Paper A

Illustrative Flow Visualization: State of the Art, Trends and Challenges

Andrea Brambilla[1], Robert Carnecky[2], Ronald Peikert[2], Ivan Viola[1], Helwig Hauser[1]

[1] Department of Informatics, University of Bergen, Norway

[2] Department of Computer Science, ETH Zurich, Switzerland

Abstract

Flow visualization is a well established branch of scientific visualization and it currently represents an invaluable resource to many fields, like automotive design, meteorology and medical imaging. Thanks to the capabilities of modern hardware, flow datasets are increasing in size and complexity, and traditional flow visualization techniques need to be updated and improved in order to deal with the upcoming challenges. A fairly recent trend to enhance the expressiveness of scientific visualization is to produce depictions of physical phenomena taking inspiration from traditional handcrafted illustrations: this approach is known as *illustrative visualization*, and it is getting a foothold in flow visualization as well.

In this state of the art report we give an overview of the existing illustrative techniques for flow visualization, we highlight which problems have been solved and which issues still need further investigation, and, finally, we provide remarks and insights on the current trends in illustrative flow visualization.

This article was published in the proceedings of *Eurographics 2012-State of the Art Reports*, 2012.

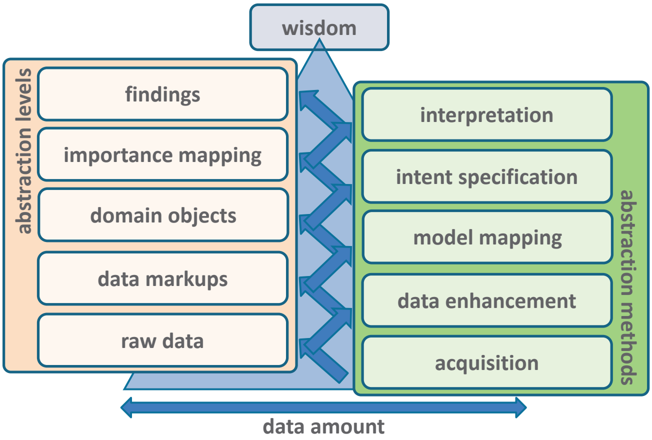


Figure 1: The data abstraction pyramid.

1 Introduction

Van Dyke's book [VD82] from 1982 begins with the following statement: "We who work in fluid mechanics are fortunate [. . .] that our subject is easily visualized". This is indeed reflected by the many years of successful research in flow visualization: with the help of visualization techniques, flow phenomena have been deeply studied and many unclear aspects of their behaviour have been explained. Over the years, this continuous investigation process have produced a considerable amount of knowledge and, in the meantime, the computational power of the hardware has been growing exponentially. Nowadays we are able to produce, through measurements or simulations, extremely faithful and high quality flow datasets, which are usually very dense, multidimensional and multivariate. It is, therefore, almost impossible to get any insight out of them without the help of automatic or semi-automatic tools.

The analysis/postprocessing phase can be more or less complex and, based on several years of expertise and research in visualization, we propose to describe it through the *data abstraction pyramid* metaphor, in Figure 1. At the lowest level, an acquisition step produces the so called *raw data*, which is an initial representation of the phenomenon of interest. At this point different processing steps can be taken: gradients and local properties can be computed in order to enrich the data, a domain-specific model can help identifying relevant feature, and so on. After every step a more abstract representation of the underlying phenomenon is obtained. The purpose of visualization techniques is to take data at a certain abstraction level and show it in a way that allow users to gain insights out of it.

Traditional flow visualization techniques have been quite effective in making flow data understandable, but they struggle to deal with the increased complexity of the most recent datasets. A novel category of visualization approaches, that has already been successful in medical [SES05, VKG05, TSS+06] and other visualization subfields [WBE+06, HBP+07, PGT+08], is *illustrative visualization*. This discipline aims at visualizing the data in a clear and understandable way through the use of techniques from traditional handcrafted illustrations. Illustrative visualization techniques explicitly address issues like cluttering, occlusion and depth perception, which are typical for flow visualization as well. Exploiting illustrative approaches in this field allows for quick exploration and in-depth analysis of dense flow datasets, consequently producing a significant amount of knowledge which would be otherwise unattainable.

The rest of the paper is organized as follows: brief overviews of the basics and most common approaches in both flow and illustrative visualization are given in Section 2 and Section 3, respectively. Section 4 is dedicated to the classification and description of currently existing illustrative flow visualization approaches, and, finally, Section 6 summarizes the present state of the art and suggests directions for possible future developments.

Contributions

Illustrative flow visualization is a newborn discipline and, as such, it still lacks of a formal structural organization and well defined boundaries. In light of this consideration, the main contributions and novelties of this STAR can be summarized as follows:

- For the first time illustrative flow visualization is thoroughly analysed and formally organised.
- We propose a user-centric classification of the techniques in this field, in order to help application experts (our users) to choose the ones that best suit their needs.
- In the context of this classification, we review the existing approaches and the most recent developments in the field.
- We give an overview of illustrative visualization focused on showing the advantages of this category of techniques over traditional visualization.

2 Traditional Flow Visualization

The term *flow* denotes an abstract concept adopted in many application fields. Fluid dynamics, for instance, is concerned with the study of fluid flows, i.e. the motion of fluids: typical examples include the motion of water in a pump or a turbine, the stream of air around a car or an airplane, blood in a vessel, oil or gas in a pipe, and so on. However, the concept of flow is much broader and different definitions arise in every area of application, such as physics or mathematics. Flow visualization usually deals with data generated via measurements, simulations or modeling, and the results are commonly expressed as *vector fields*. In the following, the formal mathematical background is discussed, then an overview of flow visualization is presented, focusing on what are the existing techniques, how they can be classified, and which are the pressing challenges.

2.1 Flow and vector fields

Firstly it is worth pointing out that the mathematical theories behind flows and vector fields are extensive and beyond the scope of this paper, here only a brief overview is given; for a more detailed introduction on the subject, the reader can refer to Asimov's tutorial from 1993 [Asi93].

Given a dense set of massless particles $i = 0, 1, 2, \ldots$ moving in the spatial domain $\Omega \subseteq \mathbb{R}^n$, an n-dimensional *steady flow* $\mathbf{v}$ is typically described with a differential equation of the particles locations $\mathbf{x}_i \in \Omega$ with respect to the time $t \in \mathbb{R}$:

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}(\mathbf{x}_i(t)) \tag{1}$$

In other words, a steady flow is associated with a vector field that describes the instantaneous velocities of the particles moving in the Euclidean space $\mathbb{R}^n$:

$$\mathbf{v} : \Omega \to \mathbb{R}^n \tag{2}$$

The term *steady* means that the velocity vectors are constant over time; in contrast an *unsteady* vector field is time-dependent and is defined as:

$$\mathbf{v} : \Omega \times \mathbb{R} \to \mathbb{R}^n \tag{3}$$

The related differential equation describing the motion of the particles is

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}(\mathbf{x}_i(t), t) \tag{4}$$

The differential equations 1 and 4 are solved via integration. In particular, a *streamline* is obtained by integration on a steady vector field and it represents the path of a particle in the steady flow; a *pathline* is the equivalent for unsteady flows. There are two other types of commonly used curves obtained through integration: a *streakline* is the imaginary line created by particles continuously seeded at a certain position, and it is closely related to the physical experiment of releasing dye into a fluid. A *timeline* instead is obtained by integrating a set of particles simultaneously released in the flow along a certain line or curve. Stream-, path-, streak- and timelines are commonly known as *integral lines* (or *integral* *curves*), and they can be extended to higher dimension (-surfaces, -volumes). In the following, integral lines, integral surfaces and integral volumes are referred to as *integral structures*.

Vector fields and integral structures are both well defined concepts but their expressiveness is limited; essentially, the resulting visualizations may be not able to fully represent certain relevant aspects of the flow behaviour. The most recurrent strategy to address this issue is to look for objects of particular interest in the vector field, the so called *flow features*. In accordance with the state of the art report on feature-based flow visualization by Post et al. [PVH+03], the most common types of flow features include:

Vortices: areas associated with swirling motion, they are almost ubiquitous in flows and are of vital importance in many applications. The main problem is that a formal, well accepted definition of vortex has yet to be found.

