Conference contribution :
Laramee, R. (2005). *Interactive 3D Flow Visualization Based on Textures and Geometric Primitives.* NAFEMS World Congress (NWC) Conference Proceedings,

Interactive 3D Flow Visualization Based on Textures and Geometric Primitives

Robert S. Laramee and Helwig Hauser
www.VRVis.at [1]

SUMMARY

As the size of CFD simulation data sets expand, the job of the engineer to analyze, explore, and present the data becomes more challenging. The scientific visualization tools used by the engineer should evolve to meet the growing demands presented by large simulation data sets. Furthermore, no single visualization technique can meet each users needs. We present a detailed selection of recently developed direct, geometric, and texture-based flow visualization techniques. These techniques address the demand, set forth by engineers, for visualization solutions which provide insight into CFD simulation data. Included are algorithms for (1) the resampling of CFD simulation data, (2) fast, animated texture-based flow visualization, and (3) geometric flow visualization including dashed, animated-streamlines, oriented streamlines, streamlets, and streamcomets. Each approach is targeted at the visual analysis of computational fluid dynamics (CFD) simulation data. This relatively new selection of techniques provides valuable tools that allow engineers to gain insight into their CFD simulation results.

1 Introduction

Visualization is an important part of exploring, analyzing, and presenting the results of a CFD simulation. As the size of CFD simulation data sets increases, the utility of scientific visualization for gaining insight into the data sets also increases. Visualization offers one way to manage such large collections of simulation data since it brings the data to a higher level of abstraction. Simply reading the raw data does not meet all of the demands set forth by the user and may not even be feasible. Furthermore, no single visualization solution can span the range of each engineer's needs. Hence a range of solutions must be at the user's disposal.

We present a selection of recent advances in flow visualization that addresses the growing demand for solutions that offer insight into the continuously expanding CFD simulation data sets. Our presentation draws upon flow visualization techniques being classified into four main categories: (1) direct, (2) texture-based, (3) geometric, and (4) feature-based [8, 13]. This classification provides a framework useful for placing our techniques in a larger context. The resampling tool we present falls into the direct flow visualization category [6]. The Image Space Advection (ISA) and Image Based Flow Visualization for Curved Surfaces (IBFVS) algorithms fall into the texture-based category [11]. And our discussion of geometric techniques includes, oriented streamlines, animated-dashed streamlines, streamlets, and streamcomets [7]. These recent advances are applied to real-world applications in the field of CFD including the investigation and visualization of patterns of flow motion specific to automotive engineering. We discuss the benefits of these techniques along the way using real-world results.

[1] Research Center for Virtual Reality and Visualization, Donau-City-Strasse 1, 1220 Vienna, Austria, {Laramee,Hauser}@VRVis.at

The rest of this paper is organized as follows: Section 2 briefly outlines some related work. Section 3 describes a resampling tool for CFD simulation data. Section 4 presents recent algorithms for the fast advection of textures on surfaces. Section 5 outlines how texture-based flow visualization can be applied to isosurfaces. Section 6 describes a collection of geometric flow visualization techniques applicable to CFD simulation data and Section 7 demonstrates how direct, geometric and texture-based flow visualization techniques can be used to solve real-world problems from the automotive industry.

2 Related Work

Four different approaches are widely used in flow visualization [8, 14]:

Direct flow visualization: This category of techniques uses a translation that is as straight forward as possible for representing flow data in the resulting visualization. The result is an overall picture of the flow. Common approaches are drawing arrows or color coding velocity. Intuitive pictures result, especially in the case of two dimensions. See Schulz et at. [16] for good illustrations of direct flow visualization techniques applied to CFD simulation data. Westermann demonstrates a volume rendering technique in order to visualize 3D vector data [23].

Geometric flow visualization: These approaches often first integrate the flow data and use geometric objects in the resulting visualization. The objects have a geometry that reflects the properties of the flow. Examples include streamlines, streaklines, streamsurfaces, and timelines. Not all geometric objects are based on integration. Another useful geometric approach is generating isosurfaces, e.g., with respect to an isovalue of pressure or magnitude of velocity. A more thorough description of geometric techniques is presented by Post et al. [13]. See the work of Schroeder et al. [15] or Zöckler et al. [24] for classic examples of geometric flow visualization technique applied to 3D vector data.

Dense, texture-based flow visualization: A texture is computed that is used to generate a dense representation of the flow. A notion of where the flow travels is incorporated through co-related texture values along the vector field. In most cases this effect is achieved through filtering of texels according to the local flow vector. Texture-based methods offer a dense representation of the flow with complete coverage of the vector field. Recent examples include Image Based Flow Visualization (IBFV) [19] and Image Space Advection (ISA) [11], which can generate both Spot Noise [18] and LIC-like [1] imagery. We note that a full comparison of texture-based flow visualization techniques is given elsewhere [8].

