

# FIRST: A Flexible and Interactive Resampling Tool for CFD Simulation Data

Robert S. Laramee  
VRVis Research Center, Austria  
Laramee@VRVis.at  
www.VRVis.at

July 7, 2003

## Abstract

We introduce a flexible, variable resolution tool for interactive resampling of computational fluid dynamics (CFD) simulation data on versatile grids. The tool and coupled algorithm afford users precise control of glyph placement during vector field visualization via six interactive degrees of freedom. Other important characteristics of this method include: (1) an algorithm that resamples any unstructured grid onto any structured grid, (2) handles changes to underlying topology and geometry, (3) handles unstructured grids with holes and discontinuities, (4) does not rely on any pre-processing of the data, and (5) processes large numbers of unstructured grid cells efficiently. We believe this tool to be a valuable asset in the engineer's pursuit of understanding and visualizing the underlying flow field in CFD simulation results.

**Computers and Graphics Keywords:** Interaction techniques, Applications, Engineering, Graphics data structures and data types

**Additional Keywords:** interactive resampling, CFD simulation data, vector field visualization

## 1 Introduction -Applied Research

The demand for analysis and visualization solutions for CFD simulation data has grown rapidly in the last decade. This is due, in part, by the interest of manufacturers in minimizing the time taken for their production cycle. This objective is realized with the use of software simulation tools to analyze design decisions rather than constructing real, heavyweight objects. At the VRVis

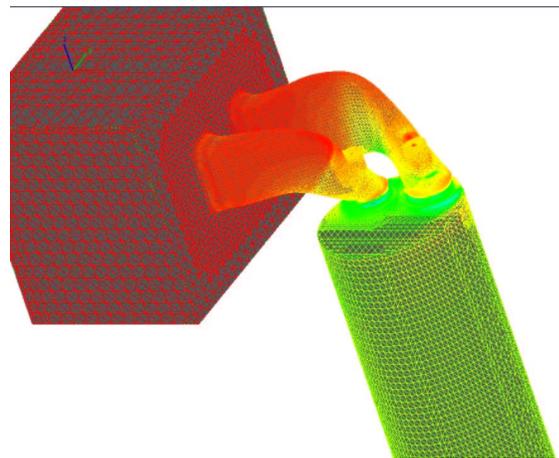


Figure 1: The CFD simulation grid of an intake port. The image illustrates the versatility of a typical, unstructured, CFD simulation grid containing a flow source on the left, two connecting pipes in the middle, two intake ports at the ends of the pipes, and a combustion chamber on the right.

Research Center we collaborate with AVL in order to provide visualization solutions for analysis of their CFD simulation result data. AVL ([www.avl.com](http://www.avl.com)) is an internationally recognized leader in providing automotive design and CFD simulation solutions to its partners in the automotive industry. AVL works with other internationally recognized companies such as Toyota, DaimlerChrysler – Mercedes-Benz, and Pierburg Instruments GmbH.

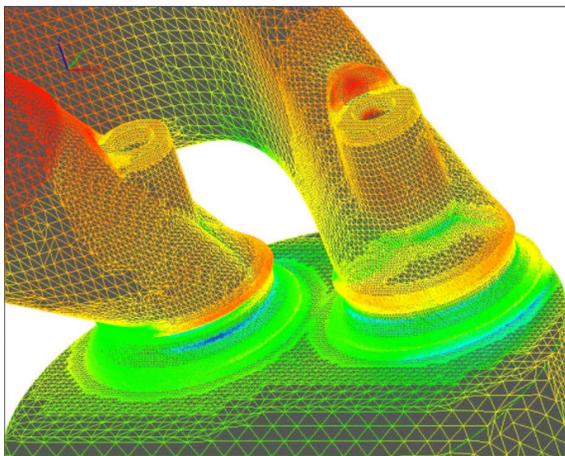


Figure 2: A close-up view of the surface of the intake port(s) shown in Figure 1. We can see many components, features, and multiple, adaptive resolution levels of unstructured grid cells. This versatility motivates interactive tools that have the ability to change their size, shape, orientation, and resolution to match the component(s) of the mesh currently undergoing user analysis.

## 1.1 Flexible Tools for Versatile Grids

Figure 1 shows two intake ports -small valves in a car engine that allow air into the engine's cylinders. The source of the flow is the cubical structure on the left. Flow travels through the connecting pipes in the middle, down through the intake ports, and is ignited in the combustion cylinder.

We use the term *versatile* to describe such meshes from CFD. By versatile we mean embracing a wide variety of components, features, and levels of resolution. Figure 2, a close up image of the intake ports, helps reveal the adaptive levels of resolution contained in the mesh.

We identify this aspect of CFD data sets because their versatility requires flexible analytic and visualization solutions. Ideally, the tools used to analyze and visualize these data sets should be flexible enough to adapt their *size, shape, orientation, and resolution* to fit the individual components of the data sets either automatically or through user-specified parameters. The resampling tool we present, called *FIRST* (a *Flexible and Interactive ReSampling Tool*), meets precisely these requirements.