Vector field topology (VFT): obtained by integrating selected streamlines close to critical points (i.e. points $\mathbf{p} \in \mathbb{R}^n$ where $\mathbf{v}(\mathbf{p}) = 0$), usually referred to as separatrices. They partition the vector field in areas that asymptotically show coherent behavior. VFT is very effective for 2D steady flows, but its extension to 3D is problematic because of cluttering and occlusion issues. More details on this subject can be found in the works of Helman and Hesselink [HH89, HH91] or in the survey from Laramee et al. [LHZP07].

Lagrangian Coherent Structures (LCS): from a conceptual point of view, they are an attempt to extend VFT to unsteady flows. A formal definition has not yet been given and the research on this subtopic is very active. Useful information can be found in related literature [Hal01, SLM05, PPF+11].

Shock waves: typical of flows around aircraft, they are characterized by sharp discontinuities in physical flow attributes. A straightforward way to detect them is to look for edges in scalar quantities such as pressure, density or velocity magnitude.

Separation and attachment lines: only present in conjunction with solid bodies or boundaries, these are the curves where the flow abruptly moves away (separation) or towards (attachment) the surface of the solid object.

2.2 Vector field discretization

Focusing now on a more practical topic, flow visualization approaches have to take into account a specific issue: data obtained from simulations or measurements is almost never given in analytic form, but sampled at specific locations in the spatial and temporal domain. The set of sampling points is usually topologically organized according to a more or less structured *grid* (also called *mesh*), ranging from Cartesian to curvilinear, or even to completely unstructured grids.

Even assuming that the Nyquist frequency condition is fulfilled, a series of problems arises: for example, *point location*, i.e. determining the grid cell which contains a certain point, is not trivial, especially for unstructured grids. Special data structures are often employed in order to speed up the search. Moreover, since the vector field is sampled only at specific points, there is also the need for a *reconstruction* strategy to determine flow attributes at generic locations of the domain. Reconstruction algorithms heavily depend on the cell shape and are usually based on linear or higher order interpolation techniques. The reconstruction problem can also cause collateral effects on the computation of derivatives and integral structures, therefore special attention is often needed when designing these algorithms. A more detailed overview of sampling grids and related issues can be found in previous state of the art reports [PVH$^+$02, LHD$^+$04].

For the sake of completeness, it is worth mentioning that, even though sampling grids are widely used, other types of representations exist, like particle-based or functional representations, and each one has its own set of related challenges.

2.3 Flow visualization techniques

One of the first attempts to formally classify and summarize decades of work in flow visualization has been proposed by Hesselink, Post and van Wijk in 1994 [HPvW94]: they suggest to differentiate existing approaches according to the type of data (scalar, vector, tensor), the dimensionality of the domain (point, line, surface, volume) and the "information level", i.e. if the displayed data is either raw (elementary), derived from a small neighbourhood (local), or dependent on the entire dataset (global). The first two criteria are still widespread nowadays, with the second one now taking into account the temporal dimension as well, and they are often combined with other classification directives.

In 2002, Post et al. [PVH$^+$02] proposed one of the most widely accepted categorizations of flow visualization techniques:

Direct Visualization: the data is directly mapped to a visual representation, without complex conversions or extraction steps. Arrow glyphs, color coding and volume rendering are the core of this category.

Texture-based Visualization: a dense representation of the flow is obtained using local flow attributes to create and/or warp a noise texture; more details on this topic can be found in [SMM00] and [LHD$^+$04].

Geometric Visualization: in order to better convey flow dynamics, integral structures are used as a basis for the graphical representation; a recent survey by McLoughlin et al. [MLP$^+$10] thoroughly describes this category of approaches.

Feature-based Visualization: a sparse visualization is obtained focusing only on the most significant areas of the vector field; a comprehensive survey on features extraction and related visualization techniques has been presented in 2003 by Post et al. [PVH$^+$03].

More recently, Salzbrunn et al. [SJWS08] propose to add a new category, i.e. **Partition-based Visualization**, which includes all those approaches aimed at effectively partitioning the spatial and temporal domain according to flow properties.

For reasons that will become clear in Section 4 we do not fully adopt this classification, but we adjust it in order to better reflect the data abstraction scheme previously introduced through the pyramid metaphor.

2.4 Challenges in flow visualization

Flow visualization has been an active research field for many years now and satisfactory solutions have been found for many problems, like the direct or texture-based visualization of 2D steady flows. However, a lot of questions still have to be answered. This topic is extensively discussed in the survey by McLoughlin et al. [MLP$^+$10], here we provide a just short list of selected challenges.

Already at the raw data level interesting research opportunities can be found: first of all, the generation of a dataset, through simulations or measurements, can take days and generate terabytes of data; in contrast, the visualization process has to be possibly interactive and, since disk access is a time-consuming operation, it can only rely on a few gigabytes of main memory.

Furthermore, data is usually defined according to a certain grid, as mentioned before, and the type of the grid has a significant impact on the visualization as well: for instance, raycasting through a Cartesian grid is straightforward, but it becomes progressively more complex with less structured grids. In general, approaches designed for a certain type of grid structure are not guaranteed to work well with more complex ones, at least not at the same frame rate.

From a visualization-related point of view, one of the most difficult challenges is associated with three-dimensional datasets: vector fields are usually very densely sampled, therefore cluttering and occlusion are almost ubiquitous. In 3D, integral structures and certain flow features, like LCS, often present twists, folds and self intersections, which make depth and shape perception very difficult.

Adding the temporal dimension makes the situation even more complicated: animation is the traditional tool to depict time-dependent information, but it can only depict one instant at a time, the temporal context is limited; on the other hand, pathlines and streaklines can be used to show the trajectory of one or more particles, but the global flow behavior cannot be effectively conveyed. In the next section we discuss how illustrative visual abstractions can be used to solve some of these open issues.

3 The Illustrative Paradigm

Illustrative visualization is an emerging branch of the visualization research field that focuses on interactive and expressive visualizations typically inspired by works from artists and illustrators [RBGV08]. Its main goal is maximizing the amount of information effectively conveyed utilizing visual abstraction techniques.

In traditional craft, the illustrator employs drawing styles such as pencil, brush, or watercolor styles; in illustrative visualization, algorithms that are concerned with visual styles are referred to as *low-level* visual abstractions [VGH+05]. Line drawings techniques, contours or silhouettes [IFH+03], and handcrafted shading, such as stippling, hatching, or toon shading [GGSC98], provide enhanced shape, depth and directional cues in order to improve the perceptual effectiveness of the results. Low-level visual abstractions, such as those mentioned above, define how to depict a certain structure and have been the primary focus of *non-photorealistic rendering* (NPR).

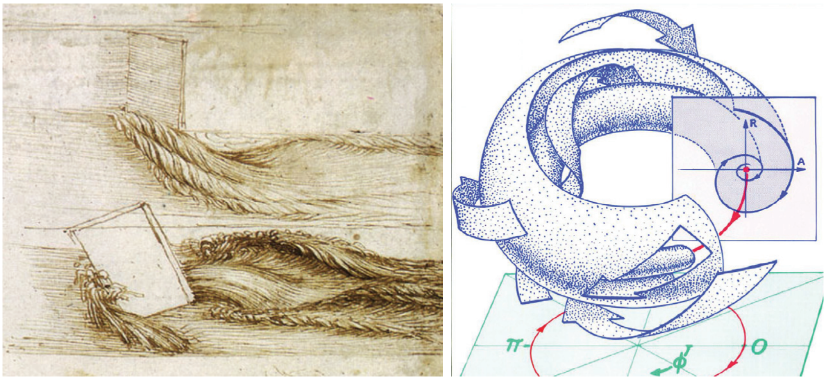
When dealing with large and dense amounts of data, illustrators work with expressive techniques that change the layout or deform features to increase the communicative intent of the illustration; these approaches are commonly referred to as *high-level* visual abstractions. Selective visualization, cutaways, close-ups, or exploded views are examples of illustrative concepts that can be simulated with computerized techniques with different purposes in mind. In particular, *visibility management* (also known as *smart visibility*) techniques [VG05, ET08] are aimed at improving the overall visibility of the data through an optimal use of the visual space. In contrast, *focus emphasis* (or *focus+context*) approaches [Hau03, VKG05] acknowledge that portions of the data are deemed more important than others. The focus, i.e. the relevant part of the dataset, has to be visually emphasized, while less important information should be used to provide the context. The mapping between domain knowledge and visual appearance is expressed by an importance measure.