Feature-based flow visualization: Another approach makes use of an abstraction and/or extraction step which is performed before visualization. Special features are extracted from the original dataset, such as important phenomena or topological information of the flow. Visualization is then based on these flow features (instead of the entire dataset), allowing for compact and efficient flow visualization, even of very large and/or time-dependent datasets. This can also be thought of as visualization of *derived* data. Post et al. [14] cover feature-based flow visualization in detail. The work of Doleisch et al. [2, 3] presents recent developments in feature specification and extraction.

3 Resampling of CFD Simulation Data

To start off, we introduce a flexible, variable resolution tool for interactive resampling of computational fluid dynamics (CFD) simulation data on unstructured grids. The tool and coupled algorithm

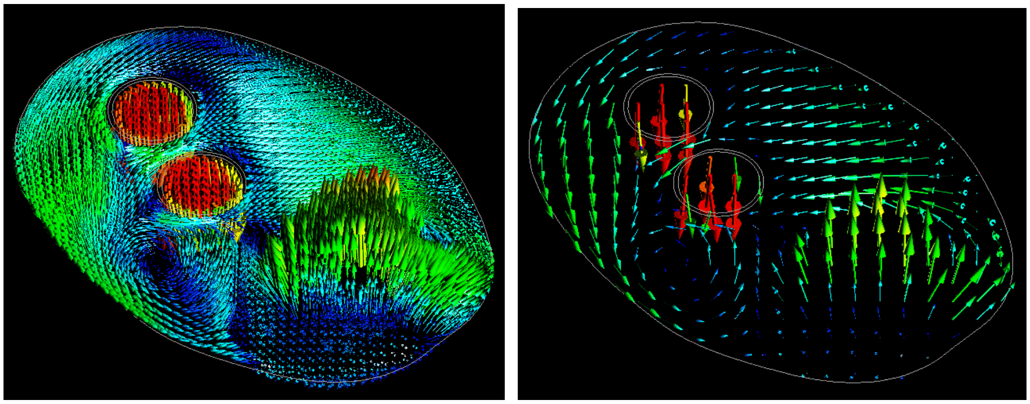

Figure 1: (top) A slice mesh with discontinuities -two gaps in the shape of rings, using hedgehog visualization (bottom) The same slice resampled onto a regular grid.

afford users precise control of glyph placement during vector field visualization via six interactive degrees of freedom. The resampling tool, called FIRST (a Flexible and Interactive ReSampling Tool), is a valuable asset in the engineer's pursuit of understanding and visualizing the underlying flow field in CFD simulation results [6]. FIRST solves both the perceptual problems resulting from a brute force hedgehog visualization approach, where a vector glyph is rendered at every CFD grid cell, and glyph placement problems by (1) giving the user control of the *resolution* of the glyphs in the image and (2) giving the user precise control of *where* to place the vector glyphs for viewing the flow with normal components.

The key distinguishing features of FIRST stem from the fact that it was specifically developed in order to provide the user with a range of flexible interactions at multiple resolutions. The reason we focus on a combination of user control with resampling is because engineers require interactive visualization solutions. This is partly due to a large amount of time engineers spend searching the data sets. The analysis of an engineer includes tasks such as searching for areas of extreme pressure, looking for symmetries in the flow, searching for critical points, and comparing simulation results previous simulation results and with measured, experimental results.

FIRST provides the following features: (1) six interactive DoFs: three translational, scaling, rotation, and resolution, (2) handles changes to both underlying topology and geometry, i.e., can be utilized for the display of time-dependent, unstructured grid slices where geometry and topology change over time or space, (3) resamples any unstructured grid onto any structured grid, (4) handles unstructured grids with holes and discontinuities, (5) does not rely on any pre-processing of the data (6) consists of a straightforward implementation, e.g., requires no neighbor-finding capabilities or complicated data structures, (7) processes large quantities of unstructured, scalene triangles efficiently. Also, the underlying algorithm operates on a per-unstructured-polygon basis, making it suitable for parallelization.

The resampler features are associated with a user-defined, 2D slice through a 3D mesh from CFD. Engineers take a slice of the data and slide the slice through the geometry in order to find features of the simulation data, e.g., areas of extreme pressure and vortices. As the user moves the slice through the 3D mesh, the resampler automatically resizes itself around the slice boundary, handling changes to both the underlying geometry and topology.

