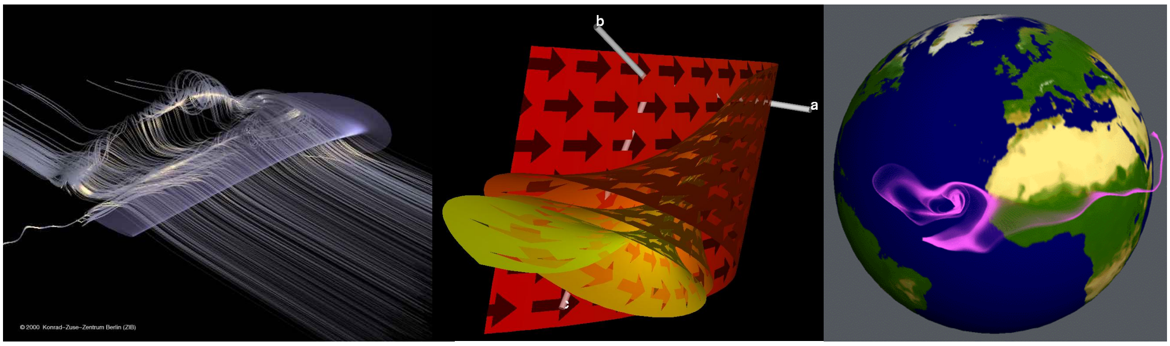

Figure 5: *examples of flow visualization using integral objects – illuminated stream lines (left), image courtesy of Hege et al.*[160]*; stream arrows (middle), image courtesy of Hauser*[86]*; and flow volumes (right), image courtesy of Max, Becker, and Crawfis*[90]*.*

line level[61]. For each frame, a feed forward algorithm computes a set of evenly-spaced stream lines as a function of the stream lines generated for the previous frame. Their method also provides full control of the image density so that smooth animations of arbitrary density can be produced.

3.2. FlowViz using integral objects on slices or boundaries

After talking about 2D FlowViz based on integral objects, this subsection shortly addressed similar approaches on subsets of 3D flows such as boundary flows. Interpretation of integral curves on sectional slices required special care, again.

- **Integrated tufts.**

  Wegenkittl et al. use *integrated tufts* (similar to streamlets), seeded on specific equilibrium surfaces, for the visualization of a complex dynamical system[154], also over variations of that system in a fourth dimension.

- **Integral objects on slices or boundaries.**

  Similar to 2D FlowViz, integral objects such as stream lines are also used for visualizing boundary flows or sectional slices through 3D flow[32]. However, it is important to note that the use of integral objects on slices may be misleading, even within steady flow datasets. A stream line on a slice may depict a closed loop, even though no particle would ever traverse the loop. The reason again lies in the fact, that flow components which are orthogonal to the slice are omitted during flow integration.

- **Stream line seeding on boundary surfaces.**

  Mao et al.[88] extend the stream line seeding of Turk and Banks[141] in order to generate evenly distributed stream lines on boundary surfaces within curvilinear grids.

3.3. 3D FlowViz using integral objects

When dealing with 3D flow, a rich variety of integral objects is available for flow visualization. This sub-section addressed a series of integral objects, from streamlets to flow volumes, primarily sorted according to their dimensionality, and within equal dimensionality roughly with respect to which technique extends which other.

- **Streamlets in 3D.**

  Streamlets easily extend to 3D, although perceptual problems might arise due to distortions resulting from the rendering projection. Also, seeding becomes more important in 3D, again. Löffelmann and Gröller use a thread of streamlets along characteristic structures of 3D flow to gain selective, but importance-based seeding as well as as enhancement of abstract flow topology through direct visualization cues[82].

- **Stream lines in 3D.**

  At NASA the Flow Analysis Software Toolkit (FAST)[2] is used to visualization CFD data based on stream lines in 3D. Careful seeding is necessary to obtain useful results, since easily visual clutter can become a problem.

- **Illuminated stream lines.**

  Zöckler, Stalling, and Hege present illuminated stream lines to improve perception of stream lines in 3D by taking advantage of the texture mapping capabilities supported by graphics hardware[160]. Their shading technique increases depth information. By making the stream lines partially transparent, they also address the problem of occlusion as shown in Fig. 5 (left).For seeding, the authors propose an interactive seeding probe which can be moved around to start stream lines at specific places of interest. Also, seeding near potential objects of interests is demonstrated.

- **Particle tracing in 3D.**

  Kenwright and Lane present an efficient, 3D particle tracing algorithm that is also accurate for interactive investigation of large, unsteady, aeronautical simulations[67]. A performance gain is obtained by applying tetrahedral decomposition to speed up point location and velocity interpolation in curvilinear grids.

Teitzel, Grosso, and Ertl analyze different integration methods in order to evaluate the trade-off between time and accuracy[135, 136]. They present a 3D particle tracing algorithm targeted at sparse grids that is very efficient with respect to storage space and computing time. The authors recommend using sparse grids as a data compression method in order to visualize huge datasets.

Nielson presents efficient and accurate methods for computing tangent curves for 3D flows[93]. Their methods work directly with physical coordinates, eliminating the need to switch back and forth with computational coordinates. Efficient particle tracing methodologies are also addressed by Sadarjoen et al.[108].

Since stream lines usually are easily computed in real-time, they offer (together with their intuitive semantics) an often chosen tool for interactive flow analysis. Bryson and Levit[10] demonstrate seeding of integral objects in a virtual 3D environment by use of a so-called *rake*. See Fig. 5 (middle) for another example of a rake being used to seed integral curves.

- **Stream ribbons and stream tubes.**

  A first extension of stream lines in 3D are *stream ribbons* (Fig. 3) and *stream tubes*. A stream ribbon basically is a stream line with a wing-like strip added to also visualize rotational behavior of 3D flow (which is not possible with stream lines alone)[142]. A stream tube is a thick stream line that can be extended to show the expansion of the flow[142]. Stream ribbons and stream tubes offer advantages over stream lines in that way that they can encode more properties such as divergence and convergence of the vector field in the geometric properties of the respective integral object.

  Ueng et al. present techniques for efficient stream line, stream ribbon, and stream tube constructions on unstructured grids[142]. A specialized Runge-Kutta method is employed to speed up stream line computation. Explicit solutions are calculated for the angular rotation rates of stream ribbons and the radii of stream tubes. The resulting speed-up in overall performance aids in the exploration of large flow fields.

  Fuhrmann and Gröller[39] use so-called *dash tubes*, i.e., animated, opacity-mapped stream tubes, as a visualization icon. An algorithm is described which places the dash tubes evenly in 3D space. They also apply a magic lens and magic box as interaction techniques for investigating densely filled areas without filling the image with visual detail and complexity.

  Laramee introduces the *stream runner* as an extension of stream tubes – an interactively controlled 3D flow visualization technique that attempts to minimize occlusion, minimize visual complexity, maximize directional cues, and maximize depth cues by letting the user control the length of the stream tubes[74].

- **Stream polygons.**

  Another extension of stream lines are *stream polygons* used by Schroeder[118]. Stream polygons are tools to visualize vectors and tensors using tubes with a polygonal cross section. The properties of the polygons such as the radius, the number of sides, the shape, the rotation reflect properties of the vector field including strain, displacement, and rotation.

- **Stream balls and streak balls.**

  Stream balls are a useful flow visualization technique used by Brill et al.[9], which visualizes divergence and *acceleration* in fluid flow. Stream balls split/merge dependent on convergence/divergence or acceleration/deceleration, respectively.

  Teitzel and Ertl introduce *streak balls* when they present and compare two different approaches to accelerate particle tracing on sparse grids and curvilinear sparse grids for unsteady flow data[133].

- **Stream surfaces.**

  Yet another extension to stream lines are *stream surfaces* which are surfaces that are everywhere tangent to a vector field. A stream surface can be approximated by connecting a set of stream lines along time lines (and varying the number of stream lines used according to convergence or divergence of the flow). Stream surfaces present challenges related to occlusion, visual complexity, and interpretation.

  Hultquist presents an interactive flow visualization technique using stream surfaces[52]. Cai and Heng[13] address the issues associated with the placement and orientation of stream surfaces in 3D.

  Löffelmann, Mroz, and Gröller present *stream arrows* (Fig. 5, middle) as an enhancement of stream surfaces by separating arrow-shaped portions from a stream surface[86, 85]. Stream arrows address the problem of occlusion associated with 3D flow visualization, but especially with stream surfaces. Stream arrows also provide additional information about the flow, usually not seen with stream surfaces, such as flow direction, convergence/divergence, etc.

  Van Wijk simulates stream surfaces by a large set of so-called surface particles[147]. Surface particles exhibit less occlusion when compared to stream surfaces. Interestingly, van Wijk's approach in a way anticipated recent advances in pixel-based rendering techniques.

- **Time surfaces in 3D.**

  A natural extension of time lines (in 2D or 3D) are *time surfaces*, when constant-time instants of moving particles are assumed, which previously have been released from a two-dimensional patch. An example of an application of this principle, are level-set surfaces used by Westermann[157].

- **Flow volumes.**

  The last (direct) extension of a stream line into 3D described here are *flow volumes* (Fig. 5, right). A flow volume is a specific sub-set of a 3D flow domain, which is traced out by a particular initial 2D patch over time as described by Max, Becker, and Crawfis. The resulting volume is divided up into a set of semi-transparent tetrahedra, which are volume rendered in hardware in a way derived from the method of Shirley and Tuchmann[125].

Becker et al. extend flow volumes to unsteady flow[5]. The resulting unsteady flow volumes are the 3D analog of streak lines. Considerations are made when extending the visualization technique to unsteady flows to since particle paths may become convoluted in time. The authors present some solutions to the problems which occur in subdivision, rendering, and system design. The resulting algorithms are applied to a variety of flow types including curvilinear grids.

4. Dense, integration-based FlowViz

Here we make a distinction between flow visualization using integral objects and dense, integration-based flow visualization, however, these two topics are closely coupled: conceptually, the path from using integral objects to dense, integration-based visualization is obtained via a dense seeding strategy. That is, densely seeded integral objects result in an image similar to that obtained by dense, integration based techniques. Likewise, the path from dense, integration-based visualization to visualization using integral objects is obtained using something such as a sparse texture for texture advection.