Closely related to high-level visual abstractions, are guided visualization [KBKG07, VFSG06] and interactive visual storytelling techniques [WH07, MLF+12]. The former guides the viewer's attention to the relevant structures by computing informative viewpoints and camera paths for refocusing from one object of interest to another. Interactive storytelling enables the user to set up a story related to a phenomenon of interest by setting up story-nodes and transitions between them. All the aspects of the story, from the rendering style to the camera parameters, can be interactively modified, so every user can adapt the narrative process to his or her own needs on the fly. Both these categories of approaches share the common goal of providing an effective visual description of the phenomenon of interest, therefore we refer to them as *visual explanation* techniques.

Notice that different amounts of domain knowledge are needed in every cate-

Visual Abstraction	Advantages over traditional visualization	Knowledge about the data
Perceptual Enhancement	Depth and/or shape perception is enhanced, local properties of the data are effectively conveyed	No domain knowledge about the data is usually taken into account
Visibility Management	Occlusion and cluttering are reduced, the expressiveness of the visual space is maximized	The data reflects the inherent complexity of the phenomenon of interest
Focus Emphasis	The data aspects in focus are always clearly visible, well depicted and not occluded by the context	A portion of the data (the focus) is deemed more important than the rest
Visual Explanation	A description of the phenomenon is presented to the user according to his or her needs	A domain-specific semantic is associated with the data

Table 1: The illustrative visual abstraction categories, with their advantages over traditional visualization and the kind of knowledge about the data they take into account.



(a) (b)

Figure 2: (a) Hand-drawn illustration of water flow behind an obstacle by Leonardo da Vinci [dV09]. (b) Depiction of a dynamical system with stream arrows (image courtesy of Abraham and Shaw [AS82] ©1982 Addison-Wesley).

gory in order to achieve expressive visualizations: this property is nicely aligned with the idea of knowledge and information-assisted visualization expressed by Chen et al. [CEH$^+$09]. Table 1 summarizes the illustrative visualization categories just introduced, emphasizing what are their strong points with respect to traditional visualization, and what kind of knowledge about the data they take into account. For more details about illustrative visualization and a review of the best known techniques, the reader can refer to [VGH$^+$05, SEV$^+$08].

Before proceeding to the next section, we would like to spend a few words about *interactivity*: besides solving occlusion problems, interactive navigation through the spatial domain is one of the most effective ways to perceive the location and the shape of an object. Its central role has been already identified 30 years ago [HS89], and it is still one of the most sought after features of a visualization system. Even though expressiveness is frequently much more discussed, illustrative visualization heavily relies on interactivity as well.

4 Illustrative Flow Visualization

The idea of using illustrations to depict and investigate flow behavior is not new, on the contrary, it has been around for more or less 500 years: Figure 2a is an illustration from Leonardo da Vinci (1452-1519) showing the motion of water behind a solid obstacle. More recently (1982), Abraham and Shaw extensively used hand-drawn pictures to visualize flow structures in their book [AS82] (see Figure 2b). Neither da Vinci nor Abraham and Shaw had access to fancy graphics hardware, but they were still able to effectively convey relevant flow information; so, how did they do it?

Analyzing their pictures, some of the concepts introduced in the previous section can be easily recognised: hatching and stippling are used to improve depth perception, only portions of the data are shown so that cluttering is avoided, cut-aways and clip planes effectively improve visibility and reduce occlusions. Nowadays the quality and the amount of available computational resources are respectively better and larger than 30 (or 500) years ago, but those concepts and guidelines are still the key to produce intuitive, effective and aesthetically pleasing visualizations.

4.1 A user-centric classification

Being at the crossroad of two disciplines, it seems natural to classify illustrative flow visualization techniques according to two different characteristics, one from the flow and one from the illustrative domain. While designing this categorization we have taken into account the following guidelines:

1. the two classification criteria should be as independent as possible;
2. the advantages of using illustrative visualization, as compared to more traditional visualization, should be clearly emphasized;
3. the classification should help a potential user (with possibly limited knowledge of visualization) to find which techniques suit his/her needs best.

Regarding the flow perspective, we realized that the traditional subdivision into direct, texture-based, geometric and feature-based visualization is not optimal since, even though perfectly clear to a visualization expert, it does not well reflect the point of view of a user. Doctors, engineers and meteorologists need a clear depiction of some specific aspects of the data, the visual tools employed are only partially relevant to them. On the other hand, they are usually well aware of what kind of data they are dealing with and which processing steps are meaningful for their applications, therefore we propose a slightly different classification based on what are the domain objects the user wants to see in a visualization:

Raw data refers to the original data produced by simulations or measurements (such as velocity or pressure), together with the information directly derivable from it (like curl or gradients); this kind of data is usually defined at the vertices or cells of the sampling grid and it can be easily visualized, for instance, via volume rendering or glyphs.

Integral structures are well known concepts in the flow community and are extensively used to investigate flow behavior; they are usually displayed as linear structures (lines, tubes, ribbons) or surfaces.

	DATA ABSTRACTION →		
	Raw Data	*Integral Structures*	*Flow Features*
Perceptual Effectiveness	[vW91] [CL93] [IG97] [dLvL97] [HWSE99] [KML99] [RSHTE99] [DPR00] [ES01] [JEH01] [WFK^{+}02] [SLM02] [vW02] [WEE03] [PBL^{+}04] [SJEG05] [SET^{+}06]	[SVL91] [USM96] [ZSH96] [SM04] [SKH^{+}04] [XZC04] [MPSS05] [SGS05] [vFWTS08] [BFTW09] [KGJ09] [EBRI09]	
Visibility Management	[LKG97] [UIM^{+}03] [UIL^{+}04] [vPBB^{+}10] [BVMG08]	[MCG94] [LMG97] [LDG98] [HWHJ99] [TVW99] [GPR^{+}01] [JL97] [GPR^{+}04] [LS07] [YWM07] [LHS08] [BWF^{+}10] [HGH^{+}10] [MCHM10] [CYY^{+}11] [LMSC11] [vPGtHRV11] [WTS11]	[RPS01] [HMCM10]
Focus Emphasis	[SJM96] [TvW03] [HM03] [SLB04] [WE04] [WBE05] [CSC07] [WSE07] [FW08] [IEGC08] [LTH08] [WYM08] [WYG^{+}11]	[FG98] [HM03] [MTHG03] [KKKW05] [WS05] [STWE07] [FBTW10] [JM10] [WWYM10]	[MM09]
Visual Explanation	[AWM10] [PTA^{+}11]	[BKKW08]	

Table 2: Classification of illustrative flow visualization approaches according to the abstraction level of the visualized data and the kind of visual abstraction they adopt. The different categories are thoroughly described in Section 4

Flow features are subsets of the data perceived as particularly relevant by the user; according to the specific application domain different definitions arise and different visualization techniques are used.

These categories on the flow axis go along very well with the illustrative visual abstraction levels presented in Section 3: there is no mutual dependency between the two classifications, and the user, in order to satisfy his or her visualization needs, has just to answer the questions "what flow representation do I want to refer to?" and "what visualization enhancement do I need?". Table 2 summarizes this classification and presents a possible categorization of existing illustrative flow visualization techniques according to the two criteria just introduced.