Figure 1 illustrates our technique on a mesh with discontinuities. The discontinuities are two gaps

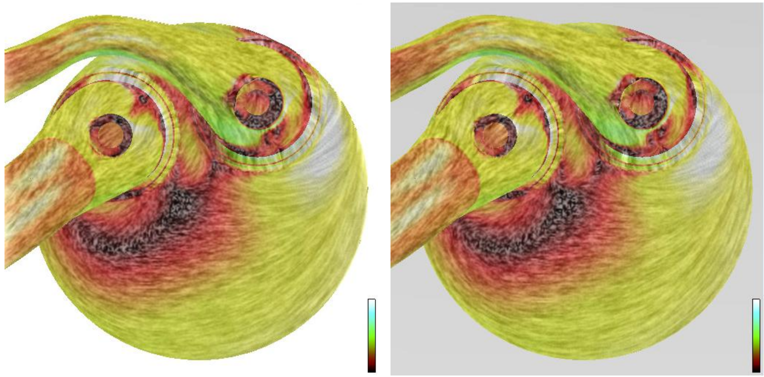

Figure 2: ISA and IBFVS applied to the flow simulation data at the surface of two intake ports side by side: ISA with the white background and IBFVS with the gray background.

in the shape of rings. Visualization of flow with normal components is shown using both the hedgehog technique versus the glyphs onto a resampled grid [2]. Figure 1 illustrates how FIRST reduces occlusion and visual complexity thus making the results more suitable for presentation. Also, rendering times are accelerated because the number vector glyphs is reduced.

4 Image Space Based Visualization of Flow on Surfaces

In this section we present a brief, side-by-side analysis of two recent image space approaches for the visualization of vector fields on surfaces. The two methods, Image Space Advection (ISA) [9] and Image Based Flow Visualization for Curved Surfaces (IBFVS) [20] generate dense representations of time-dependent vector fields with high spatio-temporal correlation (see Figure 2). While the 3D vector fields are associated with arbitrary surfaces represented by triangular meshes, the generation and advection of texture properties is confined to image space. Fast frame rates are achieved by exploiting frame-to-frame coherency and graphics hardware. In our comparison of ISA and IBFVS we point out the strengths and weaknesses of each approach and give recommendations as to when and where they are best applied.

The class of fluid flow visualization techniques that generate dense representations based on textures started with Spot Noise [18] and LIC [1] in the early 1990s. The main advantage of this class of algorithms is their *complete* depiction of the flow field while their drawbacks are, in general, the computational time required to generate the results, lack of flow orientation (upstream vs. downstream), and applicability to 3D flow. Figure 2 illustrates two texture-based techniques that maintain complete coverage of the vector field while at the same time overcome some of the aforementioned disadvantages, namely, long computation time and lack of flow orientation [11]. Texture values are correlated along the direction of the flow. The orientation of the flow is especially apparent in an interactive animation.

Traditional visualization of boundary flow using texture mapping first maps one or more 2D textures to a surface geometry defined in 3D space. The textured geometry is then rendered to image space [17]. Here, we alter this classic order of operations. First we project the surface geometry and its associated vector field to image space and then apply texturing. In other words, while conceptually texture properties are advected on boundary surfaces in 3D, in fact the algorithms realize

[2]For supplementary images and MPEG animations of the resampler, please visit:
`http://www.VRVis.at/ar3/pr2/resampler/`

texture advection in image space. In particular, the methods have the following characteristics: They (1) generate a dense representation of unsteady flow on surfaces (Figure 2), (2) visualize flow on complex surfaces composed of polygons whose number is on the order of 250,000 or more, (3) can handle arbitrary, complex meshes without relying on a parametrization, (4) support user-interaction such as rotation, translation, and zooming while maintaining a constant, high spatial resolution, (5) deliver high performance, i.e., several frames per second, and (6) visualize flow on dynamic meshes with time-dependent geometry and topology.

ISA and IBFVS simplify the problem by confining the synthesis and advection of texture properties to image space. Both methods project the surface mesh and associated vector data to image space and then apply a series of textures. Summarizing, the methods are comprised of the following pipeline:

1. associate the 3D flow data with the polygons at the boundary surface, i.e., a velocity vector is stored at each polygon vertex of the surface
2. project either the associated vector field or the distorted surface mesh onto the image plane
3. advect texture properties according to the projected vector field or the projected, distorted mesh in image space
4. inject and blend noise
5. overlay optional visualization cues such as showing a semi-transparent representation of the surface with shading

Each step of the pipeline is necessary for the dynamic cases of unsteady flow, time-dependent geometry, rotation, translation, and scaling, and only a subset is needed for the static cases involving steady-state flow and no change to the viewing parameters. Conceptually, the algorithms share several overlapping components such as projection to image space, advection mesh computation, texture-mapping, noise injection and blending, and the addition of shading or a color map. The main difference between the two methods is that ISA uses an image-based mesh in order to advect the textures, whereas the texture advection in IBFVS is driven by the original 3D mesh. As a result, differences arise stemming from parts of the algorithm that use image space vs. object space. These decisions result in advantages and disadvantages for both methods. More details are given by Laramee et al. [11] [3].

The choice of whether to apply ISA or IBFVS depends on the complexity of the model. For surface visualization IBFVS is a good choice. Polygons generally cover several pixels in image space hence the amount of computation time per-pixel is low. For visualization of flow on large meshes such as shown in Figure 2, ISA is a good choice. For these meshes, many polygons cover less than a pixel and many polygons are occluded. ISA avoids spending computation time on these pixels. Also, in terms of software development, IBFVS is generally easier to implement because it is a more straightforward extension of IBFV [19].