Integration-based techniques in flow visualization can provide dense spatial resolution images. Dense, texture-based algorithms are effective, versatile, and applicable to a wide spectrum of applications. Sanna, Montrucchio, and Montuschi present an excellent summary of this research in their survey paper[113].

4.1. Dense, integration-based FlowViz in 2D

In this subsection, we describe dense, integration-based FlowViz solutions for 2D flow data, i.e., spot noise, line integral convolution (LIC), and related approachs.

- **Spot noise (in 2D).**

Spot noise, introduced by van Wijk[146] was amongst the first texture-based technique for vector field visualization. Spot noise generates a texture by distributing a set of intensity functions, or spots, over the domain. Each spot represents a particle moving over an infinitesimal time and results in a streak in the direction of the local flow from where the particle is seeded.

One limitation of the original spot noise algorithm was the lack of velocity magnitude information in the resulting texture. Enhanced spot noise[26], by de Leeuw and van Wijk was introduced to address this problem. Spot noise has also been applied to the visualization of turbulent flow[22] by de Leeuw et al. A spot noise algorithm for interactive visualization is proposed by de Leeuw[20], also. De Leeuw and van Liere also compare spot noise to LIC[23]. Spot noise in 2D combined with color coding is shown in Fig. 6.

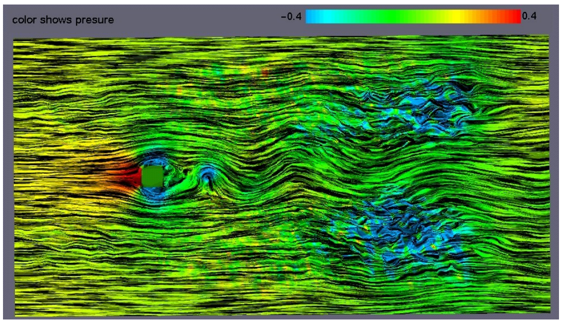

Figure 6: *colored spot noise, image courtesy of de Leeuw.*

- **Line integral convolution in 2D.**

Line integral convolution (LIC; Fig. 10, right), first introduced by Cabral and Leedom[12] is a very popular technique for the dense coverage of vector fields with flow visualization cues. The original methodology behind LIC takes as input a vector field on a cartesian grid and a white noise texture of the same size. The noise texture is locally filtered (smoothed) along the path of stream lines to acquire a dense visualization of the flow field.

The research in flow visualization based on LIC described here extends LIC in several ways: (1) adding directional cues, (2) showing velocity magnitudes, (3) added support for non-cartesian grids, (4) allowing real-time and interactive exploration, (5) extending LIC to 3D, and (6) extending LIC to unsteady vector field visualization with time coherency.

Shen et al. address the problem of directional cues in LIC by combining animation and introducing dye advection into the computation[122]. Kiu and Banks proposed to use a multi-frequency noise for LIC[70]. The spatial frequency of the noise is a function of the magnitude of the local velocity in the field.

Khouas et al. synthesize LIC-like images in 2D with fur-like textures[68]. Their technique is able to locally control attributes of the output texture such as orientation, length, density, and color.

Much research has been dedicated to bringing LIC computation to interactive rates. Stalling and Hege present significant improvements in LIC performance by exploiting coherence along stream lines[129] and[46]. Parallel implementations of LIC are presented by Cabral and Leedom[11], and Zöckler in the group of Hege[161].

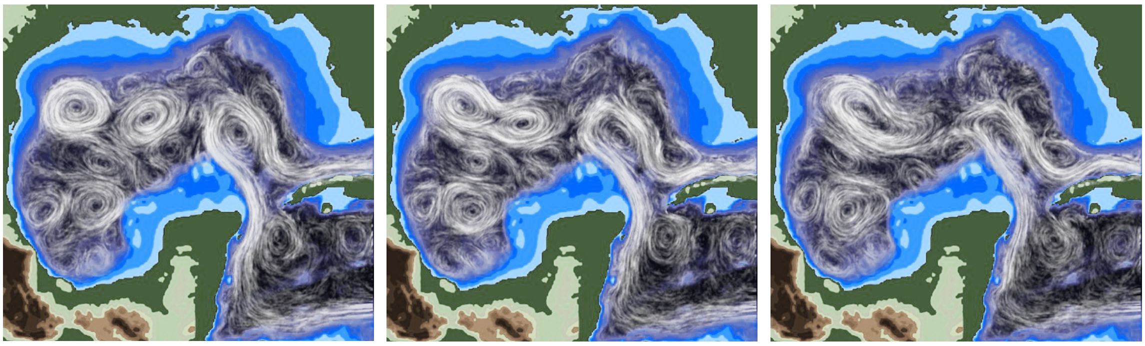

Figure 7: *three images taken from an animation of an unsteady vector field created with the Lagrangian- Eulerian advection algorithm, image courtesy of Jobard et al.*[57] *(data set provided by COAPS, Florida State University).*

- **OLIC for 2D FlowViz.**

  Wegenkittl et al. also address the problem of orientation of flow with their OLIC (Oriented Line Integral Convolution) approach[152]. Conceptually, the OLIC algorithm makes use of a sparse texture containing of many separated spots which are kind of smeared in the direction of the local vector field through integration. A fast version of OLIC (called FROLIC) is presented by Wegenkittl and Gröller[153] via a trade off of accuracy for time. Berger and Gröller present an algorithm for animating 2D FROLIC images over the world wide web[7].

  Löffelmann and Gröller use virtual ink droplets, like streamlets, to visualize 2D dynamical systems[83]. Similar to oriented line integral convolution (OLIC), the virtual ink droplet method is capable of visualizing not only direction and velocity of flow, but also the orientation of vectors. See Fig. 10 for a 2D comparison of streamlets to LIC.

- **2D Texture Advection.**

  Jobard and Lefer use a motion map data structure for animating 2D, steady-state flow fields[60]. The motion map contains both a dense representation of the flow and the information required to animate the flow. It offers the advantage of saving memory and computation time since only one image of the flow has to be computed and stored in the motion map data structure.

  Jobard et al. propose a technique to visualize dense representations of unsteady vector fields based on what they call a Lagrangian-Eulerian Advection scheme[57]. The algorithm combines a dense, time-dependent, integration based representation of the vector field with interactive frame rates. Some results from the technique are shown in figure 7.

  Unsteady flow visualization techniques may address the problem of interactive performance time through the use of texture-mapping supported by the graphics hardware. Becker and Rumpf illustrate hardware-supported texture transport for 2D, unsteady flow data[6].

  Jobard et al.[58, 56] present additional 2D, unsteady flow visualization techniques. They achieve high performance via the use of graphics hardware. They also detail spatial and temporal coherence techniques, dye advection techniques, and feature extraction.

- **Dense 2D FlowViz based on streak lines.**

  Sanna et al. present an adaptive visualization method using streak lines where the seeding of streak lines is a function of local vorticity[112].

4.2. Dense, integration-based FlowViz on surfaces or boundaries

Dense, integration-based techniques are, in general, better methods for conveying flow information on sectional slices than techniques using (long) integral objects. This is because the connection along the path of what would be a stream line is lost with dense integration-based techniques. Thus the depiction of the flow is not misleading in terms of a potential suggestions of particle paths. Let us recall that the vector component orthogonal to the slice is removed when using integral objects for visualization results.

- **Spot noise on boundaries or slices.**

  De Leeuw et al. extend the spot noise algorithm to surfaces in a study that compares experimental surface flow visualization (with oil) to that of spot noise on surfaces[21].

  A combination of both texture-based FlowViz (on slices) and 3D arrows for 3D FlowViz is employed by Telea and van Wijk[137] where arrows denote the main characteristics of the 3D flow (after clustering) and a 2D slice with spot noise or LIC is used to visualize the rest of the vector field (on a slice only). This is shown in Fig. 11.

- **LIC for boundary flows.**

  A large body of research literature is dedicated to the extension of LIC on to boundary surfaces, surveyed, for example, by Stalling[128].

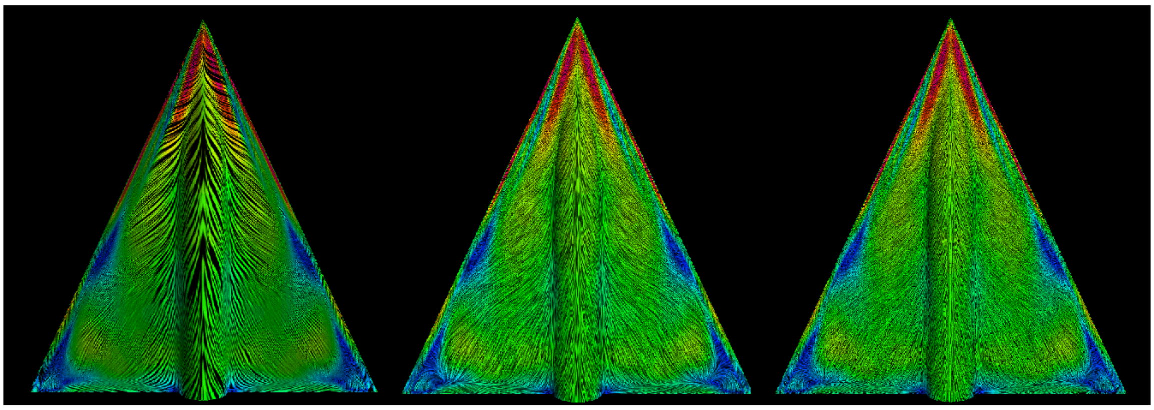

Figure 8: *a comparison of 3 LIC techniques: (left) UFLIC, (middle) ELIC, and (right) PLIC image courtesy of Pang et al.*[148]

The extension of LIC to non-cartesian grids and surfaces is presented by researchers such as Forssell[33]. Forssell and Cohen[34] extend LIC to curvilinear surfaces with animation techniques, add magnitude and direction information, and show how to use LIC to depict time-dependent flows. Their algorithm also utilizes texture-mapping hardware to improve performance time towards interactive rates.

Teitzel, Grosso, and Ertl[134] present an approach that works on both 2D unstructured grids and directly on triangulated surfaces in three-dimensional space. Mao et al.[87], present an algorithm for convolving solid white noise on triangle meshes in 3D space, and extend LIC for visualizing a vector field on arbitrary, 3D surfaces.

Battke, Stalling, and Hege[4] describe an extension of LIC for arbitrary surfaces in 3D. Some approaches are limited to curvilinear surfaces, i.e., surfaces which can be parametrized by using 2D-coordinates. Their method also handles the case of general, multiply connected surfaces.