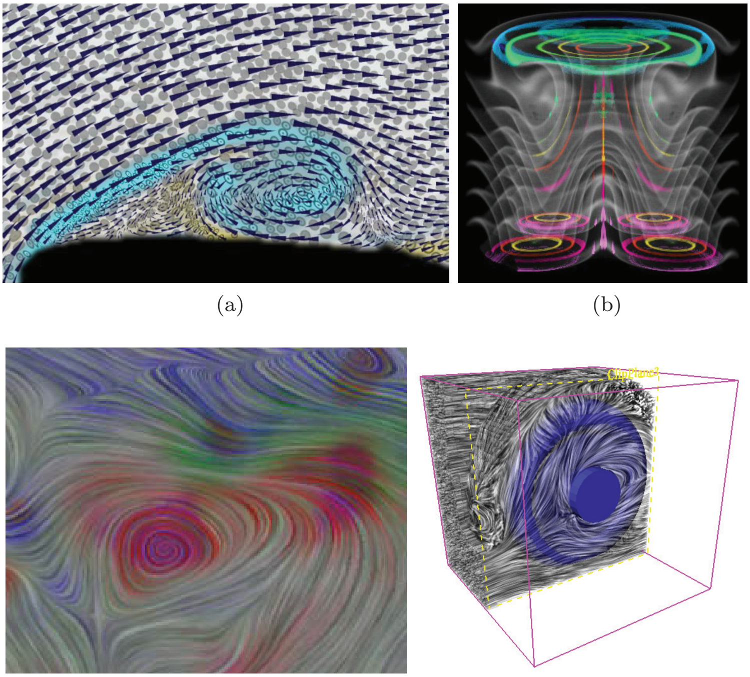
We would like to point out that there is also an alternative interpretation to our classification: the flow axis can be seen as the amount of processing that the data has been undergoing, while the illustrative categories relate to the amount of knowledge about the data taken into account during the visualization process. These are the same concepts mentioned before regarding, respectively, the abstraction pyramid (see Section 1) and the knowledge and information-assisted visualization (see Section 3).

4.2 Perceptual Effectiveness

This subsection gives an overview of the approaches focused on improving the perception of the flow data through the use of depth, shape and directional cues. This necessity has been identified in flow visualization a long time ago, here we review those approaches that clearly show some illustrative aspect, even though they have been proposed before illustrative visualization was formally introduced.

Improving perception of raw data

The techniques in this category are mostly based on well known visualization concepts, like directional glyphs (hedgehog), transfer functions or texture advection. The mapping between the data and its visual counterpart is usually very tight, and it typically leads to very dense representations of the dataset. For example, the approach proposed in [KML99] is built on a direct correspondence between flow properties and visual resources: Figure 3a has been obtained by mapping velocity direction and magnitude to direction and size of arrows, while colors represent the vorticity and ellipses represent strain, divergence and shear. A similar idea has been adopted in the work of Ebert and Shaw in 2001 [ES01], where arrows and superquadric shapes are used to convey flow properties in a three-dimensional immersive environment. Notice that these techniques essentially differ from the traditional (non-illustrative) arrow plotting techniques since particular attention is posed on the expressiveness (shape, appearance, position and so on) of the glyphs.



(a) (b)

(c) (d)

Figure 3: (a) Visualization of multiple flow attributes: arrows represent velocity, colors represent vorticity and ellipses represent strain, divergence and shear (image courtesy of Kirby et al. [KML99] ©1999 IEEE). (b) Illustrative volume rendering of flow data (image courtesy of Svakhine et al. [SJEG05] ©2005 IEEE). (c) Texture-based visualization with color-coding of local flow properties (image courtesy of Urness et al. [UIM^{+}03] ©2003 IEEE). (d) 3D-LIC of flow around a wheel, visualized with the aid of a clipping plane (image courtesy of Rezk-Salama et al. [RSHTE99] ©1999 IEEE).

The natural extension of color coding to 3D is volume rendering. This technique is known to generate cluttering and occlusion if used unwisely, therefore particular attention should be paid in the setup phase. Stompel et al. [SLM02] propose different NPR techniques for volumetric data, with the goal of improving the readability of the results. In contrast, Park et al. [PBL^{+}04] use raw and derived data as input of a customizable n-dimensional transfer function, allowing for expressive and uncluttered visualization. An in-between approach has been proposed in 2005 by Svakhine et al. [SJEG05]: only 2 variables are used to control color and transparency, therefore the long and cumbersome fine-tuning of the transfer function needed in [PBL^{+}04] is avoided. As can be seen in Figure

3b, simple illustrative techniques, like silhouette enhancing, are applied in order to improve the appearance of the results.

This last work has also been extended to tetrahedral grids in 2006 [SET$^+$06] but, since then, volume rendering of flows hasn't attracted too much attention, probably because it is not well suited for conveying directional information. On the other hand, volume rendering is an active research field on its own, and techniques developed for volumetric data are often used in flow visualization to show scalar variables like pressure or temperature.

Another category of approaches aimed at effectively conveying flow properties is *texture-based visualization*: the basic idea is to generate a noise texture and then use local characteristics of the vector field to warp or filter it. This kind of techniques is widely used in flow visualization, mainly for 2D flows or on curved surfaces in 3D, since they are able to clearly represent directional information with minimal visual resources. The direct consequence is that they can be easily combined with other techniques, like color mapping, glyphs, or even other textures, to obtain highly expressive results (see Figure 3c).

One of the first texture-based techniques, introduced by van Wijk in 1991 [vW91], is *spot noise*: a set of intensity functions (the spots) are warped over a small time step according to the velocity vectors, therefore generating a dense texture which encodes both the direction and the magnitude of the local flow. Two years later a similar technique was presented: *Line Integral Convolution* (LIC) [CL93] fetches intensity values from a random noise textures and convolves them along the streamlines of the vector field; in contrast to spot noise, LIC does not reflect the magnitude of velocity, but makes the location of critical points easier. These techniques have been later extended in many ways:

1. [dLvL97] deals with unsteady vector fields.
2. [HWSE99] proposes a GPU implementation of LIC.
3. [IG97, RSHTE99] apply LIC to 3D flows using 3D textures. In this case the results suffer of serious cluttering and occlusion problems, so halos and clipping planes are used to enhance the overall readability (see Figure 3d).

Other notable techniques in this category are based on anisotropic non-linear diffusion [DPR00], *Image Based Flow Visualization* (IBFV) [vW02], *Lagrangian-Eulerian Advection* (LEA) [JEH01], and especially *Unsteady Flow Advection-Convolution* (UFAC) [WEE03], which is able to emulate most of the previously introduced approaches. Research in texture-based flow visualization has been very active until a few years ago and is now considered an almost closed topic; for a comprehensive overview of the substantial amount of work on this topic the reader can refer to the state of the art report by Laramee at al. [LHD$^+$04].

To conclude this section, we would like to emphasize that the different techniques introduced up to now can also be used as "building blocks" of more comprehensive visualizations: [WFK$^+$02], for instance, employs textures, hue and intensity to visualize 3 different flow aspects, and mix them using partial transparency and 3D height fields.

Effective integral structures

Integral structures are widely used in flow visualization because of their inherent ability of clearly depicting the trajectories of particles in the flow, a task that cannot be achieved with raw data alone. The first attempt to improve the expressive power of integral lines was the *Stream Polygon* [SVL91], proposed in 1991 by Schroeder et al.: an n-sided polygon is swept along a streamline and it is deformed according to local flow properties, like the normal or shear strain; moreover, once the deformed polygons have been computed at every point of the streamline, they can be connected, generating a *streamtube*. This idea became very popular and different improvements and variants have been proposed: notably [USM96] presents an extension to unstructured grids, while [SKH$^+$04] describes a method similar to billboarding aimed at speeding up the rendering. More recently Stoll et al. [SGS05] introduced a novel rendering algorithm that allows to control different properties of tube-like structures and supports effects like halos, shadows and texturing to improve the visual appearance of the results (see Figure 4).

Instead of dealing with geometrical structures, another well known category of approaches focuses on shading techniques for (infinitesimally thin) lines. For example Zöckler et al. in 1996 propose a method for computing Phong illumination on streamlines, obtaining the so called *Illuminated Streamlines* [ZSH96]. Similar results are presented in [MPSS05], which reviews Zöckler's work in order to enhance depth perception, and in [SM04], which samples the lines into an anisotropic voxel representation; the voxels are then displayed via volume rendering, allowing for the visualization of whole datasets at nearly interactive rates.