5 Texture Based Visualization of Flow on Isosurfaces

For many of the automotive components that undergo evaluation, there is an ideal pattern of flow engineers try to create. Figure 3, left illustrates the swirl motion of fluid flow in a combustion chamber from a diesel engine. In order to generate swirl motion, fluid enters the combustion

[3] For supplementary images and MPEG animations of ISA and IBFVS, please visit: `http://www.VRVis/ar3/pr2/isa-ibfvs/`

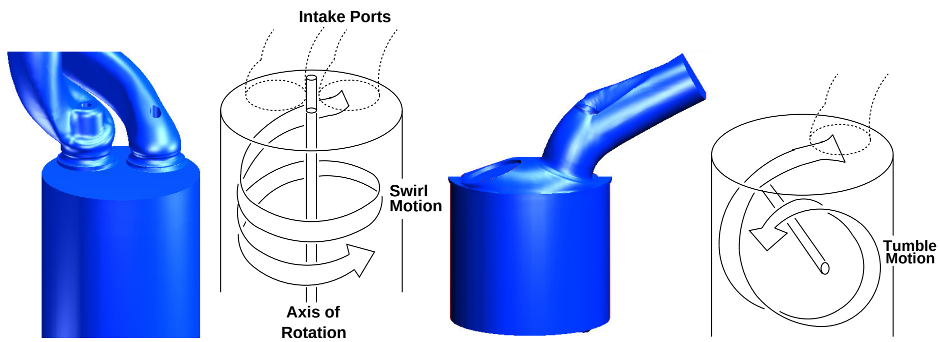

Figure 3: (left) The swirling motion of flow in the combustion chamber of a diesel engine. *Swirl* is used to describe circulation about the cylinder axis. The intake ports at the top provide the tangential component of the flow necessary for swirl. The data set consists of 776,000 unstructured, adaptive resolution grid cells. (right) Some gas engine components require a *tumble* motion flow pattern (right) in order to mix fluid with oxygen. Tumble flow circulates around an axis perpendicular to the cylinder axis, orthogonal to swirl flow. The data set is composed of 61,700 unstructured, adaptive resolution cells.

chamber from the intake ports. Later on in the engine cycle, the kinetic energy associated with this swirl motion is used to generate turbulence for mixing of fresh oxygen into the fluid. The more turbulence generated, the better the mixture of air and diesel fuel, and thus the better the combustion itself. Ideally, enough turbulent mixing is generated such that 100% of the fuel is burned.

From the point of view of the mechanical engineers designing the intake ports, increased swirl flow leads to beneficial conditions: (1) improved mixture preparation, i.e., more fuel contact with oxygen, (2) a higher EGR (Exhaust Gas Ratio) which means a decrease in fuel consumption, and (3) lower emissions. However, too much swirl displaces the flame used to ignite the fuel. As such, a balance must be achieved between generating enough swirl flow in order to create turbulence and not displacing the flame used to ignite the flow. Some routine questions that a mechanical engineer may ask when investigating swirl flow are: Can visualization provide insight into or verify the characteristic shape(s) or behavior of the flow? What tool(s) can help to visualize the swirl flow pattern? and Where in the combustion chamber is the swirl flow pattern *not* being met?

Isosurfaces are a visualization tool used routinely by mechanical engineers to investigate the properties of the flow inside a 3D volume. The shape of an isosurface can give the engineer insight into its 3D characteristics. Figure 4, left shows a velocity isosurface in the combustion chamber of the data set in Figure 3, left. The engineer can see that the flow has some of the swirling orientation that they are looking for. However, what is missing from Figure 4, left, is a clear indication of flow direction, e.g., the upstream and downstream nature of the flow. In particular, it is not obvious where the flow does *not* follow the ideal swirl pattern that the combustion chamber should encapsulate.

Applying texture-based flow visualization techniques to such isosurfaces provides engineers even more insight into the characteristics of 3D vector fields. And this has become a feasible option only recently. We applied the ISA method of Laramee et al. [9] for producing dense, texture-based representations of flow on isosurfaces. The result is a combination of two well known scientific visualization techniques, namely iso-surfacing and texture-based flow visualization, into a useful hybrid approach. Our application is a versatile visualization technique with the following characteristics: (1) generates a dense representation of flow on adaptive resolution isosurfaces, (2)