Scheuermann in the group of Hagen, presents a method for visualizing 3D vector fields that are defined on a 3D manifold[114]. Their work addresses the normal vector component to the surface that other methods do not.

A problem with many curvilinear grid LIC algorithms is that the resulting LIC textures may be distorted after being mapped onto the geometric surfaces, since a curvilinear grid usually consists of cells of different sizes. Mao in the group of Kikukawa propose a solution to the problem by using multi-granularity noise as the input image for LIC[89].

- **UFLIC, PLIC, etc.**

Shen and Kao present UFLIC (Unsteady Flow LIC, Fig. 8, left)[123, 124] which incorporates time into the convolution. Their algorithm addresses problems with temporal coherency by successively updating the convolution results over time. They also propose a parallel UFLIC algorithm.

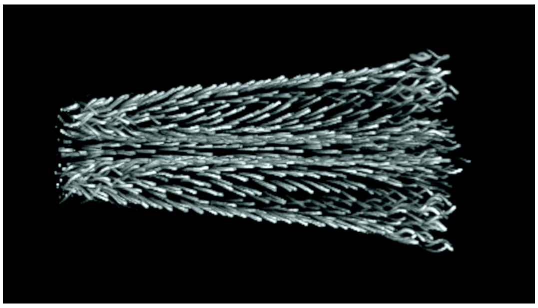

Figure 9: *3D LIC, courtesy of Interrante and Grosch*[53].

Verma in the group of Pang, presents a method for comparative analysis of stream lines and LIC called PLIC[148]. A visual comparison between ELIC (enhanced LIC)[95], PLIC, and UFLIC is shown in Fig. 8.

4.3. Dense, integration-based FlowViz in 3D

High computational costs, demanding memory requirements, occlusion, and visual complexity can all be inhibitants for dense, integration-based flow visualization in 3D.

- **LIC in 3D.**

Occlusion and interactive performance are non-trivial challenges to overcome implementing LIC in 3D (shown in Fig. 9). Rezk-Salama et al. tackle the problem of interactive performance using a 3D-texture mapping approach combined with an interactive clipping plane to address the problems of occlusion and interaction[104].

A combination approach of direct volume rendering and LIC is taken by Interrante[55] for extending LIC to 3D. Interrante and Grosch address some perceptual difficulties en-

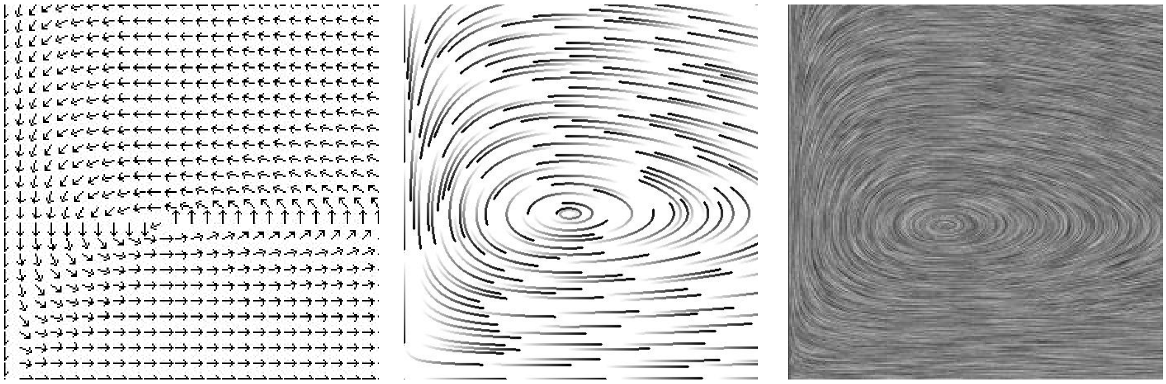

Figure 10: *example of comparing FlowViz techniques from Sects. 2, 3, and 4, image courtesy of Hauser*[80]*. FlowViz by the use of arrows (left) is compared to FlowViz based on integral objects (middle), and space-filling FlowViz by the use of LIC (right).*

countered with dense, 3D visualizations[53, 54, 55]. Techniques for selectively emphasizing important regions of interest in the flow, enhancing depth perception, and improving orientation perception of overlapping stream lines are discussed.

- **Texture advection in 3D.**

  Kao et al. discuss the use of 3D and 4D texture advection for the visualization of 3D fluid flows[64]. Formidable challenges are introduced by the memory requirements involved in using 3D and 4D textures. They also apply a steady-state animation to these 3D and 4D textures.

5. FlowViz based on derived data

The visualization methodologies presented in this section require the most computation between data acquisition and resulting perception at the user's side. Computations are run on the input data to acquire additional data about the input. This derived data (depending on the application) might be flow topology, flow features, aggregated flow data (through clustering), meta-level flow data, or others. Original data as well as derived data are then used for visualization, allowing for enhanced visualization of flow data. Associated with the derived data is added complexity. The benefits in these techniques lie on the user-side: lousely speaking, more work is done by the visualization software and less work is done on behalf of the user, e.g., less work with interpretation.

5.1. 2D FlowViz based on derived data

This section starts off with a discussion of vector field clustering techniques. Secondly, we introduce feature-based flow visualization techniques. Topology-based visualization and vortex visualization are presented as subcategories of feature-based flow visualization. Finally, we introduce two more categories: flow visualization based on local analysis and meta-level flow visualization.

- **Vector field clustering.**

  Vector field clustering methods attempt to balance visual complexity and complete flow coverage in both 2D and 3D vector fields. Instead of trying to visualize each vector in the dataset, aggregated information about the dataset may be visualized. In short, larger numbers of finer resolution vectors are replaced by fewer representatives at a coarser resolution in the visualization. The vectors that remain attempt to summarize those found at the finest level of resolution in the original dataset.

  Lodha et al. present an algorithm for compressing 2D vector fields while also preserving topology[79]. They use different types error measures including the earth mover's distance metric to measure the topological degradation. Examples with both analytic and simulated datasets.

  Telea and van Wijk present a vector field clustering technique for 2D vector fields that allows the user to adjust parameters resulting in simplified vector field visualization[137]. They also show how to extend the algorithm to 3D, as shown in Fig. 11.

  Heckel in the group of Joy, presents a vector field clustering technique for generating an entire hierarchical representation of the vector field[45]. More than one level in the hierarchy can be visualized simultaneously.

  Garcke et al. present a continuous vector field clustering technique to simulation data[40] with the goal of varying the representation from fine granularity to very few, coarse clusters. They demonstrate a general applicability of their approach, e.g., also in 3D[41]. Their algorithms are focused on 2D vector fields, however, they show their extension to 3D vector fields as well.

- **Feature-based flow visualization.**

  When describing *feature-based flow visualization*, we use the term *feature* to refer to a special subset or structure of

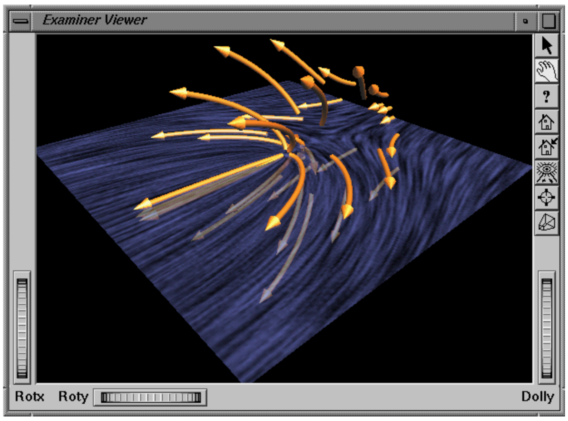

Figure 11: *3D FlowViz based on clustering, combined with a textured slice; image courtesy of Telea and van Wijk*[137].

interest in the original flow dataset. Features can be classified as *local* or *non-local*, depending on what information is being represented. Examples of local features are critical points. Vortex cores are non-local features, spanning non-infinitesimal small regions of the flow domain. Reinders et al., for example, present a 2D feature-extraction technique for application in the astrophysical data[101]. Their system is able to extract cloud features from large data sets.

Feature-based visualization techniques address problems associated with large datasets. Applying a direct flow visualization technique to a large CFD simulation dataset does not guaranteed a meaningful visualization, especially in 3D. Details are easily lost when direct volume rendering methodologies are applied the vector field visualization as a result of (1) occlusion, (2) the projection of 3D information onto a 2D screen, or (3) just because of their relatively small size compared to other features. Emphasizing the most interesting components of the flow field may be a better choice. Note that what is deemed as “interesting” is determined by the user.

Feature-based visualization provides the user with more capabilities to guide the visualization process, allowing the possibility to define the features to be visualized either *explicitly* or *implicitly*. Explicit feature definition refers to the scenario when the user knows *a priori* what features to visualize. Implicit feature definition involves separating features into subsets that have similar attributes, e.g., showing all regions of flow whose velocity lies above a given threshold. Henze uses multiple 2D views featuring geometric connectivity in an approach called Linked Derived Spaces[48], which are linked in terms of shared color maps, for examepl, and allow discrete brushing to interactively specify flow features.

The representation of different types of features is another important field of research. Features have different attributes, and to emphasize special attributes for each type of feature, suitable representations are developed. For vortices, critical points, and other topological features, glyphs or icons can be used. One example are the ellipses (ellipsoids in 3D) used by Sadarjoen and Post[109]. They encode the rotation speed vortex, size, and other attributes of vortices.

- **Topology-based flow visualization.**

An important property of a flow field is its topology. *Vector field topology visualization* was introduced by Helman and Hesselink[47]. They present essential information by partitioning the flow field according to its critical points. Lavin et al. present a technique by which they compare 2D vector fields for similarities based upon the characteristics of critical points found in each dataset[76]. The goal of their research is the ability to compare computational and experimental flow fields under the same conditions.

How to properly extract flow topology is a separate issue. Especially when talking about data from flow simulation, i.e., data which origins in a (locally) linear computation on a grid, then the extraction of flow topology is non-trivial.

Scheuermann et al. work on higher-order flow topology, i.e., topology of (locally) non-linear flow data[115, 116]. An example of this work is shown in Fig. 12 (left). Scheuermann et al. also investigate improved interpolation schemes for better extraction of flow topology[117].