## 1.2 A Wide Variety of Simulation Data Sets

The tools used to analyze and visualize these data sets have to address not only the versatility of individual geometries but also the *wide range* of simulation data sets that undergo analysis. AVL has a large, varied collection of data sets ranging from small geometries such as small fluid conduits and cylinders to mid-range size geometries such as cooling jackets, intake manifolds, catalytic converters, to large geometries such as automotive cabin interiors and automotive exteriors. The geometric sizes of these grids, as well as the sizes of the underlying polygons, differ by *six* or more orders of magnitude. Furthermore, we speculate that this difference will only increase in the future.

The unpredictable nature of both the data sets and the individual components the engineer may choose to analyze provide strong motivation for tools that offer many options and, thus, higher levels of control to the user. The tools used to visualize the simulation results also need to span this range of sizes correspondingly. The *FIRST* tool described here offers six interactive *degrees of freedom* (DoF) with the goal of addressing the nature of both the wide range of data sets and the interests of engineers.

## 1.3 Perceptual Problems in FlowViz

When analyzing the results of a CFD simulation, engineers are interested in the option of examining planar vector fields with normal components [15]. An obvious approach to this problem is to place glyphs (e.g. directed lines, arrows, cones etc.) oriented in the direction of the vector components, at selected points of the vector field. Ideally, the length of the glyphs should be equal to the normal of the vector, but often a scaling constant is added to improve the visualization result. This method (called hedgehog visualization) [16] is illustrated in Figure 3.

There are multiple drawbacks to this approach. *Perceptual problems* such as visual complexity and occlusion can result. Figure 3 shows a 2D slice taken through the intake port data set with the surface also shown as semi-transparent context information. Here, the glyphs are either too big, resulting in occlusion, or too small to clearly indicate directional information. Another problem arises due to the number and *placement* the glyphs. Critical points might be missed if they are either oc-

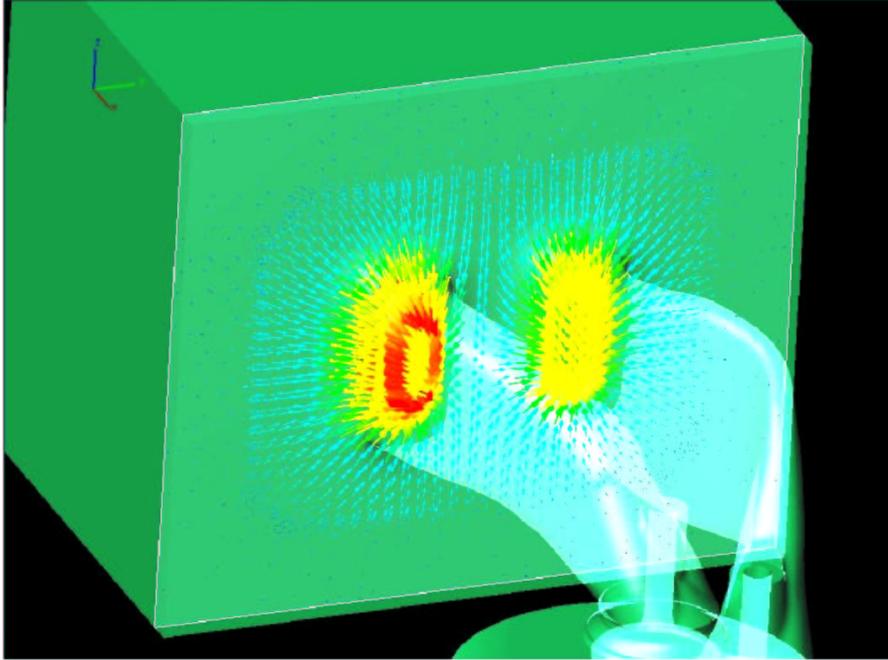


Figure 3: Visualizing the direction of the flow, including the normal component, using the classic hedgehog visualization technique can result in perceptual problems such as visual complexity and occlusion. Here the problems are illustrated in a slice through the intake port data set with semi-transparent context information.

cluded by too many glyphs placed in the wrong areas or if an insufficient number of glyphs is chosen in certain areas. FIRST solves both the perceptual and 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 rest of the paper is organized as follows: Section 2 describes related research work. Section 3 describes the contribution of this work. The resampling algorithm and user options are described in Section 4. Timing and results are presented in Section 5. Conclusions are drawn in Section 6.

## 2 Related Work

When describing grids, we follow the terminology of Yagel et al. [20]. *Structured grids*, including *cartesian grids*, *regular grids*, and *rectilinear grids*, all maintain an implicit neighborhood connectivity (Figure 4(a-d)), i.e. the position of grid cells can be computed, rather

than stored explicitly. In contrast, grids whose neighbors must be explicitly stored are categorized as *unstructured* grids. This distinction is important in understanding our claim that the resampling technique here can be used to represent *any* unstructured grid with *any* structured grid. We demonstrate this assertion with implementations of the resampler from unstructured triangular grids (Figure 4(f)) to cartesian and polar grids (Figure 4(a-d)). The basis for this claim hinges on the supposition that cell positions in structured grids can be computed.