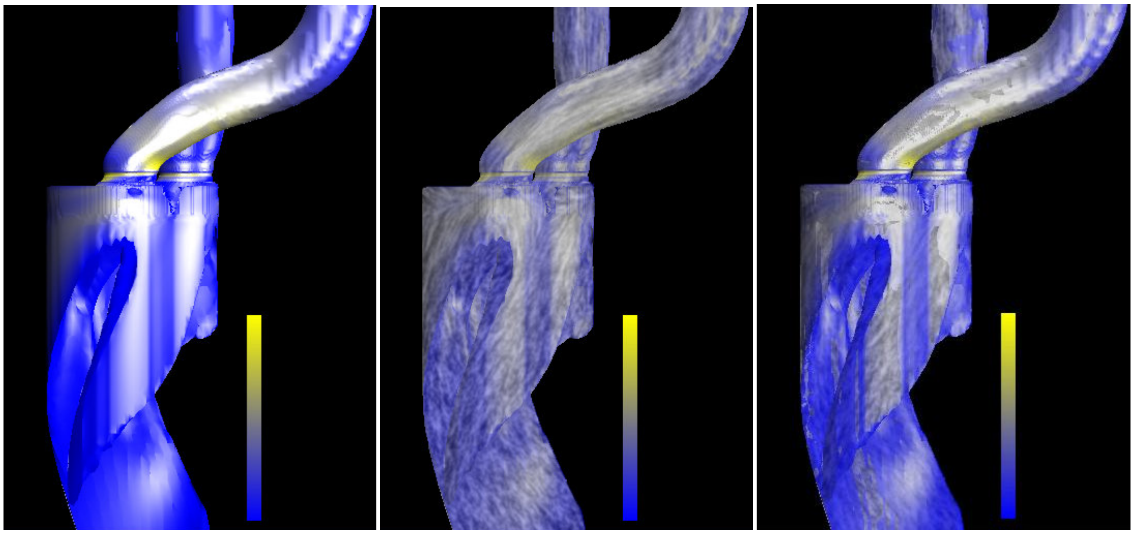

Figure 4: (left) a velocity isosurface of value 5.0 m/s with a CFD simulation attribute mapped to hue, (middle) with texture-based flow visualization applied, (right) texture-based flow visualization on the isosurface combined with a normal mask.

visualizes flow on complex isosurfaces composed of polygons whose number is on the order of 200,000 or more, (3) visualizes flow independent of the isosurface mesh's complexity and resolution, (4) supports user-interaction such as rotation, translation, and zooming always maintaining a constant, high spatial resolution, and (5) produces fast animations, realizing up to 60 frames per second.

When visualizing flow on normal boundary surfaces the direction of the flow generally coincides with the surface itself. As the flow approaches the boundary, it is not allowed to pass through and is pushed in a tangential direction, i.e., it can be described as surface aligned flow. However, in the case of isosurfaces this is no longer true. The flow at an isosurface can sometimes exhibit a strong flow that is normal to the surface, e.g., cross-surface flow. The same also holds true for the case of arbitrary clipping geometries. Simply advecting texture properties according to the vector field projected onto the isosurface could be considered misleading.

We propose an idea inspired by the well known velocity mask [4], namely, a *normal mask*. A velocity mask can be used to dim or highlight high frequency noise in low velocity regions. Whereas, a normal mask can be used to dim regions of the vector field that have strong cross-flow component to the isosurface. For more details about how we implemented the normal mask, please refer to Laramee et al. [10]. Some results of applying this normal mask to an isosurface are shown in Figure 4, right. We can see that the flow at the isosurface just below the intake port in the foreground (in white) has a strong normal component to the isosurface. The higher frequency texture in this region is difficult to see. Note also that we have chosen a simpler color scale in this case to reduce the visual complexity of the result. We find that using using a full range of hue for the color mapping in combination with variable opacity for the normal mask is visually complex. So we provide the option of trading off some complexity in the color map while applying the normal mask [4].

[4]For supplementary images and animations of texture-based flow visualization on isosurfaces, please visit: http://www.VRVis.at/ar3/pr2/VisSym04/

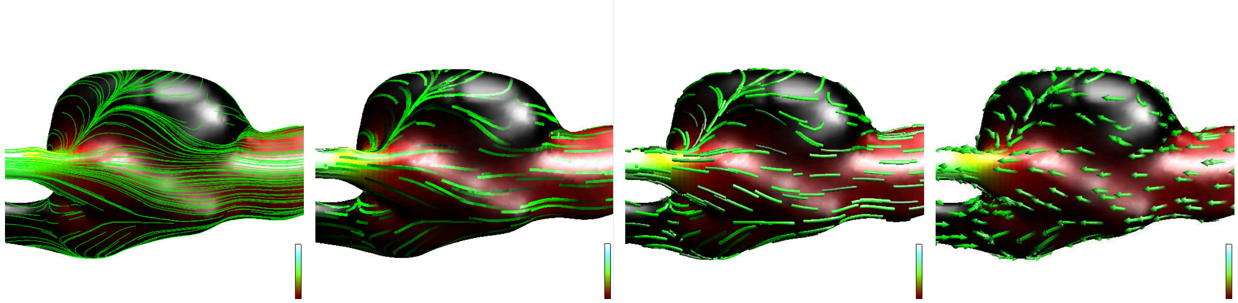

Figure 5: The visualization of blood flow at the surface of an aneurysm: (left) geometric flow visualization using streamlines (middle-left) oriented streamlines and (middle-right) streamlets, and (right) streamcomets.