- **Multi-resolution flow topology.**

The visualization of topology can be impressive when the underlying flow fields are not too complex. But challenges may arise in high-resolution, turbulent flows. Rich CFD simulation datasets can contain a large number of critical points that clutter a resulting image.

An approach using multiple levels of topology by de Leeuw and van Liere[18] proposes a solution to this problem. Their solution is a topological filter that leaves out cluttering but retains the global structure of the flow. The topological filter is called a *pair distance filter* which is defined as the distance between two critical points. This distance can be used as the metric by which to remove (or filter) critical points from the visualization. De Leeuw and van Liere illustrate limitations of the pair distance filter[24]. As an alternative, they propose a method to calculate the inflow and outflow areas of the critical points, which they apply to data from meteorology[19]. The dataset is based on a curvilinear grid and contains measurements of atmospheric wind direction and magnitude.

One disadvantage of the methods presented by de Leeuw and van Liere is that they do not handle higher order critical points. This challenge is addressed by Tricoche et al.[139]. Tricoche et al. present also a topological simplification of vector and tensor fields on irregular grids[140].

A vector field topology simplification can be achieved by merging critical points within a prescribed radius into higher

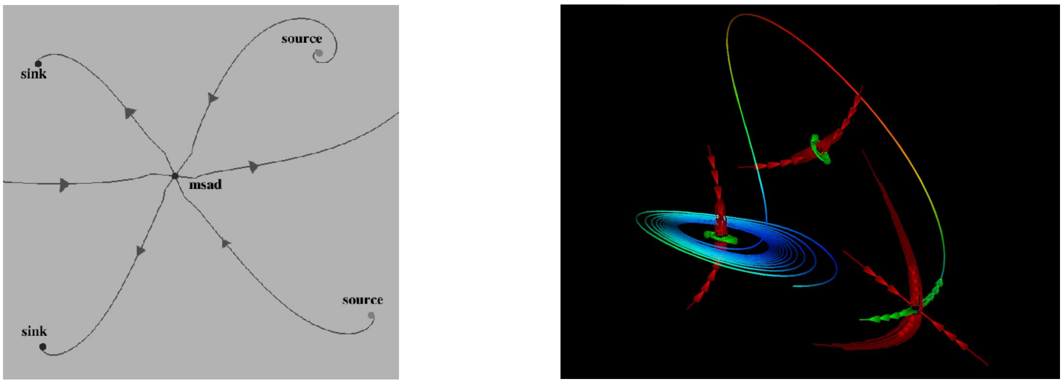

Figure 12: *examples of FlowViz based on flow topology: non-linear flow topology depicted (left), image courtesy by Scheuermann et al.*[115]*; 3D FlowViz based on critical point analysis (right), image courtesy by Hauser et al.*[81].

order critical points. After building clusters containing the singularities to merge, their method generates a representation in which each cluster contains one higher order singularity. Any visualization method can be applied to the result after this process.

- **Vortex visualization.**

  Another valuable area of research in FlowViz is the identification of vortices and their cores. Most related research presents special case techniques for vortex identification as opposed to methodologies that apply in general cases. *Weak Vortices*, for example, present special challenges. Weak vortices have a relatively slow rotational component when compared to the velocity of the flow along the axis of rotation (the vortex core).

  Sadarjoen et al. shows applications of two categories of vortex detection criteria: (1) point-based scalar quantities, calculated at single points, and (2) curve-based geometric criteria, calculated for, e.g., stream lines[111]. The first category is easier to compute, but does not work in all cases. The second category is more intuitive but currently only works in 2D (or 3D projected) flows. They present applications of both approaches in hydrodynamic flows.

  Sadarjoen and Post present two vortex detection methods that are based on the geometric properties of stream lines[109]. Unlike many vortex detection methods that are based on point-samples of physical quantities, one of their methods is also effective in detecting weak vortices. In addition, their methodology allows for quantitative feature extraction using numerical attributes of vortices.

  Sadarjoen and Post extended their vortex visualization methods to unsteady flow[110], also. Their method is applied to areas such as ocean mapping.

- **The accuracy of feature-based flow visualization.**

  Feature-based visualization poses some special challenges: (1) the computational effort required for the extraction process, (2) the evaluation and accuracy of the resulting output, and (3) the topic of completeness arises, e.g., what, if any, important properties been left out of the visualization.

  Reinders et al. address the evaluation and accuracy of feature extraction[103]. Their methodology is applied and tested for different grid resolutions, noise levels, and feature extraction parameters. Peikert and Roth introduce a mathematical framework targeted at a more accurate feature comparisons[98].

5.2. Derived data for FlowViz on slices or boundaries

As boundary flows often are primarily interesting to visualization users anyway, extraction of derived data for flow visualization also is done for flow sub-sets like that. An example is the automatic extraction of attachment and separation lines along flow boundaries, as presented by Kenwright[66].

Wong et al. apply a vector field clustering technique to 2D slices of a regional climate modeling dataset covering East Asia[159]. They also show how to combine their results of multiple slices to gain 3D visualization. Their goal is to eliminate less interesting and sporadic critical points in a multiresolution fashion.

5.3. 3D FlowViz based on derived data

The more extended a flow dataset is the more appropriate FlowViz solutions get which are based on derived data. 3D flow data, especially if it is time-dependent, often is composed of hundreds of thousands of data samples, which (in case of unsteady flows) are given on hundreds of time steps. Clearly, data of that extent pose special challenges to the exploration phase. With FlowViz solutions based on derived data, the user can be supported to more quickly investigate those parts of the data, in which he or she is actually interested in.

De Leeuw and van Wijk, for example, present a technique for interactive flow analysis in 3D[25], which is based on the visualization of locally derived data. Local flow attributes, such as velocity, acceleration, convergence or divergence, rotation, etc., are visualized by the use of glyphs, which change their (geometrical) appearance according to the data to be visualized.

- **FlowViz based on 3D feature extraction.**

Tank and Medioni present a 3D feature extraction technique that aims to extract coherent surfaces and 3D space curves from a dense or sparse 3D grid[132]. Based on noisy data, extremal sub-sets of the data are extracted and visualized by the means of surfaces. The authors demonstrate the extraction of: (1) shock waves, (2) vortex cores, as well as (3) crest-lines and ridges in a terrain map, and (4) grooves, anatomical lines, and complex surfaces from noisy dental data.

Van Walsum et al. present a feature extraction technique that provides a compact abstraction of the original data, and uses icons to visualize the extracted features[145]. Reinders et al. show another feature-based representation[100] that uses a skeletal description of features in the flow via the use of a long, slim, geometric shape.

- **Feature tracking.**

Temporal coherency, also called *feature tracking*, is also valuable for feature-based visualization. Feature tracking addresses the challenge of following flow features over multiple time steps.

Reinders et al. suggest feature tracking methodologies and evaluate the success rate of these techniques[102]. Prediction rules, derived from data from previous time steps, is used to compute regions in which there is a high probability of finding a feature in a future time step. The path of features over time is shown in a feature graph.

Silver and Wang present a basic framework for the visualization of time-varying datasets that features an algorithm and data structure to track volume features in un-structured datasets[127]. The algorithm and data structure are general enough to be applied to structured, curvilinear, adaptive and hybrid grids. Silver and Wang also present a technique that isolates and tracks 3D representations of regions of interest from 3D regular and curvilinear, CFD datasets[126]. Features are extracted from each time step and matched to features in subsequent time steps.

- **Focus and context visualization in 3D.**

The *focus and context* metaphor comes from the field of information visualization and is important when visualizing large datasets resulting from CFD simulation. The *focus* of a visualization (derived from the depth-of-field metaphor from photography) is the region or object with the highest amount of interest to the user. The *context* of the visualization is usually represented in less detail and provides a semantic framework in which to visualize the focus. See the work by Card for more on information visualization[14].

Doleisch and Hauser take advantage of the focus and context metaphor to visualize CFD simulation data with several attributes[29].They extend the work of Henze[48] to allow also for non-discrete feature definition. Also, results from interactive feature extraction are used to modulate opacity within 3D rendering.

- **Topology-based flow visualization in 3D.**

When dealing with 3D data, artists do a fairly good job – intelligently they use only those parts of the data for depiction, which are crucial for understanding the data. *Hand-drawn* flow illustrations, therefore, have been popular since the beginning of scientific flow investigation. Leonardo da Vinci, for example, efficiently used hand drawings to communicate his research results on fluid flows. More recently, Abraham and Shaw came up with visualizing flow structures by using hand-drawn images[1]. They also concentrate on elementary structures within 3D flow data, based on 3D flow topology.

Helman and Hesselink[47] proposed to visualize the geometry of the topological structure of flow dynamics. Stream lines along the eigenvectors of critical points are used to show separatrices. Icons composed of line segments and small disks encode the Jacobian matrix near critical points.

Batra and Hesselink extend the technique presented by Lavin et al.[76] to 3D vector fields[3]. A method for comparing 3D vector fields constructed from simple critical points is described. Globus et al.[43] came up with a tool to identify topological elements within data that is given at a discrete grid.

Löffelmann and Gröller use the results of topology extraction in 3D flow data for selectively placing 3D streamlets[82]. With this strategy, an over-population of phase space with occlusion problems as a consequence is avoided. Furthermore direct, and thus intuitive visualization cues are used for visualization, instead of representing topological structures only.

- **Vortex visualization in 3D.**

Roth and Peikert address visualization of CFD data for turbomachinery design[105]. They argue that FlowViz for turbomachinery design poses some special requirements which are often not addressed by standard FlowViz systems. They discuss the issues involved with this particular application and its requirements with respect to flow visualization. Roth and Peikert also present a method to automatically extract vortical structures from 3D CFD vector field[106] (Fig. 13). They discusses the underlying theory and some aspects of the implementation.

Pagendarm et al. extend the 2D vortex detection method of Sadarjoen and Post[109] to 3D[97]. Kenwright and Haimes present an eigenvector method for vortex identification in an aerodynamic simulation[65]. It is shown to be an effective way

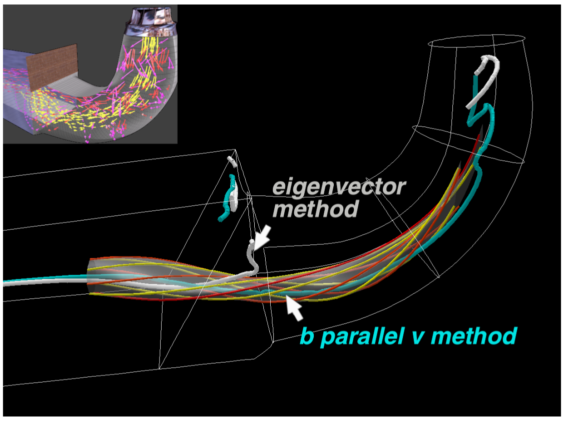

Figure 13: *3D FlowViz based on vortex core extraction, image courtesy by Peikert and Roth*[98].

to extract and visualize features such as vortex cores, spiral vortex breakdowns, vortex bursting, and vortex diffusion. Kenwright and Haims address several challenges including identifying disjointed line segments, detecting non-vortical flow features, and computing vortex core displacement.