The literature dedicated to the topic of resampling spans multiple, sometimes disparate, goals. While we designed our resampler with the goal of a high level of user-control, other resampling techniques focus on the goals of volume rendering, surface reconstruction, and accurate surface normal representation. In contrast with these approaches, we concentrate on 2D slices through a 3D mesh from CFD. We contrast other resampling goals with ours.

Resampling techniques whose goal is to achieve faster

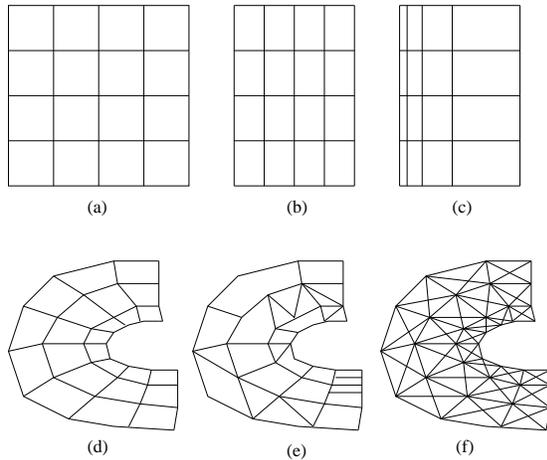


Figure 4: (a) cartesian grid, (b) regular grid, (c) general rectilinear grid, (d) structured grid, (e) unstructured grid, and (f) unstructured triangular grid [20]. The FIRST technique described in Section 4 can resample any unstructured grid onto any other structured grid.

volume rendering speeds may make use of the hardware capabilities offered by modern graphics cards.

Westermann presents a resampling technique for resampling scalar fields given on unstructured tetrahedral grids [19]. The goal of this research is to bring direct volume rendering towards interactive frame rates with the assistance of graphics hardware. Determining the visibility ordering of grid elements is a major challenge in this scenario. While the goal of this approach is dissimilar to ours, one similarity worthy of note is that Westermann’s technique can be used to display time-dependent unstructured grids with changing geometry and topology.

Weiler and Ertl [18] also present three resampling approaches and contrast the results with the work of Westermann [19]. Their goal is to minimize performance time via balancing the computing workload between the native processor and the graphics board processor.

In addition to resampling techniques for volume rendering, some resampling approaches are targeted at representing surfaces. Botch and Kobbelt [3] present a technique to resample feature regions of a given triangle mesh with the goal of reducing aliasing artifacts on surfaces, i.e., surface anti-aliasing. Similar the work here, there is a strong interaction component to their work.

The user chooses the areas on the surface to which the surface normal anti-aliasing algorithm is applied. To remodel the structure of a sample CFD grid can take an interactive session lasting one hour.

Rocchini et al. [14] present an algorithm whose goal is the removal of small topological inconsistencies and high frequency details from surfaces. One of their goals is the simplification of huge meshes, i.e., meshes composed of millions of faces.

It is also worthy of note that some resampling algorithms from the image processing domain magnify or minify the original image via sampling at regular intervals [11]. Our approach samples the original data irregular intervals and generates an evenly-spaced representation.

### 3 Interactive Visualization and Analysis

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 the engineers at AVL require interactive visualization solutions. We speculate that this is 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 with measured, experimental results. FIRST has the following advantages:

1. 6 interactive DoFs: 3 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 (Section 4)
3. resamples any unstructured grid (Figure 4(f)) onto any structured grid (Figure 4(a-d))
4. handles unstructured grids with holes and discontinuities (Figure 11)
5. does not rely on any pre-processing of the data

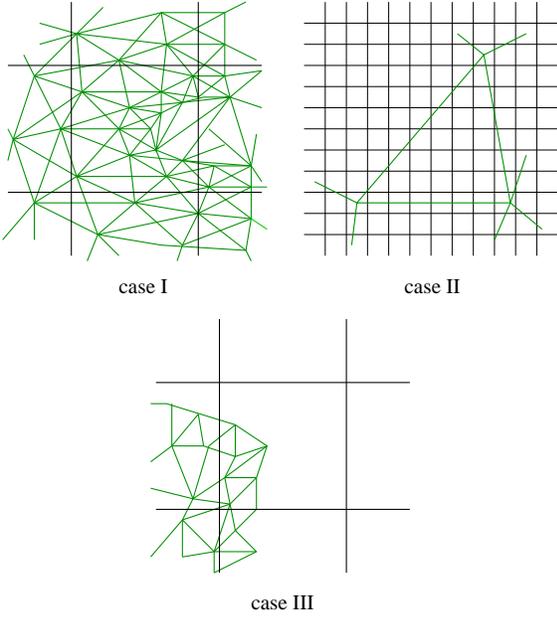


Figure 5: The FIRST algorithm handles 3 cases: (I) when several scalene triangles influence one resampler cell, (II) when one triangle influences several resampler cells, and (III) when several triangles influence one resampler cell, however, the resampler cell is not rendered. The resampler cells are outlined in black while the original unstructured grid is drawn colored.