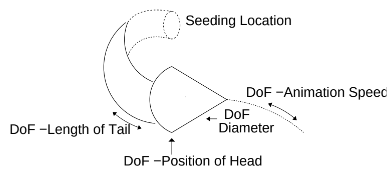

Figure 6: The streamcomet promotes four interactive DoFs: (1) the position of the comet head along the path of integration, (2) the diameter of the comet head and tail, (3) the length of the semi-transparent comet tail, and optionally (4) the animation speed of the comets along the path of integration.

6 Geometric Flow Visualization Techniques

This section turns our attention to a collection of geometric flow visualization techniques including oriented streamlines, streamlets, a streamrunner tool, streamcomets, and a real-time animated streamline technique. We place special emphasis on necessary measures required in order for geometric techniques to be applicable to real-world data sets. While some of the geometric techniques here have been presented in previous literature, they are often not illustrated in the context of real-world data sets. For full overview of related research, see the work by Post et al. [13].

One of the drawbacks of conventional streamlines is the lack of flow orientation (upstream vs. downstream direction) depicted in a still image. Our system incorporates an oriented streamline implementation. Oriented streamlines convey the downstream direction of the flow by varying the opacity as a function of particle trace evolution. In other words, the further downstream an integration path is traced, the higher the opacity of the streamline. This can be implemented by giving the streamlines a finite width, either automatically or through user-defined parameters, and using semi-transparent polygons in order to depict an oriented streamline (Figure 5, middle-left). Arrow heads could also be used to achieve the same effect. However, arrow head glyphs can more easily lead to visual clutter without careful treatment. The result is similar to that of OLIC (Oriented Line Integral Convolution) [21, 22]. One important difference is that OLIC is based on a traditionally slower approach derived from LIC and OLIC is more applicable to 2D flow rather than 3D.

For the case of unsteady flow, drawing a continuous particle path using only a single time step can be considered misleading. This is because no particle actually traces such a path. For the case of slices and surfaces, the visualization becomes even more problematic because a component of the vector field is taken away, namely that component orthogonal to the slice or surface, absent after a projection onto the slice or surface. One approach to handling this is through the use of streamlets (short streamlines) as illustrated in Figure 5 middle-right.

Streamcomets are an extension of the streamrunner [5]. The streamrunner addresses the problems of occlusion and scene complexity directly by giving the user control over the evolution of streamlines from seeding time until they terminate. A streamline may terminate when it reaches a boundary in the geometry, reaches a region of zero velocity, or reaches a maximum length set by the user. Streamcomets follow a very intuitive metaphor. They offer four interactive degrees of freedom as shown in Figure 6. Coupled with more interactive degrees of freedom, streamcomets offer the advantage of showing local flow direction and curvature for static images.

Another useful feature is the option of animating the streamcomets. Conceptually, animating the streamcomets such that the comet head position is automatically incremented along the streamline path, acts as a visual search function. The viewer is able to use the animation to search for optimal comet head positions. This is very useful when the user is not sure where to position the head, searching for interesting features in the flow field, or optimizing the other interactive DoFs. We emphasize the importance of the user's ability to resize the streamcomets along arbitrary dimensions when zooming in and out of the data sets.

Figure 5, left-to-right, shows the use of streamlines, oriented streamlines, streamlets and streamcomets all applied to the same data set. The data set in this case is simulation data coming from blood flow through an aneurysm. Note that these techniques highlight flow characteristics such as areas of divergence and convergence. Furthermore, they are well suited for 3D flow visualization of which we will see more in Section 7. Also We use a stippling approach to animate streamlines such that the downstream direction of the flow is depicted. See Laramee and Hauser [7] for more details about the implementation. [5]

7 Investigating Swirl and Tumble Flow

The VRVis Research Center collaborates with AVL (www.avl.com) in order to provide visualization solutions for analysis of their CFD simulation results. Previously, AVL engineers used a series of several color-mapped slices to assess and visualize the results of their CFD simulations. Here, new solutions for the visualization of CFD simulation data have been introduced. This section reports on the application of these techniques in addition to the more traditional approaches. We describe: (a) the application of different visualization techniques to specific application cases, (b) advantages and disadvantages of what these techniques offer, and (c) a comparison which may apply to other application cases. We also to give recommendations on when to use specific techniques and in which application scenario.

In the flow within a cylinder, we can distinguish between two types of motion: swirl flow commonly found in diesel engines and tumble flow commonly found in gas engines. In both cases, rotational motion occurs about an axis, though the position of the respective axis is different. In the case of swirl flow, the axis is more or less coincident with the cylinder axis, as shown in Figure 3, left. In the case of tumble (Figure 3, right), the rotation axis is perpendicular to the cylinder axis and more complex, thus making tumble flow more difficult to control than swirl flow.