- **3D FlowViz using Poincaré maps.**

  Poincaré sections are a useful mathematical tool for the investigation of 3D flows which exhibit a strong rotational component, as, for example, particle traces in a fusion reactor. Using a Poincaré section, the dominating rotational component is separated from the 3D flow, and second-level structures become apparent.

  Löffelmann et al. propose a set of visualization techniques based on Poincaré maps in 3D[84]. They suggest adapting some well-known visualization techniques as, e.g., spot noise, to Poincaré maps to improve the visual representation of the 2D map. They also present an embedding of these techniques within a 3D visualization of the underlying flow. This approach allows to significantly reduce some limitations of previously known techniques.

  Schussman et al. make use of a Poincaré map in order to visualize magnetic field lines[121].

6. Related work

In this section about related work, we address topics, which are closely coupled to flow visualization, but do not belong themselves to FlowViz. These topics include experimental flow visualization, flow integration, grids involved, tensor visualization, visualization of maps, uncertainty visualization, and simulated fluids in computer graphics. We relax our rigid organization for this section in order to simplify the presentation of related work for the reader.

- **Experimental flow visualization.**

  As already metioned previously, this state-of-the-art report clearly focuses on computational flow visualization. However, there is a long history and a large amount of literature about experimental flow visualization. Fluid researchers have been using real *experiments* to get an impression of fluid flow properties and structures for a long time. Experimental methods have advantages including intuitive and immediate feedback and fewer numerical errors. There are also significant disadvantages: most severe is that experimental methods may influence the flow itself. Next, experimental setups usually are time consuming and very expensive. For in-depth information about experimental flow visualization techniques, see Merzkirch[91], Post and van Walsum[99], and Milton van Dyke[144].

- **Flow integration methods.**

  One of the backbones of flow visualization is *flow integration methods*. This is because flow visualization often relies heavily on this type of computation. Hence, several research papers focus solely on optimizing the integration algorithms themselves. *Accurate* numerical integration is an important topic relating to flow visualization. Here, again, the classic trade-off is made between accuracy and computation time. Euler's method of integrating is the simplest, but it is not accurate enough for some applications. A Runge-Kutta integration method of order 2 or 4 offers greater accuracy but at the expense of computation. Integrators may have adaptive step sizes[129], use interpolated data, and may be or implicit or explicit type. Teitzel et al. analyze different integration methods in order to evaluate the trade-off between time and accuracy[135, 136].

  Knight and Mallinson present the use of dual stream functions for generating stream lines and stream surfaces[72]. Van Wijk, for example, uses this kind of approach for his surface particles[147].

- **Grids.**

  In flow visualization, many different types of grids are used. This mostly is due to the fact that data origins in flow simulation and in this fields grids are adapted to support the simulation process as well as possible. As a consequence, non-Cartesian grids such as curvi-linear grids often show up when flows around certain bodies or obstacles are investigated.

  In principal, different types of grids also imply different type of algorithms. Many solutions work well for one type of grids, but are not trivially extended to others. Grids involved range from Cartesian, recti-linear, and curvi-linear grids, to unstructured, irregular grids, and even hybrid combinations such as block-structured grids. Special other types of grids, like sparse grids or multi-resolution grids, or even moving grids, are also experienced. Papers which specifically address a special type of grids have be interleaved within the previous sections.

- **Tensor visualization.**

In addition to flow data visualization, *tensor data visualization* is an active area of research. Tensor fields provide multidimensional data usually represented by the use of matrices. Stress propagation within certain objects like engines, turbines, etc., produce tensor data. Simulation techniques that are similar to methods known from CFD are used to compute dense datasets of volume tensors.

Delmarcelle and Hesselink use hyperstreamlines – a generalization of vector field stream lines – in order to visualize 3D, second-order tensor fields along continuous paths[27].

Hesselink, Levy, and Lavin study the topology of 3D tensor fields[50, 77]. Their goal is to represent their structure by a set of points, lines, and surfaces analogous to approaches in vector field topology. They extract topological skeletons of the eigenvector fields and use them as a description of the tensor field.

Weinstein et al. present a feature-tracking algorithm to tensor field datasets[155].

- **Visualization of discrete flow data.**

Flow data represent temporal changes in a continuous way like appropriate for fluid flows, gas flows, or similar. However, in other applications, like econometric modeling, for example, temporal changes often are represented discretely, i.e., in changes from one discrete point in time to the next one, e.g., population size year after year or temperature heights for series of days.

In their work on how to use Poincaré sections for 3D FlowViz [84], Löffelmann et al. adapt well-known FlowViz solutions from the continuous domain to discrete, 2D Poinaré maps, such as arrow plos, depictions of trajectories, and spot noise.

Hauser et al. present a two-level approach for volume rendering[44], which they use to visualize flow data of such discrete nature. Rendered are so-called basins of attraction, which separate the flow domain into distinct portions which are charaterized by coherent long-term evolution. They also show chaotic attractors which regularly occur in conjunction with such discrete flow data.

- **Uncertainty visualization.**

Uncertainty or errors may be introduced in fluid flow data as the data is acquired, stored, and rendered. Although researchers are aware of such errors, seldom do visualization systems incorporate such uncertainty information. In the absence of the integrated presentation of data and its associated error, the analysis of the visualization is incomplete and may result in misleading, inaccurate, or incorrect conclusions.

Wittenbrink et al. apply an uncertainty visualization technique to meteorologic data in 2D[158]. Environmental data has inherent error which is often ignored in visualization. Their approach includes uncertainty in direction and magnitude, as well as the mean direction and length, in vector glyph plots. The design of the glyph and numerous examples using environmental data are shown.

Cedilnik and Rheingans developed a method for showing the uncertainty information together with data with minimal distraction for 2D results[15]. This method uses procedurally generated annotations and glyphs that are deformed according to the uncertainty information. Lodha in the group of Wittenbrink, presents UFLOW – a system for visualizing uncertainty in fluid flow[78]. The techniques employed to visualize uncertainty in fluid flow include uncertainty glyphs, flow envelopes, animations, priority sequences, twirling batons of trace viewpoints, and rakes. These techniques are effective in visualizing the effects of different integration methods.

Djurcilov in the group with Pang, present a volume rendering algorithm that includes uncertainty visualization in the result[28].

- **Simulated flows in computer graphics.**

Foster et al. present techniques for *user controlled fluid animation*[35, 36, 37]. With their tools, it is possible for computer graphics animators to specify and control a 3D fluid animation, without knowledge of the underlying simulation equations. To illustrate the methods, animations of moving objects, fountains, explosions, and hot gas interacting with solid objects are presented.

7. Summary and conclusions

This paper gives an overview of the current state of the art in flow visualization. The large body of research in vector field visualization is a reflection of its importance to researchers in scientific visualization as well as private industry. We give careful consideration to how techniques in fluid visualization can be categorized and reflect this organization in the presentation. We put emphasis on flow visualization based on derived data because this field of research gained special importance recently. More specifically, we believe feature-based flow visualization, and time-dependent flow visualization are areas for much future work. One possible future challenge of FlowViz based on derived data might be to better support intuition, to support the user also with understanding the visualization process itself (for better acceptance of the results).

8. Acknowledgements

The authors thank all those who have contributed to this research including AVL (`www.avl.com`), the Austrian governmental research program called Kplus (`www.kplus.at`), and the VRVis Center For Virtual Reality and Visualization (`www.VRVis.at`). We also thank Zoltan Konyha for his valuable contributions and feedback. Furthermore, the authors thank all the permission holders of images shown in this report.

References

1. Ralph H. Abraham and Christopher D. Shaw. *Dynamics - the Geometry of Behavior*. Addison-Wesley, 1992. 15

2. Gordon V. Bancroft, Fergus J. Merritt, Todd C. Plessel, P.G. Kelaita, Rich K. McCabe, and Al Globus. FAST: A multiprocessed environment for visualization of computational fluid dynamics. In *Proc. of Visualization 90*, pages 14–27, 1990. 7

3. Rajesh K. Batra and Lambertus Hesselink. Feature comparisons of 3D vector fields using earth mover's distance. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 105–114, San Francisco, 1999. IEEE. 15

4. Henrik Battke, Detlev Stalling, and Hans-Christian Hege. Fast line integral convolution for arbitrary surfaces in 3D. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics*, pages 181–195. Springer-Verlag, Heidelberg, 1997. 11

5. Barry G. Becker, David A. Lane, and Nelson L. Max. Unsteady flow volumes. In *IEEE Visualization '95*, 1995. 9

6. Joachim Becker and Martin Rumpf. Visualization of time-dependent velocity fields by texture transport. In Dirk Bartz, editor, *Visualization in Sientific Computing '98*, Eurographics, pages 91–102. Springer-Verlag Wien New York, 1998. 10

7. Siegrun Berger and Eduard Gröller. Color-table animation of fast oriented line intgral convolution for vector field visualization. In *WSCG 2000 Conference Proceedings*, pages 4–11, 2000. 10

8. Ed Boring and Alex Pang. Directional flow visualization of vector fields. In Roni Yagel and Gregory M. Nielson, editors, *IEEE Visualization '96*, pages 389–392. IEEE, 1996. 5

9. M. Brill, H. Hagen, H.-C. Rodrian, W. Djatschin, and S. V. Klimenko. Streamball techniques for flow visualization. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 225–231, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press. 8

10. Steve Bryson and Creon Levit. The virtual wind tunnel. *IEEE Computer Graphics and Applications*, 12(4):25–34, July 1992. 8

11. Brian Cabral and Casey Leedom. Highly parallel vector visual environments using line integral convolution. In David H. Bailey, Petter E. Bjorstad, John R. Gilbert, Michael V. Mascagni, Robert S. Schreiber, Horst D. Simon, Virgina J. Torczon, and Layne T. Watson, editors, *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, pages 802–807, Philadelphia, PA, February 1995. SIAM. 3, 9