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

The resampler we describe provides flexible user-interaction capabilities beyond those offered by these previous methods. Also, the algorithm operates on a per-unstructured-polygon basis, making it suitable for parallelization.

## 4 FIRST: A Flexible and Interactive ReSampling Tool

Because we are dealing with unstructured, adaptive resolution grids and because the users require the option of fine resampling, the resampling algorithm presented here is required to handle a minimum of three cases

(Figure 5): (I) the underlying unstructured grid cells are generally smaller than the structured resampler cells, (II) the unstructured grid cells are generally larger than the resampler cells, and (III) some unstructured grid cells are inside a resampler cell, however, the resampler cell should not be rendered. A resampler cell is rendered only if its center is covered by the underlying unstructured grid. In this paper we use the term *resampler cell* to refer to a structured grid cell that represents, or summarizes, the original data.

### 4.1 Algorithm and Implementation

It is important to note that, unlike many other applications where vector field values might be stored at triangle vertices, the vector field here is associated with the centers of the triangles in the slice mesh. By slice mesh, we mean the user-defined 2D slice through the original 3D mesh. Engineers require the option of visualizing the *original* values resulting from the simulation rather than interpolated values at the triangle vertices. Storing the CFD simulation data values at the center of the grid cells may result from using a finite volume method to solve for the flow quantities [5].

We have, however, implemented the option of viewing the interpolated scalar values at the triangle vertices. We note that the algorithm here is easily modified to handle unstructured grids where the scalar field is stored at the vertices.

#### 4.1.1 Algorithm Overview

To describe the algorithm, we introduce the notion of *influence*, similar to the notion introduced by Rhodes et al. [12, 13]. A triangle *influences* a resampler cell if the triangle contributes to the summary (or resampled) data of a structured grid cell. This happens when a triangle is in the vicinity of a resampler cell. In short, the resampling algorithm traverses multiple loops of resampler cells in the neighborhood of each unstructured cell in order to see if they are influenced by the original unstructured grid cells. In the most common case (Figure 5, Case I), we can think of a resampler cell as summarizing multiple underlying unstructured grid cells.

#### 4.1.2 Algorithm Detail

The following high level pseudo-code summarizes the resampling algorithm (Figure 6):

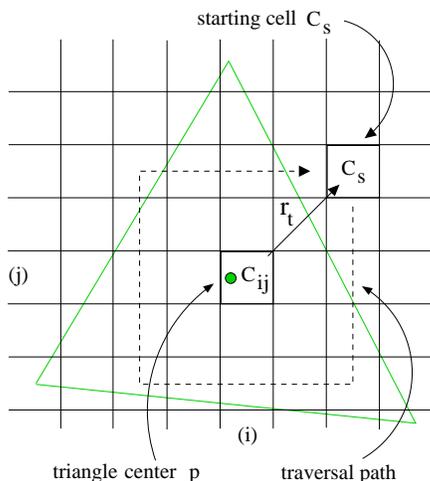


Figure 6: A schematic of the resampling algorithm. A loop of structured grid cells is tested for influence by the unstructured grid cell.

**resample():**

```

FOR each scalene triangle,  $T$ 
  compute center point,  $p$ , of  $T$ 
  compute resampler cell index,  $C_{ij}(p)$ 
  int  $r_t = 0$ 
  bool influence = FALSE
  do
    influence = computeInfluence( $T$ ,  $r_t$ )
     $r_t++$ 
  while (influence == TRUE)

```

The resampling algorithm requires exactly *one* visit to each unstructured grid cell. In the **resample()** method, first the resampler cell,  $C_{i,j}$  in Figure 6, surrounding the triangle center is computed. Next, a loop of resampler cells is traversed clockwise, starting with cell,  $C_s$ . The size of the loop traversed is determined by a topological radius,  $r_t$ , starting with a loop radius of 0, and incremented until the triangle,  $T$ , is found to have no influence on the loop of resampler cells.

**computeInfluence(triangle  $T$ , index  $r_t$ ):**

```

  compute resampler cell index,  $C_s(r_t)$ 
  bool triangleInfluence = FALSE
  FOR each resampler cell in loop
    IF (testInfluenceOfTriangle( $T$ ,  $C_s$ ))
      triangleInfluence = TRUE
  return triangleInfluence

```

In the **computeInfluence()** method, each resampler cell in the loop is tested for influence by the current triangle,  $T$ , in the unstructured grid. A sample traversal loop is shown in Figure 6. A triangle influences a loop when any resampler cell in the loop tests positive for influence.