A slightly different and very interesting approach has been proposed by Everts et al. [EBRI09]: they display dense bundles of lines with a simple pen-and-ink style (black and white), while depth information is effectively conveyed by a smart use of halos. In general, the main difference between geometric and shading approaches is that, while the former are able to convey

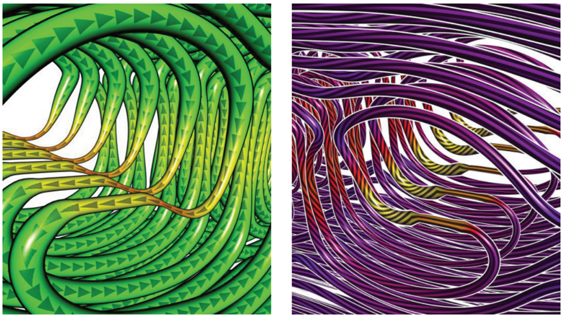


Figure 4: Different visual enhancements applied to integral lines (image courtesy of Stoll et al. [SGS05] ©2005 IEEE).

local properties of the flow, the latter can guarantee a denser coverage of the spatial domain.

Focusing now on 2D integral structures, it immediately stands out that the illustrative visualization of flow surfaces has followed a completely different path as compared to integral lines. This is actually not so surprising: even a minimal swirling motion can make an integral surface roll up, occluding itself. Moreover, in the case of pathsurfaces, self intersections occur quite frequently. The direct consequence is that "visibility" issues, discussed in the next section, have been much more investigated than "perceptual" ones. Here we present three techniques focused on the interactive illustrative visualization of time- and streaksurfaces.

As already stated at the end of Section 3, interaction is extremely effective in improving depth and shape perception, but, due to their high computational cost, visualizing time- and streaksurfaces at interactive rates has been a difficult challenge. The approach by von Funck et al. [vFWTS08] consists of a rendering technique that gives surfaces a smoke-like appearance. Besides the visually pleasant look, the main advantage is that the algorithm for surface construction can be largely simplified, since the resulting artifacts are not shown by the smoke-like rendering. In contrast, the other two approaches [BFTW09, KGJ09] explicitly review the surface construction procedure, computing it on the GPU and employing different optimizations and workarounds. They also apply different illustrative techniques, like transparency, silhouette enhancement and ribbon-like textures, while still achieving interactive frame rates.

Integral volumes were introduced in 1993 by Max et al [MBC93], but they have never received too much attention, probably because their inherent complexity is not matched by a significant improvement of expressiveness. However, it is worth mentioning the approach by Xue et al. [XZC04], which computes streamvolumes and visually enhances them with texture advection techniques.

Appearance of flow features

To our knowledge, there is no technique which explicitly addresses the perceptual effectiveness of flow feature visualizations. A possible reason could be that many kinds of features, like vortices or shock waves, can be mapped to sparse and easy to understand visual representations, therefore they rarely present perceptual problems. This situation may soon change as a consequence of the recent increasing interest in Lagrangian Coherent Structures: LCS are typically represented by very complex surfaces, which can be self-intersecting, self-occluding, or even non-manifold. As this kind of features will grow in popularity, we expect that different approaches will be developed in order to visualize them in an effective way.

4.3 Visibility Management

Visibility management includes all those approaches that explicitly address visibility, occlusion and cluttering issues. It is important to point out that, at this abstraction level, only the visual appearance of the result is taken into account. The fact that some portions of the data can be more relevant than others is discussed later, in the "Focus Emphasis" section. Notice that in this category, a particular class of illustrative techniques can be identified, i.e. *temporal implosion*. Since this idea has been applied to all the three kinds of flow entities, we discuss it in a separate subsection.

Raw data visibility

Raw data is usually dense and this entails different consequences depending on the dimensionality of the dataset. In 2D the whole data can be easily displayed on a plane, therefore visibility issues are minimal. It is however worth mentioning the *Color Weaving* technique by Urness et al. [UIM$^+$03]. In their work a texture-based visualization conveys directional information, while multiple color maps are used to represent different local properties of the flow. Thanks to a smart interleaving algorithm, color mixing is avoided and many attributes can be visualized at once: in Figure 3c red and blue denote areas of respectively positive and negative vorticity, green represents the shear stress, while orange and magenta highlight swirling regions. The expressiveness of this technique has been further improved one year later [UIL$^+$04], adding shading effects based on contrast, luminance and embossing.

In 3D the situation is much more complex: the degree of occlusion is extremely elevated and visualizing the data as a whole is, at least, highly challenging. A quite typical approach is clipping portions of the dataset or showing only sections of it. Löffelmann et al. [LKG97] apply this concept for the visualization of *Poincaré maps*, i.e. functions describing the behavior of an orbit through a lower dimensional space, a plane in this case. They visualize a section of the flow with glyphs or spot noise, while the orbit-plane intersections are highlighted with colored dots. Another example is given in the recent work by van Pelt et al. [vPBB$^+$10], who use cross-sections to emphasize the direction and intensity of blood flow in a vessel.

In the case of raw data, visibility management hasn't attracted too much attention. Taking into account some kind of importance measure can greatly increase the effectiveness of the results, therefore focus emphasis approaches are much more widespread for this kind of flow representation.

Visibility enhancement for integral structures

In this category a large number of approaches is concerned with the optimal placement of integral lines. There are two main issues that have to be considered:

1. too many lines would lead to cluttered results with high degree of occlusion;
2. uniformly placing seeding points in the space does not guarantee that the lines will be uniformly distributed as well.

The need for a seeding strategy was already identified in the early 90ies: for example Max et al. [MCG94] suggest to visualize particles and streamlines only close to previously computed surfaces. Similarly Löffelmann et al. [LDG98] suggest to seed streamlines in the proximity of selected critical points, which are usually relevant areas of the flow.

Another group of techniques try instead to partition the flow according to a specific clustering criteria, and then display one streamline (or a piece of it) for every cluster. Two approaches have been proposed in 1999, based on two different ideas: Heckel et al. [HWHJ99] use a top-down clustering, which iteratively subdivides the domain according to an error measure. In contrast, Telea and van Wijk [TVW99] employ a bottom-up strategy, merging the two most similar clusters at every step. The former has the advantage of showing more information in turbulent areas, but a large number cluster is needed to effectively represent the flow. The latter, in contrast, achieves good results with just a few clusters, but the similarity function requires many parameters to be set.

A more advanced technique has been proposed two years later by Garcke et al. [GPR$^+$01]: they use a continuous clustering based on the Cahn-Hilliard model, which describes phase separation in binary alloys. The main idea is to minimize a specific energy function, that can be customized in order to control the clustering process; the resulting partitions are nicely aligned with the flow and they can be visualized using either deformed particles or oriented streamline segments. Griebel et al. [GPR$^+$04] instead define an anisotropic diffusion tensor based on the flow direction, which, in turn, induce a strong (parallel to flow) and weak (orthogonal to flow) coupling between neighbour points. Once again the clusters are displayed using oriented curved arrows aligned with the streamlines. More recently Yu et al. [YWM07] propose a parallel approach for clustering unsteady vector fields in 4D, allowing for a cluster-based visualization of pathlines.

All these clustering approaches have the appreciated property of being hierarchical, therefore the density of the generated integral lines can be easily controlled. However, they have a major downside: even though the lines are usually nicely distributed over the domain, clustering takes into account only local properties of the vector field, so there are no guarantees that the resulting placement reflects any particular flow aspect or that the visibility is effectively optimized.

Approaches exist to explicitly address the distribution and the appearance of the streamlines in the final image: a notable example for 2D flows is due to Jobard and Lefer [JL97], who describe an approach to evenly place streamlines over the image with density specified by the user. In 2008 Li et al. [LHS08] suggest to compute a distance-based similarity measure, derived from information theory, to determine if a streamline is redundant or if it actually conveys new relevant information. For 3D vector fields, Li and Shen [LS07] propose a seeding strategy that takes into account image space information in order to avoid visual cluttering.

Three interesting approaches have been recently proposed that also take into account the communicative power of the integral curves they visualize; all of them are based on the information theory concept of *entropy*, which quantify the expected value of information contained in a message. Besides the different metrics adopted, these works share the idea of seeding streamlines from areas with high entropy measures. Usually the resulting visualization can still be cluttered, so an additional pruning process is needed. In particular, [MCHM10] use a view-dependent approach similar to [LS07], while [LMSC11] evaluates also the image space entropy obtained via *Maximum Entropy Projection*. Figure 5 instead is obtained with the approach of Chen et al. [CYY+11], which partitions the high entropy streamlines using a clustering technique and then visualizes only a few curves per cluster using the so called *Streamtapes*.