12. Brian Cabral and Leith (Casey) Leedom. Imaging vector fields using line integral convolution. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 263–272, August 1993. 9

13. Wenli Cai and Pheng-Ann Heng. Principal stream surfaces. In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 75–80, Los Alamitos, October 19–24 1997. IEEE Computer Society Press. 8

14. Stuart K. Card. Visualizing retrieved information: a survey. *IEEE Computer Graphics and Applications*, 16(2):63–67, March 1996. 15

15. Andrej Cedilnik and Penny Rheingans. Procedural annotation of uncertain information. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 77–84. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 17

16. J. Clyne and J. Dennis. Interactive direct volume rendering of time-varying data. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 109–120. Springer-Verlag Wien, May 1999. 5

17. Roger Crawfis, Nelson Max, Barry Becker, and Brian Cabral. Volume rendering of 3D scalar and vector fields at LLNL. In IEEE, editor, *Proceedings, Supercomputing '93: Portland, Oregon, November 15–19, 1993*, pages 570–576, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1993. IEEE Computer Society Press. 5

18. Wim de Leeuw and Robert van Liere. Visualization of global flow structures using multiple levels of topology. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 45–52. Springer-Verlag Wien, May 1999. 13

19. Wim de Leeuw and Robert van Liere. Multi-level topology for flow visualization. *Computers and Graphics*, 24(3):325–331, June 2000. 13

20. Wim C. de Leeuw. Divide and conquer spot noise. In *Proceedings of Supercomputing'97 (CD-ROM)*, San Jose, CA, November 1997. ACM SIGARCH and IEEE. Center for Mathematics and Computer Science, Kruislaan, Amsterdam, The Netherlands. 9

21. Wim C. de Leeuw, Hans-Georg Pagendarm, Frits H. Post, and Brigit Waltzer. Visual simulation of experimental oil-flow visualization by spot noise from numerical flow simulation. In R. Scanteni, Jarke J. van Wijk, and P. Zanarini, editors, *Visualization in Scientific Computing '95*, pages 135–148. Springer-Verlag Wien, May 1995. 10

22. Wim C. de Leeuw, Frits H. Post, and Remco W. Vaatstra. Visualization of turbulent flow by spot noise. In M. Göbel, J. David, P. Slavik, and Jarke J. van Wijk, editors, *Virtual Environments and Scientific Visualization '96*, pages 286–295. Springer-Verlag Wien, April 1996. 9

23. Wim C. de Leeuw and Robert van Liere. Comparing LIC and spot noise,. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 359–366. IEEE, 1998. 9

24. Wim C. de Leeuw and Robert van Liere. Collapsing flow topology using area metrics. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 349–354, San Francisco, 1999. IEEE. 13

25. Wim C. de Leeuw and Jarke J. van Wijk. A probe for local flow field visualization. In Gregory M. Nielson and R. Daniel Bergeron, editors, *Proceedings of the Visualization '93 Conference*,

pages 39–45, San Jose, CA, October 1993. IEEE Computer Society Press. 15

26. Wim C. de Leeuw and Jarke J. van Wijk. Enhanced spot noise for vector field visualization. In *IEEE Visualization '95 Proceedings*, pages 233–239. IEEE Computer Society, October 1995. 9

27. Thierry Delmarcelle and Lambertus Hesselink. Visualizing second-order tensor fields with hyperstream lines. *IEEE Computer Graphics and Applications*, 13(4):25–33, July 1993. 17

28. Suzana Djurcilov, Kwansik Kim, Pierre Lermusiaux, and Alex Pang. Volume rendering data with uncertainty information. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 243–252, Wien, Austria, May 28–30 2001. Springer-Verlag. 17

29. Helmut Doleisch and Helwig Hauser. Smooth Brushing for Focus and Context Visualization of Simulation Data in 3D. In *WSCG 2002, International Conference in Central Europe on Computer Graphics, Visualization and Digital Interactive Media*, Plzen, Czech Republic, 2002. accepted for publication. 15

30. David S. Ebert and Penny Rheingans. Volume illustration: Non-photorealistic rendering of volume models. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 195–202. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 5

31. David S. Ebert, R. Yagel, J. Scott, and Y. Kurzion. Volume rendering methods for computational fluid dynamics visualization. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 232–239, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press. 5

32. A. J. Fenlon and T. David. An integrated visualization and design toolkit for flexible prosthetic heart valves. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 453–456. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 4, 7

33. L. K. Forssell. Visualizing flow over curvilinear grid surfaces using line integral convolution. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 240–247, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press. 11

34. Lisa K. Forssell and Scott D. Cohen. Using line integral convolution for flow visualization: Curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):133–141, June 1995. 11

35. Nick Foster and Ronald Fedkiw. Practical animations of liquids. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 23–30. ACM Press / ACM SIGGRAPH, 2001. 17

36. Nick Foster and Dimitri Metaxas. Controlling fluid animation. In *Proceedings CGI '97*, pages 178–188, 1997. Winner of the Androme Award 1997. 17

37. Nick Foster and Dimitri Metaxas. Modeling the motion of hot, turbulent gas. In *Proceedings of SigGraph '97*, 1997. 17

38. Thomas Frühauf. Raycasting vector fields. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 115–120, Los Alamitos, October 27–November 1 1996. IEEE. 5

39. Anton L. Fuhrmann and Eduard Gröller. Real-time techniques for 3D flow visualization. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 305–312. IEEE, 1998. 8

40. Harald Garcke, Tobias Preußer, Martin Rumpf, Alexandru Telea, Ulrich Weikard, and Jarke J. van Wijk. A continuous clustering method for vector fields. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 351–358. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 12

41. Harald Garcke, Tobias Prußer, Martin Rumpf, Alexandru C. Telea, Ulrich Weikard, and Jarke J. van Wijk. A phase field model for continuous clustering on vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):230–241, July-September 2001. 12

42. T. Glau. Exploring instationary fluid flows by interactive volume movies. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 277–283. Springer-Verlag Wien, May 1999. 5

43. Al Globus, Creon Levit, and Thomas Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Visualization '91*, pages 33–40, 1991. 15

44. Helwig Hauser, Lukas Mroz, Gian Italo Bischi, and M. Eduard Gröller. Two-level volume rendering. In *IEEE Transactions on Visualization and Computer Graphics*, volume 7(3), pages 242–252. IEEE Computer Society, 2001. 17

45. Bjoern Heckel, Gunther H. Weber, Bernd Hamann, and Kenneth I. Joy. Construction of vector field hierarchies. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 19–26, San Francisco, 1999. IEEE. 12

46. Hans-Christian Hege and Detlev Stalling. Fast LIC with piecewise polynomial filter kernels. In Hans-Christian Hege and Konrad Polthier, editors, *Mathematical Visualization*, pages 295–314. Springer Verlag, Heidelberg, 1998. 9

47. James L. Helman and Lambertus Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991. 13, 15

48. Chris Henze. Feature detection in linked derived spaces. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 87–94. IEEE, 1998. 13, 15

49. L. Hesselink, Frits H. Post, and Jarke J. van Wijk. Research issues in vector and tensor field visualization. *IEEE Computer Graphics and Applications*, 14(2):76–79, March 1994. 1

50. Lambertus Hesselink, Yuval Levy, and Yingmei Lavin. The Topology of Symmetric, Second-Order 3D Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, March 1997. Survey paper. 17

51. William Hibbard and David Santek. Interactivity is the key. In *Proc. Chapel Hill Workshop on Volume Visualization*, 1989. 3

52. Jeff P. M. Hultquist. Interactive numerical flow visualization using Stream Surfaces. *Computing Systems in Engineering*, 1(2-4):349–353, 1990. 8

53. Victoria Interrante and Chester Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In *Proceedings of Visualization '97*, pages 421–424, 1997. 11, 12

54. Victoria Interrante and Chester Grosch. Visualizing 3D flow. *IEEE Computer Graphics & Applications*, 18(4):49–53, 1998. 12

55. Victoria L. Interrante. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 109–116. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7. 11, 12

56. Bruno Jobard, Gordon Erlebacher, and M. Youssuf Hussaini. Tiled hardware accelerated texture advection for unsteady flow visualization. In *Proceedings of Graphicon 2000*, pages 189–196, Moscow, Russia, August 2000. 10th International Conference on Computer Graphics and Vision. 10

57. Bruno Jobard, Gordon Erlebacher, and M. Youssuf Hussaini. Lagrangian-eulerian advection for unsteady flow visualization. In Wilfred Lefer and M. Grave, editors, *IEEE Visualization*, San Diego, California, October 2001. IEEE. 10

58. Bruno Jobard, Gordon Erlebacher, and M. Yousuff Hussaini. Hardware-accelerated texture advection. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 155–162. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 10

59. Bruno Jobard and Wilfrid Lefer. Creating evenly–spaced streamlines of arbitrary density. In Wilfrid Lefer and M. Grave, editors, *Proceedings of the Eurographics Workshop*, volume 7 of *Visualization in Scientific Computing '97*, Boulogne-sur-Mer, France, April 28-30 1997. Eurographics, Springer-Verlag WienNewYork. 6

60. Bruno Jobard and Wilfrid Lefer. The motion map: Efficient computation of steady flow animations. In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 323–328, Los Alamitos, October 19–24 1997. IEEE Computer Society Press. 10

61. Bruno Jobard and Wilfrid Lefer. Unsteady flow visualization by animating evenly-spaced streamlines. In M. Gross and F. R. A. Hopgood, editors, *Computer Graphics Proceedings (Eurographics 2000)*, volume 19(3), 2000. 7

62. Bruno Jobard and Wilfrid Lefer. Multiresolution flow visualization. *WSCG '01*, 5-9 February 2001. Plzen (Republique Tcheque). 6

63. Chuan kai Yang, Tulika Mitra, and Tzi cker Chiueh. On-the-fly rendering of losslessly compressed irregular volume data. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 101–108. IEEE Computer Society Technical Committee on Computer graphics, 2000. 3

64. David Kao, Bing Zhang, Kwansik Kim, and Alex Pang. 3d flow visualization using texture advection. In *International Conference on Computer Graphics and Imaging '01*, Honolulu, Hawaii, August 2001. 12