**testInfluenceOfTriangle(triangle  $T$ , cell  $C_s$ ):**

```

bool influence = FALSE
IF center( $C_s$ ) bounded by  $T$ 
  influence = TRUE // Cases I & III
  render  $C_s$  = TRUE
  add  $V_T$  to  $C_s$ 
IF center( $T$ ) bounded by  $C_s$ 
  influence = TRUE // Case II
  add  $V_T$  to  $C_s$ 
return influence

```

In the **testInfluenceOfTriangle()** method, two simple tests are performed: (1) if the resampler cell center is bounded by the triangle and, (2) if the triangle center is bounded by the resampler cell. If the resampler cell center is bounded by the triangle, the velocity vector of the unstructured, scalene triangle,  $V_T$  is added to the resampler cell's summary vector and the resampler cell is rendered (during the rendering pass). By rendered, we mean simply that a vector glyph is drawn at the center of the resampler cell. If the triangle center is bounded by the resampler cell, the velocity vector of the unstructured, scalene triangle,  $V_T$  is added to the resampler cell's summary vector; however, the resampler cell is not necessarily rendered. These two tests cover all three cases shown in Figure 5. It is helpful to note that a triangle cell,  $T$ , may influence a resampler cell,  $C_s$ , without  $C_s$  being rendered.

In the actual implementation, the test to see if a resampler cell center falls within the bounds of an unstructured, scalene triangle includes an explicit test of whether the resampler cell center falls precisely on the edge of a triangle. This is because often, the triangles in the mesh are derived from highly structured portions of CFD components –such as the flow source shown in the left of Figure 1.

The resampling algorithm generalizes to other types of unstructured grids, not just those composed of scalene triangles. The reason is two-fold. First, the algorithm operates on a per-unstructured-grid-cell basis making any search for unstructured grid cells unnecessary. Second, given any location in space, such as the center point of an unstructured grid cell, it is easy to compute

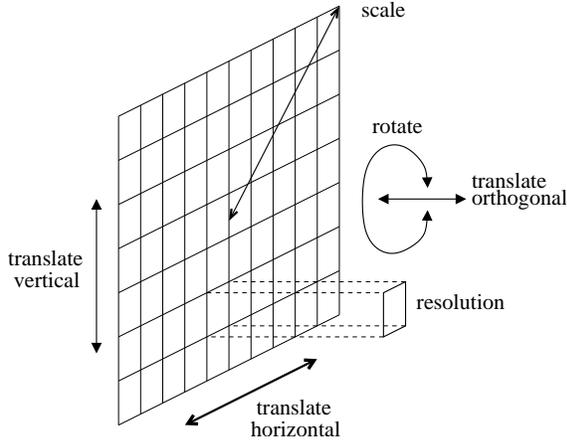


Figure 7: FIRST is a 6 DoF tool: (1-3) three translations, (4) a rotation, (5) a scale, and (6) resolution.

a shell of regular grid cells around that location. The position of regular grid cells can be computed by definition.

Some nice consequences of the algorithm are that (1) *no* special boundary conditions are checked during the computation and (2) *no* knowledge of the underlying grid's resolution is required. In other words, FIRST does not have to make any distinction between the three cases shown in Figure 5. This lays the groundwork for the claim that the algorithm is characterized by ease of implementation.

## 4.2 Interactive Resampling Options

FIRST provides the user with 6 interactive degrees of freedom (Figure 7): (1-3) translation along three dimensions, (4) rotation about the center, (5) scaling, and (6) the resolution of resampling cells, i.e., cells/m<sup>2</sup>.

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. This is important with respect to addressing the versatility aspect of our CFD simulation data sets. Furthermore, requiring the user to manually adjust the size of the resampling grid would slow down

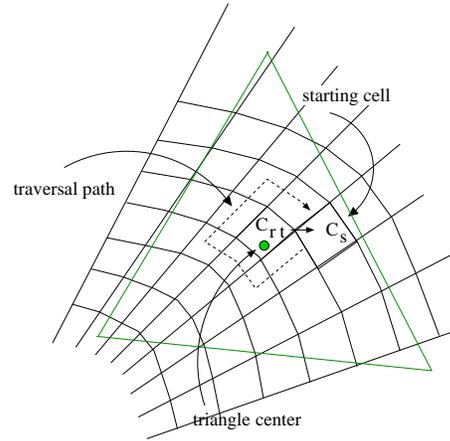


Figure 8: The resampling algorithm applied to a polar resampling grid. Polar resampling is a natural application for many of the cylindrical components common in CFD simulation models.

the visualization and analysis process considerably.

In addition to the ability to define slices parallel to the X-Y, X-Z, and Y-Z coordinate planes, the user may also define arbitrary cutting planes in 3D space. The user may click on any three locations on the CFD grid and the three points are used to define an interactive slice. For this capability, the unstructured grid cells are sorted into an octree so coordinate searching is fast. All resampling options are available for arbitrary slices.

## 4.3 Discrete Polar Resampling

The FIRST algorithm detailed in Section 4.1 easily generalizes to any structured grid. Figure 6 illustrates the algorithm applied to a cartesian grid while Figure 8 illustrates the same algorithm applied to a polar grid. The only difference is the use of polar coordinates instead of cartesian coordinates. The traversal path is a function of both radius and degrees. And, the resolution of the polar grid is specified in terms of slices and rings.