In order to generate swirl or tumble motion, fluid enters the combustion chamber from the intake ports. Later on in the engine cycle, the kinetic energy associated with this motion is used to generate turbulence for mixing of fresh oxygen with evaporated fuel. The more turbulence generated, the better the mixture of air and fuel, and thus the more stable the combustion itself. By stable

[5]For supplementary images and MPEG animations, please visit:
`http://www.VRVis.at/ar3/pr2/geometricApproach/`

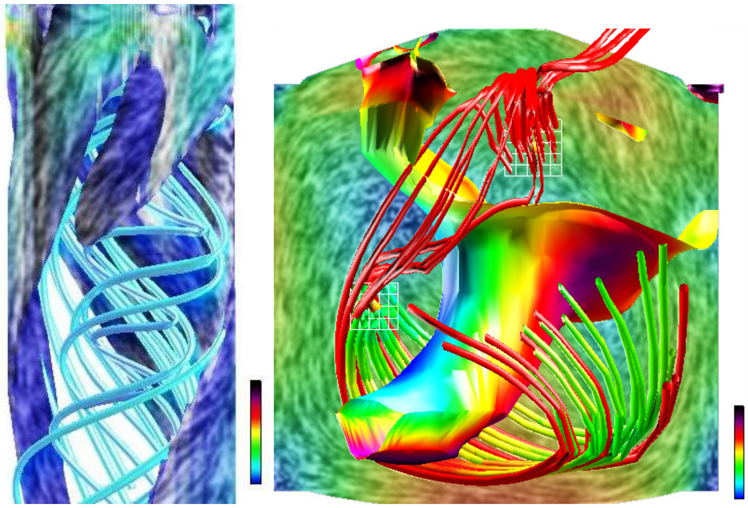

Figure 7: (left) Visualizing swirl flow using 3D streamlines and texture-based flow visualization on an isosurface. Compare with Figure 3, left. (right) An isosurface and 3D streamlines visualize tumble motion with the addition of texture-based flow visualization on a color-mapped slice. Compare to Figure 3, right.

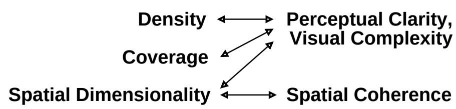

Figure 8: In general, trade-offs are made between the density, coverage, and spatial dimensionality of the visualization with that of perceptual clarity, visual complexity, and spatial coherence.

we mean achieving the same conditions for each engine cycle. Ideally, enough turbulent mixing is generated such that 100% of the fuel is burned. The swirl or tumble motion should be maximized to maximize turbulence. From the point of view of the mechanical engineers designing the intake ports, the ideal flow pattern leads to beneficial conditions namely, those outlined in Section 5. A controlled flow motion is used to get stable and reproducible conditions at each engine cycle.

We investigated two typical flow patterns from CFD using three classes of flow visualization techniques commonly available in 2D, 2.5D, and 3D[6]. By 2.5D we mean surfaces through 3D space. Figure 7 show some of the results of this investigation. Figure 7, left is a hybrid of multiple visualization approaches including 3D streamlines, isosurfacing, color-mapping, and texture-based flow visualization. The streamlines highlight the dominant characteristic structure of the flow inside the volume while the texturing applied to the isosurface points out some of the destructive areas of the flow at the top. Figure 7 helps verify that the overall behavior of the flow is characteristic of that of swirl motion. Figure 7, right is another hybrid of approaches including slicing with color-mapping and texture-based flow visualization applied, a pressure isosurface, and 3D streamlines seeded from two seeding planes. Figure 7, right shows that in this case the tumble motion axis is off-center, pointing down and to the left rather than straight out towards the reader in the ideal case.

Our side-by-side comparison of each flow visualization category illustrates that each has its respective advantages and disadvantages. Figure 8 summarizes some of the trade-offs that are made

[6]For supplementary material including high resolution images and MPEG animations, please visit: `http://www.VRVis.at/ar3/pr2/swirl-tumble/`

when visualizing swirl and tumble motion. For example, The more dense the visualization, generally the more difficult is to perceive the result. Thus a trade-off is often made between these two factors. As a result of these trade-offs, the flexible combinations of approaches offered by our system are good alternatives. For a more elaborate discussion of these results, please refer to Laramee et al [12].

8 Conclusions and Acknowledgements

The larger the data sets from CFD simulation become, the more useful scientific visualization is in order to gain insight into those results. In addition, no single "one-size-fits-all" approach exists, hence engineers require a range of tools in order to carry out their analysis, exploration, and presentation. We have presented a selection of recent advances in flow visualization. These recent advances have been applied to real-world applications in the field of CFD including the investigation and visualization of patterns of flow motion specific to automotive engineering. We have also discussed the benefits of these techniques and how engineers gain valuable insight into their CFD simulation results using this recently developed selection of tools. We would like to thank all those who have contributed to financing this research, including AVL (`www.avl.com`) and the Austrian governmental research program Kplus (`www.kplus.at`). All CFD simulation data is courtesy of AVL.