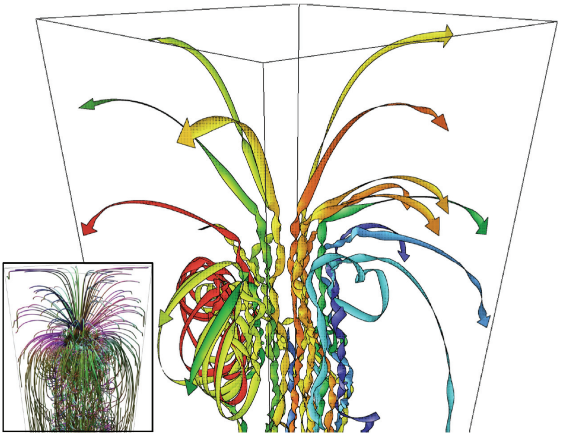


Figure 5: Streamtape visualization of the solar plume dataset compared to traditional streamlines (image courtesy of Chen et al. [CYY+11] ©2011 Eurographics).

Line placement is currently a very active research direction, but optimal visibility can be achieved in other ways as well. For instance, Weinkauf et al. [WTS11] propose to remove cusps and intersections created when path- and streaklines are projected onto the viewing plane, therefore obtaining a clean and expressive visualization. Regarding integral surfaces, already in 1997, Löffelmann et al. proposed the *stream arrow* metaphor [LMG97] in order to emulate Abraham and Shaw illustrations (like the streamsurface in Figure 2b). More recently, two notable approaches explicitly address visibility problems for streamsurfaces using different illustrative rendering techniques: Born et al. [BWF+10] suggest to use contour lines and halftoning to enhance the overall shape perception. Flow direction on the surface is depicted with oriented streamlines, while movable cuts and slabs allows for an interactive exploration of the flow. In contrast, the work of Hummel et al. [HGH+10] proposes two novel transparency techniques (angle-based and normal-variation) explicitly designed to expose hidden parts of the surface. They also employ additional illustrative techniques, i.e. adaptive stripe textures and silhouette enhancement: Figure 6 shows a streamsurface of a flow behind an ellipsoid obtained using a grid-like texture, normal-variation transparency and emphasized contours.

A special case: temporal implosion

Temporal implosion is an illustrative technique aimed at depicting the temporal evolution of a certain system in a single, static image. It is extensively used, for example, in comics and photos (via post processing) to convey the motion of an object, and, in the last few years, it has been successfully employed to visualize the behaviour of dynamical systems.

In flow visualization, temporal implosion is well suited to show the trajectory of features, like vortices or saddles: a first approach was proposed in 2001 by Reinders et al. [RPS01], who employ a prediction and verification method to track the features over time, and visualize their past, current and predicted positions using 3D elliptical icons. They also detect events like feature splitting or merging, and summarize the evolution of the currently tracked features in a graph.

Even though temporal implosion essentially relies on the tracking of a specific object, the basic idea can be applied also to raw data: Balabanian et al. [BVMG08] developed a 4D raycasting algorithm which is able to render multiple volumes (corresponding to multiple timesteps) simultaneously. They also employ an adapted version of the *style transfer functions* [BG07] in order to vary the rendering style along the temporal dimension. Figure 7 has been generated using the approach by Hsu et al. [HMCM10]: in this case every timestep is treated separately, so its visual appearance can be finely tuned and even the final layout can be modified in order to create either overlapped or side-by-side views.

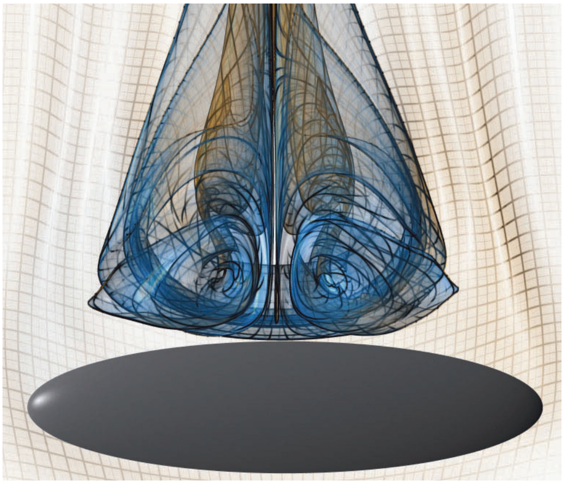


Figure 6: Streamsurface of the ellipsoid dataset rendered with normal-variation transparency, grid-like texture and silhouette enhancement (image courtesy of Hummel et al. [HGH$^+$10] ©2010 IEEE).

Recently van Pelt et al. [vPGtHRV11], following the guidelines in [JR05], propose to convey the motion of blood using illustration techniques typical of comics and cartoons: blood particles, represented by spheres, are deformed along flow direction according to the velocity magnitude, while the illustrative rendering of reversed pathline effectively depicts particles trajectories.

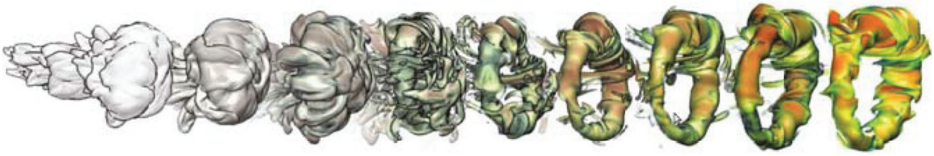


Figure 7: Side-by-side view of a smoke ring extracted from the argon bubble dataset, rendered using different styles (saturation and silhouettes) for each timestep (image courtesy of Hsu et al. [HMCM10] ©2010 Springer Berlin Heidelberg).

4.4 Focus Emphasis

The techniques included in this category are all based on an importance measure that clearly distinguishes the relevant portions of the data from the less interesting ones. This concept is actually well-known in data analysis approaches, like Interactive Visual Analysis (IVA) or Visual Analytics (VA), to identify significant subsets of the data in usually multiple linked views. In illustrative visualization the importance values are instead used to explicitly control one or more visual properties: it is even possible to render parts of the dataset with completely different techniques according to their importance values.

Raw data with importance values

It is a quite common practice to let the user define what he deems relevant and what he does not. In the most simple case, the important area is limited to a small number of isolated locations manually specified in spatio-temporal domain, while everything else is considered uninteresting. This idea has been exploited, for example, by many texture-based algorithms in order to simulate the classical experiment of dye injection in a flow. Already in 1996, Shen et al. [SJM96] proposed to use a grayscale 3D LIC to show the context while the dye (the focus) is rendered with colors smeared according to the velocity vectors. A similar approach was adopted by Telea and van Wijk [TvW03] in 2003, but their technique is based on 3D IBFV instead of LIC. One year later, Weiskopf and Ertl [WE04] proposed another dye advection technique based on 3D IBFV, which is able to visualize even unsteady flow at interactive rates. Two more recent approach are specifically aimed at making the appearence of the dye more realistic: Weiskopf et al. [WBE05] suggest to vary the intensity of the colors in order to explicitly represent the variation of the dye density caused by the convergence and divergence of the flow. Li et al. [LTH08], instead, introduce a novel 2D dye advection algorithm which, applying concepts typically used in computational fluid dynamics, is able to generate highly realistic results.