We have implemented both cartesian and polar versions of the algorithm, both with identical user options. The reason for offering polar resampling lies in the cylindrical structure of many of the components from common CFD simulation models. The combustion chamber shown in Figure 1 is a common example of such a component.

## 4.4 Interactive Visualization Options

Multiple visualization options are associated with FIRST including: (1) the ability to toggle wire-frame or semi-transparent context information, i.e., the 3D grid from which slices are defined and (2) several interactive glyph options. This includes any combination of glyph shapes, e.g., cones vs. arrows with rendering options, e.g., wire-frame vs. solid. The user also defines the glyph scaling options that may be used to avoid the “visualization lie” [17]. Several other FlowViz options are provided directly on unstructured grids [6, 7].

## 4.5 Speed vs. Accuracy

If we take a closer look at the resampling algorithm, we note that we make a trade-off between speed and accuracy. A straightforward averaging scheme is used to summarize the vectors influencing a resampler cell. When the resolution of the resampling grid approaches or exceeds that of the underlying, unstructured grid, our algorithm amounts to nearest neighbor interpolation scheme, or a box filter when described in the frequency domain [1, 2]. Reconstructing the flow field at a higher resolution than the original simulation results is orthogonal to the goal of this tool. Users make this trade-off for interactivity because of their tasks related to searching the flow field. Engineers spend time visually searching for areas of extreme pressure, symmetries in the vector field, and critical points. We speculate that users require searching tasks to be as fast as possible. Only after the features of interest are found is more detail required. We offer 3 options to users requiring higher accuracy: (1) The user may interactively increase the resolution of the resampling grid, thus increasing its accuracy. (2) Users may view the unstructured grid directly, displaying only the original values resulting from the simulation. (3) We have also implemented a *dynamically annotated user dialog* not dissimilar to that of Loughlin and Hughes [8]. This feature allows the user to click anywhere on the slice of interest. Then a dialog will automatically display the original simulation result values of each variate (e.g. velocity, pressure, temperature, etc.).

## 5 Results and Discussion

Figures 9, 10, and 11 illustrate an example of viewing a vector field with normal component using the resam-

pler.<sup>1</sup> We can compare Figure 9 with Figure 3 and see that viewing the normal components of the vector field is easier using summary vectors rather than the brute force hedgehog technique. Perceptual problems such as occlusion and visual complexity have been greatly reduced. Furthermore, visualization algorithms operating on regular grids are generally faster than on unstructured grids because particle tracing algorithms no longer require neighbor-finding techniques and the more costly flow reconstruction methodology.

Figure 11 illustrates our technique on a mesh with discontinuities. The discontinuities are two gaps 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. A close-up view of one of the discontinuities is provided.

In order to achieve our goal of implementing a flexible

number of polygons	resolution of resampler cells	frames per second
1,552	5 × 5	10.0-11.0
	10 × 10	5.0-6.0
	50 × 50	1.0-6.0
	100 × 100	0.2-0.3
3,157	5 × 5	6.0-7.0
	10 × 10	5.0-6.0
	50 × 50	1.0-3.0
	100 × 100	0.2-1.0
14,085	5 × 5	1.0-2.0
	10 × 10	0.7-1.3
	50 × 50	0.17-0.30
	100 × 100	0.07-0.10

Table 1: Sample frame rates for the resampling algorithm. Performance was evaluated on a machine running Red Hat Linux 7.2 with a 1 GHz Pentium III dual processor and 512 Mbytes of RAM. Note that the frame rate also varies as a result of caching.

and interactive resampler, interactive frame rates must be achievable. We have tested our algorithm on typical slices of CFD simulation data. Some sample performance times (measured in frames per second) are shown in Table 1. The performance times depend on both the number of scalene triangles composing the

<sup>1</sup>Supplementary animations of the resampler can be found at <http://www.VRVis.at/ar3/pr2/resampler/>