65. David Kenwright and Robert Haimes. Vortex identification - applications in aerodynamics. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization 9́7*, pages 413–416. IEEE, November 1997. 15

66. David N. Kenwright. Automatic detection of open and closed separation and attachment lines. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 151–158. IEEE, 1998. 14

67. David N. Kenwright and David A. Lane. Interactive Time-Dependent Particle Tracing Using Tetrahedral Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):120–129, June 1996. 7

68. L. Khouas, C. Odet, and D. Friboulet. 2D vector field visualization using furlike texture. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 35–44. Springer-Verlag Wien, May 1999. 9

69. Robert M. Kirby, H. Marmanis, and David H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of the 1999 IEEE Conference on Visualization (VIS-99)*, pages 333–340, N.Y., October 25–29 1999. ACM Press. 4

70. Ming-Hoe Kiu and David C. Banks. Multi-frequency noise for LIC. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 121–126, Los Alamitos, October 27–November 1 1996. IEEE. 9

71. R. Victor Klassen and Steven J. Harrington. Shadowed hedgehogs: A technique for visualizing 2D slices of 3D vector fields. In *Visualization '91*, pages 148–153, 1991. 4

72. David Knight and Gordon Mallinson. Visualizing Unstructured Flow Data Using Dual Stream Functions. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):355–363, December 1996. 16

73. David A. Lane. Parallelizing a particle tracer for flow visualization. In David H. Bailey, Petter E. Bjorstad, John R. Gilbert, Michael V. Mascagni, Robert S. Schreiber, Horst D. Simon, Virgina J. Torczon, and Layne T. Watson, editors, *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, pages 784–789, Philadelphia, PA, February 1995. SIAM. 3

74. Robert S. Laramee. Interactive 3D Flow Visualization Using a Streamrunner. In *CHI 2002, ACM Conference on Human Factors in Computing Systems, Extended Abstracts*, Minneapolis, Minnesota, April 20-25 2002. ACM SIGCHI, ACM Press. accepted for publication. 8

75. Robert S. Laramee and R. Daniel Bergeron. An Isosurface Continuity Algorithm for Super Adaptive Resolution Data. In *CGI 2002, Computer Graphics International*, Bradford, UK, July 1-5 2002. Computer Graphics Society. accepted for publication. 5

76. Yingmei Lavin, Rajesh Kumar Batra, and Lambertus Hesselink. Feature comparisons of vector fields using earth

mover's distance. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 103–110, 1998. 13, 15

77. Yingmei Lavin, Yuval Levy, and Lambertus Hesselink. Singularities in nonuniform tensor fields. In *IEEE Visualization '97*, October 1997. 17

78. Suresh K. Lodha, Alex Pang, Robert E. Sheehan, and Craig M. Wittenbrink. UFLOW: Visualizing uncertainty in fluid flow. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 249–254, Los Alamitos, October 27–November 1 1996. IEEE. 17

79. Suresh K. Lodha, Jose C. Renteria, and Krishna M. Roskin. Topology preserving compression of 2D vector fields. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 343–350. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 12

80. Helwig Löffelmann. *Visualizing Local Properties and Characteristic Structures of Dynamical Systems*. PhD thesis, Technical University of Vienna, http://www.cg.tuwien.ac.at/ helwig/diss/, 1998. 12

81. Helwig Löffelmann, Helmut Doleisch, and Eduard Gröller. Visualizing dynamical systems near critical points. In László Szirmay Kalos, editor, *14th Spring Conference on Computer Graphics*, pages 175–184. Comenius University, Bratislava, Slovakia, April 1998. ISBN 80-223-0837-4. 14

82. Helwig Löffelmann and Eduard Gröller. Enhancing the visualization of characteristic structures in dynamical systems. In Dirk Bartz, editor, *Visualization in Scientific Computing '98*, Eurographics, pages 59–68. Springer-Verlag Wien New York, 1998. 7, 15

83. Helwig Löffelmann, Andreas König, and Eduard Gröller. Fast visualization of 2D dynamical systems by the use of virtual ink droplets. In Wolfgang Straßer, editor, *13th Spring Conference on Computer Graphics*, pages 111–118. Comenius University, Bratislava, Slovakia, June 1997. ISBN 80-223-1176-6. 10

84. Helwig Löffelmann, Thomas Kučera, and Eduard Gröller. Visualizing poincaré maps together with the underlying flow. In Hans-Christian Hege and Konrad Polthier, editors, *Mathematical Visualization*, pages 315–328. Springer Verlag, Heidelberg, 1998. 16, 17

85. Helwig Löffelmann, Lukas Mroz, and Eduard Gröller. Hierarchical streamarrows for the visualization of dynamical systems. In Wilfred Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, Eurographics, pages 155–164. Springer-Verlag Wien New York, 1997. 8

86. Helwig Löffelmann, Lukas Mroz, Eduard Gröller, and Werner Purgathofer. Stream arrows: Enhancing the use of streamsurfaces for the visualization of dynamical systems. *The Visual Computer*, 13:359–369, 1997. 7, 8

87. X. Mao, M. Kikukawa, N. Fujita, and A. Imamiya. Line integral convolution for 3D surfaces. In Wilfred Lefer and M. Grave, editors, *Visualization in Scientific Computing '97. Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France*, pages 57–70. Eurographics, Springer Verlag, Wien, New York, 1997. 11

88. Xiaoyang Mao, Yuji Hatanaka, Hidenori Higashida, and Atsumi Imamiya. Image-guided streamline placement on curvilinear grid surfaces. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 135–142. IEEE, 1998. 7

89. Xiaoyang Mao, Lichan Hong, Arie Kaufman, Noboru Fujita, and Makoto Kikukawa. Multi-granularity noise for curvilinear grid LIC. In *Graphics Interface*, pages 193–200, June 1998. 11

90. Nelson Max, B. Becker, and Roger Crawfis. Flow volumes for interactive vector field visualization. In Gregory M. Nielson and R. Daniel Bergeron, editors, *Proceedings of the Visualization '93 Conference*, pages 19–24, San Jose, CA, October 1993. IEEE Computer Society Press. 7

91. W. Merzkirch. *Flow Visualization, 2nd edition.* Academic Press, 1987. 16

92. Patrick Moran, Chris Henze, David Ellsworth, Steve Bryson, and David Kenwright. The Field Encapsulation Library (FEL). The Field Encapsulation Library (FEL) is a library for representing fields and the meshes that fields are based on, see http://www.nas.nasa.gov/Groups/VisTech/projects/fel/. 4

93. Gregory M. Nielson. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):360–372, October – December 1999. ISSN 1077-2626. 8

94. Gregory M. Nielson, Hans Hagen, and Heinrich Müller. *Scientific Visualization: Overviews, Methodologies, and Techniques.* IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1997. 6

95. A. Okada and David L. Kao. Enhanced line integral convolution with flow feature detection. In *SPIE Vol. 3017 Visual Data Exploration and Analysis IV*, pages 206–217, February 1997. 11

96. Kenji Ono, Hideki Matsumoto, and Ryutaro Himeno. Visualization of thermal flows in an automotive cabin with volume rendering method. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 301–308, Wien, Austria, May 28–30 2001. Springer-Verlag. 5

97. Hans-Georg Pagendarm, B. Henne, and M. Rutten. Detecting vortical phenomena in vector data by medium-scale correlation. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 409–412, San Francisco, 1999. IEEE. 15

98. Ronald Peikert and Martin Roth. The parallel vectors operator - A vector field visualization primitive. In David S. Ebert, M. Gross, and Bernd Hamann, editors, *Procceedings of IEEE Visualization*, pages 263–270. IEEE Computer Society Press, October 1999. 14, 16

99. Frits H. Post and T. van Walsum. Fluid flow visualization. In H. Hagen, H. Müller, and Gregory M. Nielson, editors, *Focus on Scientific Visualization*, pages 1–40. Springer, 1993. 16

100. Freek Reinders, Melwin E. D. Jacobson, and Frits H. Post.

Skeleton graph generation for feature shape description. In Wim de Leeuw and Robert van Liere, editors, *Data Visualization 2000*, Eurographics, pages 73–82. Springer-Verlag Wien, May 2000. 15