Dye injection is justified by historical reasons and it has indeed proven its usefulness over the years; however, in a computerized environment, more advanced

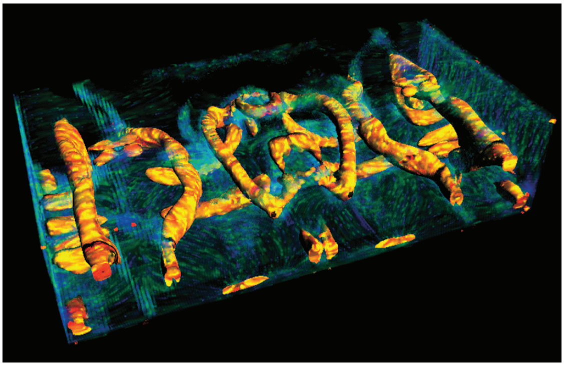


Figure 8: Focus emphasis of swirling areas of the flow (image courtesy of Weiskopf et al. [WSE07] ©2007 IEEE).

exploration techniques can be developed. This is the case, for instance, of the *Chamaleon* rendering framework [SLB04]: the focus is represented by the so called *trace volume*, which encodes particles trajectories and is visualized via volume rendering; a specific texture is used to control the appearance of the reuslts. The spatial locations included in the trace volume can be interactively specified and, choosing the appropriate texture, many illustrative effects can be obtained, like directional glyphs or tone shading. A curious exploration framework has been recently proposed by Isenberg et al. [IEGC08], whose goal is maximizing user interactions using a multitouch display: the 2D flow under investigation can be displayed using a background texture and the user can interactively place customizable animated glyphs in the area of interests.

When analysing complex, multivariate phenomena like flows, relevant portions of the data cannot be easily identified by their spatial locations alone, usually precise criteria based on one or more variables are needed. A significant work in this direction has been proposed by Hauser and Mlejnek in 2003 [HM03]: they describe an IVA framework where the user can select interesting areas in histograms or scatterplots of the flow variables. Each point of the dataset is therefore associated with a fuzzy importance value, the *degree of interest* (DOI), which is used in the visualization phase to modulate transparency or to define isosurfaces. A fuzzy importance measure is adopted also in [WSE07] and [FW08] to control different visualization parameters. The proposed visualizations are based on 3D texture advection and different shading and illumination schemes, and the importance values are directly mapped to appearance and transparency through a transfer function.

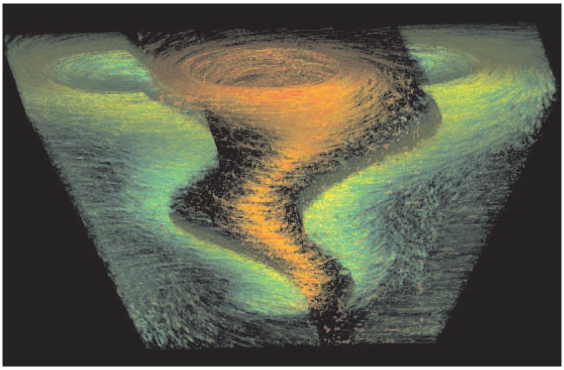


Figure 9: Illustrative deformation of flow in order to avoid the occlusion of the focus region (image courtesy of Correa et al. [CSC07] ©2007 IEEE).

In Figure 8, for example, swirling areas are emphasized using high opacity and a warm colors, while the context is more transparent and less colorful.

Figure 9 has been obtained with the approach of Correa et al. [CSC07]: they employ a fuzzy importance function, named *Level of Desired Attention*, and guarantee that the focus is never occluded deforming the context according to one of the many templates provided. With the due exceptions [AH11], deformations are in general not well suited for flows, since they alter the directional information, but in this case only the context is modified, the relevant part of the data is untouched and effectively emphasized.

All the focus emphasis techniques discussed up to now are based on a user specified importance measure, but some attempts have been made to automatically identify significant regions of the dataset. According to the concept of *entropy* already introduced in section 4.3, Wang et al. [WYM08] suggest to subdivide the dataset into blocks and evaluate the amount of information contained in each block over time. A clustering algorithm groups together blocks with similar entropy evolutions and the resulting clusters can be then visualized in different ways. This approach has been recently extended in [WYG$^+$11], adopting an importance function based on *transfer entropy*, i.e. a measure of causal dependencies between variables; the normalized importance value is then used to modulate saturation and opacity, producing expressive visualizations like the one in Figure 10.

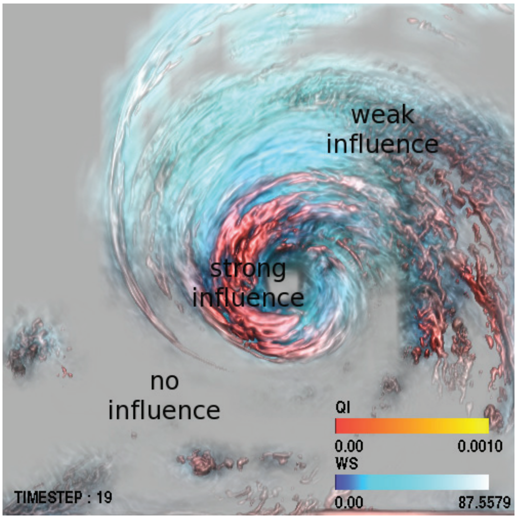


Figure 10: High transfer entropy regions are depicted with full saturated colors, while context is completely desaturated (image courtesy of Wang et al. [WYG$^+$11] ©2011 IEEE).

Focus+context approaches for integral structures

The concept of focus emphasis has been successfully applied to integral structures as well. In particular, three main directions have been investigated, each one with different goals and expected results. The first one is based on the simple assumption that the user can freely specify the focus directly on the spatial domain, and integral structures in that area have to be visually emphasized. For example, in the work of Fuhrmann and Gröller [FG98] from 1998, the focus is specified as a portion of the volume defined by either a 2D (magic lens) or a 3D (magic box) selection. Streamlines are then visualized according to three different criteria: (1) context in front of the focus is suppressed to avoid occlusions; (2) in the remaining context areas, only a few streamlines are visualized; (3) the focus areas are rendered with dense bundles of streamlines.

A second category is instead represented by those approaches that use the focus as the seeding area for the integral structures. Based on this idea, [HM03], already mentioned before, seeds streamlines in regions where the DOI is maximal, while in [MTHG03] the focus can be either used to increase streamlines density or to seed new streamlines. [KKKW05] follows a similar strategy, but particles are seeded instead of streamlines, and, in a second phase, different types of particle-based visualizations can be generated, like oriented glyphs, stream balls or stream ribbons.

Two notable approaches deal instead with the construction of integral surfaces: Schafhitzel et al. [STWE07] construct either stream- or pathsurfaces starting from a curve selected by the user in the spatial domain; their algorithm is based on a GPU implementation and runs at interactive rates. On the other hand, Wiebel and Scheuermann [WS05] suggest to focus on a set of user selected seeding points, the so called *eyelets*, and streaklines, pathline and pathsurfaces are simultaneously computed. They also provide the user with different guidelines for the selection of the eyelets, in order to maximize the expressiveness of the resulting structures.

Recently, a more advanced exploration technique has been proposed by Ferstl et al. [FBTW10] (see Figure 11): the user first places a 2D plane (the focus) in the 3D spatial domain, then FTLE values, i.e. a measure of particles' divergence over time, are computed on it. Ridges of the FTLE scalar field are then detected, and the resulting curves are used to seed streaksurfaces. Moreover, timesurfaces, rendered using green arrows, are periodically released from the plane. For more details on FTLE, the reader can refer to [Hal01].

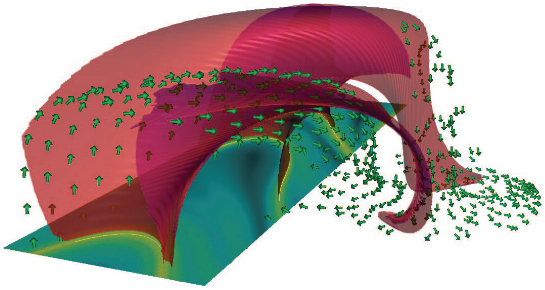


Figure 11: A streaksurface, in red, is seeded from the FTLE ridge on the plane, while green arrows convey directional information on timesurfaces (image courtesy of Ferstl et al. [FBTW10] ©2010 IEEE).

The third and final category includes approaches that use whole integral curves as the focus of the visualization. In 2010, two notable approaches based on this concept have been presented: Jones and Ma [JM10] present a flow exploration framework that, among other functionality, allows the user to select groups of streamlines which will be visualized in a focus+context fashion. In particular, occluding geometries are removed, and additional information, like the distance from the focus curves, are depicted in the context. Wei et al. instead [WWYM10] developed a sketch-based interface for streamlines selection: the user first sketch the shape of the streamlines he deems interesting, then, according to a similarity measure, the corresponding curves are highlighted. This direction has not been heavily investigated yet, but the current results seem very promising and we expect substantial developments in the near future.