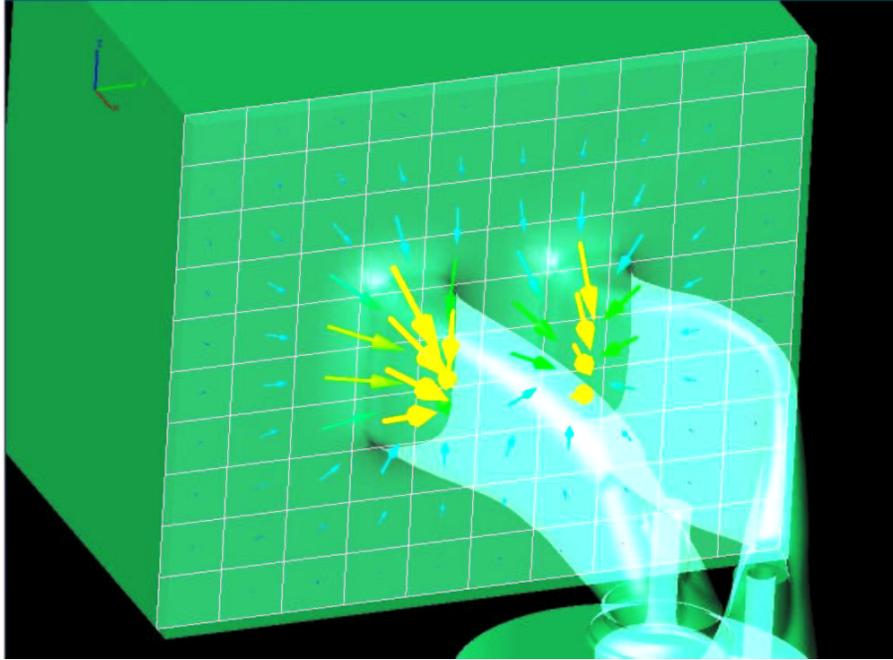


Figure 9: Visualizing the direction of the flow, including the normal component, using FIRST. (Compare with Figure 3) The resampling grid is outlined in white.

slice and the number of resampler cells. We can see from the table that this resampling algorithm does support interactive frame rates with several frames per second. We note that in our application a resampling grid of 10,000 cells is not practical due to perceptual problems as well as pixel resolution limits. In this case vector glyphs typically cover only one or two pixels. Also, a resampling grid with such a high resolution defeats one purpose for which it was intended, to provide a structured *summary* of the underlying data. This grid size is evaluated here in order to show the upper limits of the resampling algorithm. Grid sizes between  $10 \times 10$  (like that of Figure 9) and  $50 \times 50$  are much more likely in our case. In practice we find that rendering the geometry is often the performance bottleneck.

Another result of the resampling algorithm is the ability to visualize unsteady flow with the normal component of the slice through which the flow is passing. In our review of the flow visualization literature [9, 10], we see several flow visualization solutions for 2D, steady-state, vector fields. However, we see no unsteady solutions that include all three vector field components at *interactive* frame rates. In the case of Scheuermann

et. al. [15], theirs is a visualization solution for steady-state flow, i.e., instantaneous flow from one time step. In our case, we visualize the flow, with normal component, over time, also at interactive frame rates.

We have implemented an animation control that allows the user to load several time steps of CFD simulation result data, and “play them back” at user-specified time intervals. The user is allowed to pause, rewind, forward, and stop the animation using controls similar to any home media player. We combine this animation control with the resampling feature for an intuitive visualization of unsteady flow with normal components.

## 6 Conclusions and Future Work

We believe FIRST to be a valuable asset in the engineer’s pursuit of understanding the underlying flow field in their CFD simulation models. FIRST reduces perceptual problems such as occlusion and visual complexity when visualizing a vector field, steady or unsteady, with all three vector field components. Searching for flow features by sliding a slice through the 3D volume

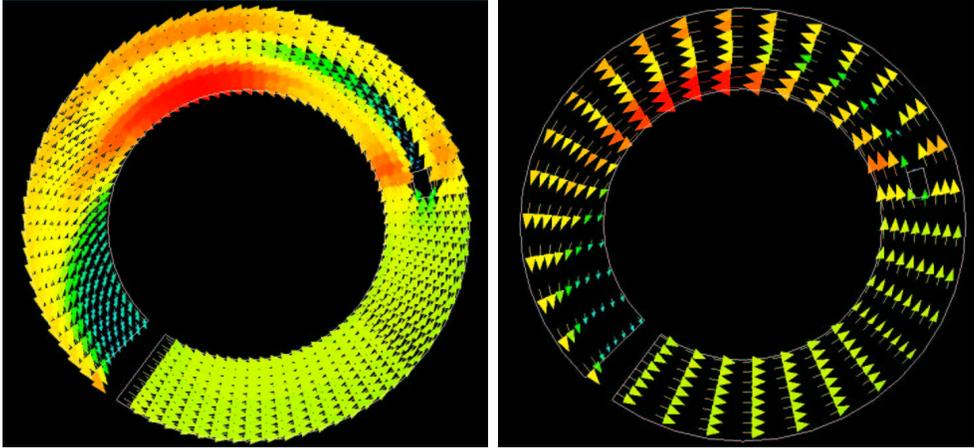


Figure 10: (left) Visualizing the direction of the flow through a cylindrical slice from CFD using a brute force hedgehog technique. (right) Visualizing the direction of the flow, including the normal component, using the polar resampling option of FIRST.

is also hastened.

Future work can take multiple directions including the addition of more user-interaction controls, extending the algorithm to 3D, and porting the algorithm to Java for inclusion into the VisAD open source, scientific visualization library [4].

Future work includes the addition of more user-

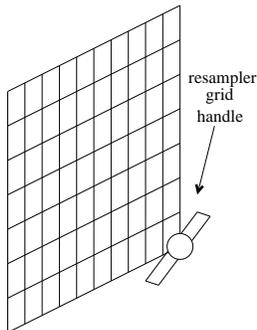


Figure 12: User interaction can include handles that are attached to the resampling grid. These handles, while attached to the grid, may be placed arbitrarily via the common click-and-drag paradigm resulting in user-specified grid placement or user-specified scaling.

interaction techniques. One such technique involves the addition of handles attached to the resampling grid

(Figure 12). The user can click-and-drag a handle and change the location or size of the resampling grid. Two different strategies are possible: (1) where the other corners of the resampling grid are held fixed while one handle is dragged resulting in a change of size or (2) where the other corners move with the grid retaining a constant size but specifying a new location.