References

[1] B. Cabral and L. C. Leedom. Imaging Vector Fields Using Line Integral Convolution. In *Poceedings of ACM SIGGRAPH 1993*, Annual Conference Series, pages 263–272. ACM Press / ACM SIGGRAPH, 1993.

[2] H. Doleisch, M. Gasser, and H. Hauser. Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In *Proceedings of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239–248, May 2003.

[3] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser. Case Study: Visual Analysis of Complex, Time-Dependent Simulation Results of a Diesel Exhaust System. In *Proceedings of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2004)*, pages 91–96, May 2004.

[4] B. Jobard, G. Erlebacher, and Y. Hussaini. Lagrangian-Eulerian Advection of Noise and Dye Textures for Unsteady Flow Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):211–222, 2002.

[5] R. S. Laramee. Interactive 3D Flow Visualization Using a Streamrunner. In *CHI 2002, Conference on Human Factors in Computing Systems, Extended Abstracts*, pages 804–805. ACM SIGCHI, ACM Press, April 20-25 2002.

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

[7] R. S. Laramee and H. Hauser. Geometric Flow Visualization Techniques for CFD Simulation Data. In *VRVis Research Center Technical Report TR-VRVis-2004-029*, Vienna, Austria, Aug 2004.

[8] R. S. Laramee, H. Hauser, H. Doleisch, F. H. Post, B. Vrolijk, and D. Weiskopf. The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum*, 23(2):203–221, June 2004.

[9] R. S. Laramee, B. Jobard, and H. Hauser. Image Space Based Visualization of Unsteady Flow on Surfaces. In *Proceedings IEEE Visualization '03*, pages 131–138. IEEE Computer Society, 2003.

[10] R. S. Laramee, J. Schneider, and H. Hauser. Texture-Based Flow Visualization on Isosurfaces from Computational Fluid Dynamics. In *Data Visualization, The Joint Eurographics-IEEE TVCG Symposium on Visualization (VisSym '04)*, pages 85–90,342. Eurographics Association, 2004.

[11] R. S. Laramee, J. J. van Wijk, B. Jobard, and H. Hauser. ISA and IBFVS: Image Space Based Visualization of Flow on Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):637–648, November 2004.

[12] R. S. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating Swirl and Tumble Flow with a Comparison of Visuaization Techniques. In *Proceedings IEEE Visualization '04*, pages 51–58. IEEE Computer Society, 2004.

[13] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. Feature Extraction and Visualization of Flow Fields. In *Eurographics 2002 State-of-the-Art Reports*, pages 69–100. The Eurographics Association, 2–6 September 2002.

[14] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum*, 22(4):775–792, Dec. 2003.

[15] W. Schroeder, C. R. Volpe, and W. E. Lorensen. The Stream Polygon: A Technique for 3D Vector Field Visualization. In *Proceedings IEEE Visualization '91*, pages 126–132, 1991.

[16] M. Schulz, F. Reck, W. Bartelheimer, and T. Ertl. Interactive Visualization of Fluid Dynamics Simulations in Locally Refined Cartesian Grids. In *Proceedings IEEE Visualization '99*, pages 413–416, 1999.

[17] D. Stalling. LIC on Surfaces. In *Texture Synthesis with Line Integral Convolution*, pages 51–64. ACM SIGGRAPH 97, International Conference on Computer Graphics and Interactive Techniques, 1997.

[18] J. J. van Wijk. Spot noise-Texture Synthesis for Data Visualization. In Thomas W. Sederberg, editor, *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, volume 25, pages 309–318. ACM, 1991.

[19] J. J. van Wijk. Image Based Flow Visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002.

[20] J. J. van Wijk. Image Based Flow Visualization for Curved Surfaces. In *Proceedings IEEE Visualization '03*, pages 123–130. IEEE Computer Society, 2003.

[21] R. Wegenkittl and M. E. Gröller. Fast Oriented Line Integral Convolution for Vector Field Visualization via the Internet. In *Proceedings IEEE Visualization '97*, pages 309–316, October 19–24 1997.

[22] R. Wegenkittl, M. E. Gröller, and W. Purgathofer. Animating Flow Fields: Rendering of Oriented Line Integral Convolution. In *Computer Animation '97 Proceedings*, pages 15–21, June 1997.

[23] R. Westermann. The Rendering of Unstructured Grids Revisited. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualizatation (VisSym-01)*, pages 65–74. Springer-Verlag, May 28–30 2001.

[24] M. Zöckler, D. Stalling, and H. Hege. Interactive Visualization of 3D-Vector Fields Using Illuminated Streamlines. In *Proceedings IEEE Visualization '96*, pages 107–113, October 1996.