Flow features

At this point it should be clear enough that the number of illustrative approaches that deal with flow features is meagre, but at least one work can be identified that complies with the focus emphasis principle. This technique has been proposed in 2009 by Muelder and Ma [MM09], and its main focus is the interactive extraction and tracking of flow features. The details of their algorithm exceed the scope of this manuscript, what really matters here is that, at the end of the procedure, flow features are clearly segmented. As can be seen in Figure 12, this segmentation can be used during the volume rendering process to emphasize the features of interest while still providing the less relevant context information. Notice that many illustrative visualization techniques can be easily integrated into volume rendering algorithms, therefore it is our opinion that this approach would be just the first in a long series of works on the illustrative depiction of flow features.

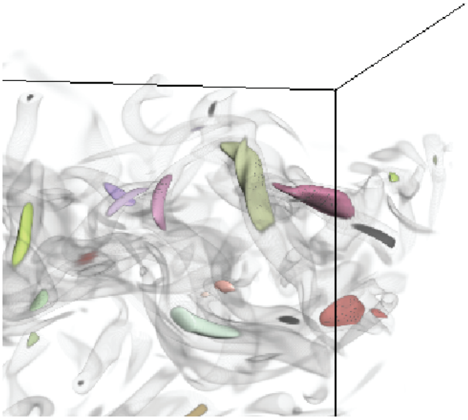


Figure 12: Flow features tracked over time and visualized together with context information (image courtesy of Muelder and Ma [MM09] ©2009 IEEE).

4.5 Visual Explanation

Visual explanation approaches are a superset of Guided Visualization and Interactive Storytelling, and their goal is to give an explicative visual description of the underlying phenomena. The challenge here is twofold: besides the usual visualization issues, it is also necessary to identify an appropriate set of objects, properties and relations that can lead to an effective and concise explanation of the phenomenon of interest. A natural and straightforward way to define this piece of semantic information is to apply one or more selection criteria to the data according to spatial, temporal or attribute values. This is the approach adopted

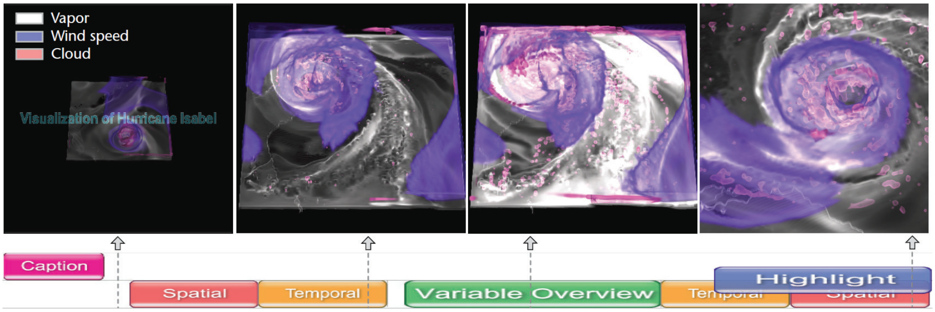


Figure 13: Visual explanation using the Aniviz framework: visualization templates are arranged along the timeline to produce an effective presentation of the dataset (image courtesy of Akiba et al. [AWM10] ©2010 IEEE).

by *Aniviz* [AWM10], an animation framework designed to highlight different aspects of volume data. The user is presented with multiple animation templates which can be freely arranged along a timeline in order to build a visual presentation of the dataset. The templates and the transitions between them can be controlled by a set of parameters, which can be used to finely tune what aspects of the data has to be emphasized (see Figure 13). This framework currently deals with raw data only, but extending it to integral structures or flow features should be fairly easy.

A recent work by Pobitzer et al. [PTA$^+$11] suggests to describe the flow using a scale-space approach: they decompose the vector field according to the level of transport energy present in the flow, so the user can choose to focus either on the main transport structures or on the smaller scale turbulence. Only raw data is taken into account in the computations, but the derived vector fields can be further processed: this can be useful, for instance, to identify which flow features have a greater transport energy.

Regarding particle integration and integral structures, an interesting approach has been proposed by Bürger et al. [BKKW08] in 2008. Their work is based on two additional functions defined on the flow domain: a customizable fuzzy importance degree and a measure of the local coherence of the vector field; different visualization techniques are then employed according to these two parameters. The

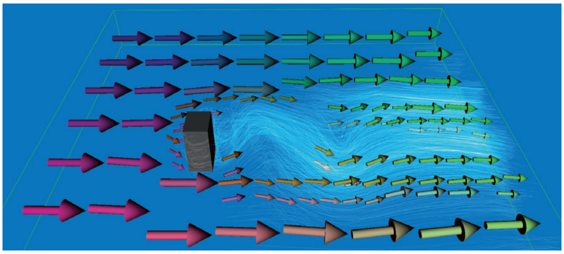


Figure 14: Flow depicted at different levels of detail (image courtesy of Bürger et al. [BKKW08] ©2008 IEEE).

descriptive part is given mostly by the coherence measure, which in fact affects the level of detail of the visualization: as can be seen in Figure 14, where the local coherence is low, streamlines are used to convey the (turbulent) flow behavior. In contrast, in areas where the flow is more stable, oriented glyphs are placed, and both their size and density are varied as the coherence value increases. The importance function can be used either to explore specific locations of the domain or to emphasize particular aspects of the flow. Notice that the concept of "adaptive visualization", which is nicely exploited in this approach, is very effective in conveying different aspects of the flow simultaneously, and we expect to see future work based on it.

5 Final Remarks and Future Expectations

We have reviewed and classified many illustrative flow visualization approaches, emphasizing their respective merits and discussing their downsides. The classification introduced at the beginning of Section 4 clearly highlights that there is no approach that is universally better then the others, many aspects have to be taken into account in every situation. In our opinion, the answers to "what flow representation do I want to refer to?" and "what visualization enhancement do I need?" are two excellent guidelines to make a first choice, but further refinements are possible: for example, the personal preferences and habits of the final users hold a certain relevance. Moreover, since the various algorithms have very different computational complexity, some considerations have to be made regarding the available hardware and the size and structure of the datasets. It should not come as a surprise that the best approach varies from case to case and from user to user.

Over the last decade, illustrative flow visualization techniques have been proposed and applied in many different contexts, and, looking back, interesting considerations can be made:

- Perceptual effectiveness has been the first illustrative concept being applied to visualization in general. It has been thoroughly investigated and many effective and useful algorithms have been proposed. Nowadays there is the feeling that these techniques alone are not effective enough, so they are usually coupled with some higher visual abstraction approaches.
- Smart placement of integral curves were, and still are, one of the most prominent research directions in the field, but visibility enhancement for integral surfaces has been recently attracting a lot of attention as well.
- Temporal implosion has not been investigated too much, probably because of its inherent technical complexity, but many see a lot of potential in it, especially for the visualization of unsteady flows.

- Focus emphasis techniques are highly appreciated especially for flow exploration and analysis, and we expect that in the near future any visualization framework will implement similar functionality.

Visual explanation techniques and illustrative visualization of flow features are still somewhat unexplored, probably because of the same underlying problem: both these categories rely on a semantic knowledge base of the phenomenon of interest, but our understanding of flows is still limited. It is our opinion that, as new aspects of flow dynamics are discovered, research in both these areas will grow accordingly.

In the near future we expect to see a constantly increasing degree of interaction between the flow and the illustrative visualization branches: existing approaches typically consider just the raw information encoded in the dataset, but usually no knowledge about the physical properties of the underlying flow is considered. In an ideal setup, flow investigation would become an iterative process with continuous interaction between the illustrative flow visualization expert and the user: the former should initially provide a set of suitable visualization tools, the latter should use them for a preliminary analysis of the dataset, producing feedback about the tools and a set of initial findings. Both the feedback and the new findings will be taken into account to improve the visualization tools, and the whole process will start anew.

Of course illustrative flow visualization cannot answer all the open questions or solve all the current issues in flow analysis, but hopefully it will help to shed some light on this still not well understood phenomenon.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback.
This report has been worked out within the scope of the SemSeg project and we acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042.