101. Freek Reinders, Frits H. Post, and Hans J. W. Spoelder. Feature extraction from pioneer venus OCPP data. In Wilfred Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, Eurographics, pages 85–94. Springer-Verlag Wien New York, 1997. 13
102. Freek Reinders, Frits H. Post, and Hans J. W. Spoelder. Attribute-based feature tracking. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 63–72. Springer-Verlag Wien, May 1999. 15
103. Freek Reinders, Hans J. W. Spoelder, and Frits H. Post. Experiments on the accuracy of feature extraction. In Dirk Bartz, editor, *Visualization in Sientific Computing '98*, Eurographics, pages 49–58. Springer-Verlag Wien New York, 1998. 14
104. Christof Rezk-Salama, Peter Hastreiter, Christian Teitzel, and Thomas Ertl. Interactive exploration of volume line integral convolution based on 3D-texture mapping. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 233–240, San Francisco, 1999. IEEE. 11
105. Martin Roth and Ronald Peikert. Flow visualization for turbomachinery design. In *Proceedings of IEEE Visualization*, pages 381–384. IEEE CS Press, October 1996. 15
106. Martin Roth and Ronald Peikert. A higher-order method for finding vortex core lines. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 143–150. IEEE, 1998. 15
107. Stefan Röttger, Martin Kraus, and Thomas Ertl. Hardware-accelerated volume and isosurface rendering based on cell-projection. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 109–116. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 5
108. I. Ari Sadarjoen, Alex J. de Boer, Frits H. Post, and Arthur E. Mynett. Particle tracing in σ-transformed grids using tetrahedral 6-decomposition. In Dirk Bartz, editor, *Visualization in Scientific Computing '98*, Eurographics, pages 71–80. Springer-Verlag Wien New York, 1998. 8
109. I. Ari Sadarjoen and Frits H. Post. Geometric methods for vortex extraction. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 53–62. Springer-Verlag Wien, May 1999. 13, 14, 15
110. I. Ari Sadarjoen and Frits H. Post. Detection, Quantification, and Tracking of Vortices using Streamline Geometry. *Computers and Graphics*, 24(3):333–341, June 2000. 14
111. I. Ari Sadarjoen, Frits H. Post, Bing Ma, David C. Banks, and Hans-Georg Pagendarm. Selective visualization of vortices in hydrodynamic flows. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 419–422. IEEE, 1998. 14
112. Andrea Sanna, Bartolomeo Montrucchio, and R. Arinaz. Visualizing unsteady flows by adaptive streaklines. In *WSCG 2000 Conference Proceedings*, 2000. 10
113. Andrea Sanna, Bartolomeo Montrucchio, and Paolo Montuschi. A survey on visualization of vector fields by texture-based methods. *Research Developments in Pattern Recognition*, 1(1), 2000. publisher: www.transworldresearch.com. 9
114. Gerik Scheuermann, Holger Burbach, and Hans Hagen. Visualizing planar vector fields with normal component using line integral convolution. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 255–262, San Francisco, 1999. IEEE. 11
115. Gerik Scheuermann, Hans Hagen, Heinz Krüger, Martin Menzel, and Alyn Rockwood. Visualization of higher order singularities in vector fields. In *IEEE Visualization '97*, October 1997. 13, 14
116. Gerik Scheuermann, Heinz Krüger, Martin Menzel, and Alyn P. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, April – June 1998. ISSN 1077-2626. 13
117. Gerik Scheuermann, Xavier Tricoche, and Hans Hagen. C1-interpolation for vector field topology visualization. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 271–278, San Francisco, 1999. IEEE. 13
118. William Schroeder, Christopher R. Volpe, and William E. Lorensen. The stream polygon: A technique for 3D vector field visualization. In *Visualization '91*, pages 126–132, 1991. 8
119. William J. Schroeder, Kenneth M. Martin, and William E. Lorensen. *The Visualization Toolkit.* Prentice-Hall, Upper Saddle River, NJ 07458, USA, second edition, 1998. With special contributors Lisa Sobierajski Avila, Rick Avila, and C. Charles Law. Includes CD-ROM with vtk-2.0. 4, 6
120. Martin Schulz, Frank Recks, Wolf Bartelheimer, and Thomas Ertl. Interactive visualization of fluid dynamics simulations in locally refined cartesian grids. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 413–416, San Francisco, 1999. IEEE. 4
121. Greg Schussman, Kwan-Liu Ma, David Schissel, and Todd Evans. Visualizing DIII-D Tokamak Magnetic Field Lines. In Robert Van Liere, editor, *Proceedings of IEEE Visualization 2000 (Case Studies)*, pages 501–504, Salt Lake City, Utah, October 9-13 2000. 16
122. Han-Wei Shen, Christopher R. Johnson, and Kwan-Liu Ma. Visualizing vector fields using line integral convolution and dye advection. In *1996 Volume Visualization Symposium*, pages 63–70. IEEE, October 1996. ISBN 0-89791-741-3. 9
123. Han-Wei Shen and David L. Kao. UFLIC: A line integral convolution algorithm for visualizing unsteady flows. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization '97*, pages 317–323. IEEE, 1997. 11
124. Han-Wei Shen and David L. Kao. A new line integral convolution algorithm for visualizing time-varying flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 4(2), April – June 1998. ISSN 1077-2626. 11

125. Peter Shirley and Allan Tuchman. A polygonal approximation to direct scalar volume rendering. In *Computer Graphics (San Diego Workshop on Volume Visualization)*, volume 24, pages 63–70, November 1990. 9

126. Deborah Silver and Xin Wang. Tracking and Visualizing Turbulent 3D Features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, April 1997. 15

127. Deborah Silver and Xin Wang. Tracking features in unstructured datasets. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 79–86. IEEE, 1998. 15

128. Detlev Stalling. LIC on surfaces. In *Texture Synthesis with Line Integral Convolution*, pages 51–64. Siggraph '97, Int. Conf. Computer Graphics and Interactive Techniques, 1997. 10

129. Detlev Stalling and Hans-Christian Hege. Fast and resolution independent line integral convolution. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 249–256. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995. 9, 16

130. J. Edward Swan, Marco Lanzagorta, Doug Maxwell, Eddy Kou, Jeff Uhlmann, Wendell Anderson, Haw-Jye Shyu, and William Smith. A computational steering system for studying microwave interactions with missile bodies. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 441–444. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 5

131. Yoshiko Tamura, Kazuyuki Fujii, and Toshiyuki Ogawa. Parallel computations and visualizations of flows around high-speed trains. In Anonymous, editor, *Mechatronics and supercomputing applications in the transportation industries: ISATA International Symposium on Automotive Technology and Automation (27th: 1994: Aachen, Germany)*, pages 771–778, Croydon, UK, 1994. Automotive Automation Limited. 3

132. Chi-Keung Tang and Gérard G. Medioni. Extremal feature extraction from 3D vector and noisy scalar fields. In David S. Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 95–102. IEEE, 1998. 5, 15

133. Christian Teitzel and Thomas Ertl. New approaches for particle tracing on sparse grids. In Eduard Gröller, Helwig Löffelmann, and William Ribarsky, editors, *Data Visualization '99*, Eurographics, pages 73–84. Springer-Verlag Wien, May 1999. 8

134. Christian Teitzel, R. Grosso, and Thomas Ertl. Line integral convolution on triangulated surfaces. In *Proceedings of the Fifth International Conference in Central Europe on Computer Graphics and Visualization '97*, number 8, pages 572–581, 1997. erscheint in: Proc. WSCG '97. 11

135. Christian Teitzel, Roberto Grosso, and Thomas Ertl. Efficient and reliable integration methods for particle tracing in unsteady flows on discrete meshes. In Wilfred Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, Eurographics, pages 31–42. Springer-Verlag Wien New York, 1997. 8, 16

136. Christian Teitzel, Roberto Grosso, and Thomas Ertl. Particle tracing on sparse grids. In Dirk Bartz, editor, *Visualization in Sientific Computing '98*, Eurographics, pages 81–90. Springer-Verlag Wien New York, 1998. 8, 16

137. Alexandru Telea and Jarke J. van Wijk. Simplified representation of vector fields. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 35–42, San Francisco, 1999. IEEE. 10, 12, 13

138. Lloys A. Treinish. Multi-resolution visualization techniques for nested weather models. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *Proceedings Visualization 2000*, pages 513–516. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 5

139. Xavier Tricoche, Gerik Scheuermann, and Hans Hagen. A topology simplification method for 2D vector fields. In *Proc. of the 11th Ann. IEEE Visualization Conference (Vis) 2000*, 2000. 13

140. Xavier Tricoche, Gerik Scheuermann, Hans Hagen, and Stefan Clauss. Vector and tensor field topology simplification on irregular grids. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 107–116, Wien, Austria, May 28–30 2001. Springer-Verlag. 13

141. Greg Turk and David Banks. Image-guided streamline placement. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 453–460. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996. 6, 7

142. Shyh-Kuang Ueng, Christopher Sikorski, and Kwan-Liu Ma. Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):100–110, June 1996. 8

143. Shyh-Kuang Ueng, Christopher Sikorski, and Kwan-Liu Ma. Out-of-Core Streamline Visualization on Large Unstructured Meshes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):370–?, October 1997. 3

144. Milton van Dyke. *An Album of Fluid Motion*. The Parabolic Press, 1982. 16

145. Theo van Walsum, Frits H. Post, Deborah Silver, and Frank J. Post. Feature Extraction and Iconic Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):111–119, June 1996. 15

146. Jarke J. van Wijk. Spot noise-texture synthesis for data visualization. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 309–318, July 1991. 9

147. Jarke J. van Wijk. Flow visualization with surface particles. *IEEE Computer Graphics and Applications*, 13(4):18–24, July 1993. 8, 16

148. Vivek Verma, David Kao, and Alex Pang. PLIC: Bridging the gap between streamlines and LIC. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of the 1999 IEEE Conference on Visualization (VIS-99)*, pages 341–348, N.Y., October 25–29 1999. ACM Press. 11

149. Vivek Verma, David Kao, and Alex Pang. A flow-guided streamline seeding strategy. In *Proceedings Visualization 2000*, pages 163–170. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 6

150. Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualising information nets in three dimensions. *ACM Transactions on Graphics*, 1995. 3

151. Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, and Kenneth I. Joy. Extraction of crack-free isosurfaces from adaptive mesh refinement data. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 25–34, Wien, Austria, May 28–30 2001. Springer-Verlag. 5

152. R. Wegenkittl, Eduard Gröller, and Werner Purgathofer. Animating flow fields: Rendering of oriented line integral convolution. In *Computer Animation '97 Proceedings*, pages 15–21. IEEE Computer Society, June 1997. 10

153. Rainer Wegenkittl and Eduard Gröller. Fast oriented line integral convolution for vector field visualization via the Internet. In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 309–316, Los Alamitos, October 19–24 1997. IEEE Computer Society Press. 10

154. Rainer Wegenkittl, Eduard Gröller, and Werner Purgathofer. Visualizing the dynamical behavior of Wonderland. *IEEE Computer Graphics and Applications*, 17(6):71–79, November/December 1997. 7

155. David M. Weinstein, Gordon L. Kindlmann, and Eric C. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 249–254, San Francisco, 1999. IEEE. 17

156. Rüdiger Westermann. The rendering of unstructured grids revisited. In David S. Ebert, Jean M. Favre, and Ronald Peikert, editors, *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 65–74, Wien, Austria, May 28–30 2001. Springer-Verlag. 4, 5

157. Rüdiger Westermann, Christopher Johnson, and Thomas Ertl. Topology-preserving smoothing of vector fields. In *IEEE Transactions on Visualization and Computer Graphics*, volume 7(3), pages 222–229. IEEE Computer Society, 2001. 8

158. Craig M. Wittenbrink, Alex T. Pang, and Suresh K. Lodha. Glyphs for Visualizing Uncertainty in Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, September 1996. 17

159. Pak Chung Wong, Harlan Foote, Ruby Leung, Elizabeth Jurrus, Dan Adams, and Jim Thomas. Vector fields simplification - a case study of visualizing climate modeling and simulation data sets. In *Proc. of the 11th Ann. IEEE Visualization Conference (Vis) 2000*, 2000. 14

160. Malte Zöckler, Detlev Stalling, and Hans-Christian Hege. Interactive visualization of 3D-vector fields using illuminated streamlines. In *Proceedings of IEEE Visualization '96, San Francisco*, pages 107–113, October 1996. 7

161. Malte Zöckler, Detlev Stalling, and Hans-Christian Hege. Parallel line integral convolution. *Parallel Computing*, 23(7):975–989, July 1997. 3, 9