One of goals is to extend the algorithm to handle 3D simulation data. We believe that the algorithm outlined in Section 4 can be extended to 3D in a very straightforward manner. The only difference would be a modification to the traversal part of the algorithm to include multiple layers. We realize, however, that performance time will be biggest challenge in realizing this goal.

## 7 Acknowledgements

The author(s) thank all those who have contributed to this research including AVL ([www.avl.com](http://www.avl.com)) and the Austrian governmental research program called Kplus ([www.kplus.at](http://www.kplus.at)). We extend a special thanks to Helwig Hauser and Zoltan Konyha of the VRVis Center For Virtual Reality and Visualization ([www.vrvis.at](http://www.vrvis.at)) for their valuable feedback. We thank Colin Ware at the University of New Hampshire for his valuable contributions and feedback. And finally we thank Thomas Gößler of AVL ([www.avl.com](http://www.avl.com)) for his technical contributions. All CFD simulation data shown in this paper has been provided courtesy of AVL.

## References

- [1] James F. Blinn. Jim Blinn's Corner: Dirty Pixels. *IEEE Computer Graphics and Applications*, 9(4):100–105, July 1989.
- [2] James F. Blinn. Jim Blinn's Corner: Return of the Jaggy. *IEEE Computer Graphics and Applications*, 9(2):82–89, March 1989.
- [3] M. Botsch and L. P. Kobbelt. Resampling Feature and Blend Regions in Polygonal Meshes for Surface Anti-Aliasing. In *Eurographics 2001 Proceedings*, volume 20(3) of *Computer Graphics Forum*, pages 402–410. Blackwell Publishing, 2001.
- [4] W. Hibbard. Connecting People to Computations and People to People. *Computer Graphics*, 32(3):10–12, 1998.
- [5] D. A. Lane. *Scientific Visualization of Large-Scale Unsteady Fluid Flows*, chapter 5, pages 125–145. Scientific Visualization: Overviews, Methodologies, and Techniques. IEEE Computer Science Press, Los Alamitos, 1997.
- [6] 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, Minneapolis, Minnesota, April 20–25 2002. ACM SIGCHI, ACM Press.
- [7] R. S. Laramee, B. Jobard., and H. Hauser. Image Space Based Visualization of Unsteady Flow on Surfaces. In *Proceedings of the IEEE Conference on Visualization (Vis 03)*. IEEE, 2003. forthcoming.
- [8] M. M. Loughlin and J. F. Hughes. An Annotation System for 3D Fluid Flow Visualization. In *Proceedings of the IEEE Conference on Visualization*, pages 273–280, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press.
- [9] 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, Saarbrücken Germany, 2–6 September 2002. The Eurographics Association.
- [10] 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), 2003. forthcoming.
- [11] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing. Principles, Algorithms, and Applications*. Prentice Hall International, third edition, 1996.
- [12] P. J. Rhodes, R. D. Bergeron, and T. M. Sparr. A Data Model for Multiresolution Scientific Data Environments. In *NSF/DoE Lake Tahoe Workshop on Hierarchical Approximation and Geometrical Methods for Scientific Visualization*, Tahoe City, California, October 15–17 2000.
- [13] P. J. Rhodes, R. D. Bergeron, and T. M. Sparr. A Data Model for Distributed Multiresolution Multisource Scientific Data. In G. Farin, H. Hagen, and B. Hamann, editors, *Hierarchical and Geometrical Methods in Scientific Visualization*, Heidelberg, Germany, 2002.
- [14] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pinci, and R. Scopigno. Marching intersections: an efficient resampling algorithm for surface management. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI-01)*, pages 296–305, Los Alamitos, CA, May 7–11 2001. IEEE Computer Society.
- [15] G. Scheuermann, H. Burbach, and H. Hagen. Visualizing planar vector fields with normal component using line integral convolution. In *IEEE Visualization '99*, pages 255–262, San Francisco, 1999. IEEE.
- [16] W. J. Schroeder, K. M. Martin, and W. E. Lorensen. *The Visualization Toolkit*. Prentice-Hall, Upper Saddle River, NJ 07458, USA, 2nd edition, 1998.
- [17] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.
- [18] M. Weiler and T. Ertl. Hardware-Software-Balanced Resampling for the Interactive Visualization of Unstructured Grids. In *IEEE Visualization 2001*, pages 199–206, 2001.
- [19] R. Westermann. The rendering of unstructured grids revisited. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym-01)*, pages 65–74, Wien, Austria, May 28–30 2001. Springer-Verlag.
- [20] R. Yagel, D. M. Reed, A. Law, P. Shih, and N. Sha-reef. Hardware assisted volume rendering of unstructured grids by incremental slicing. In *Proceedings 1996 Symposium on Volume Visualization*, pages 55–62, September 1996.

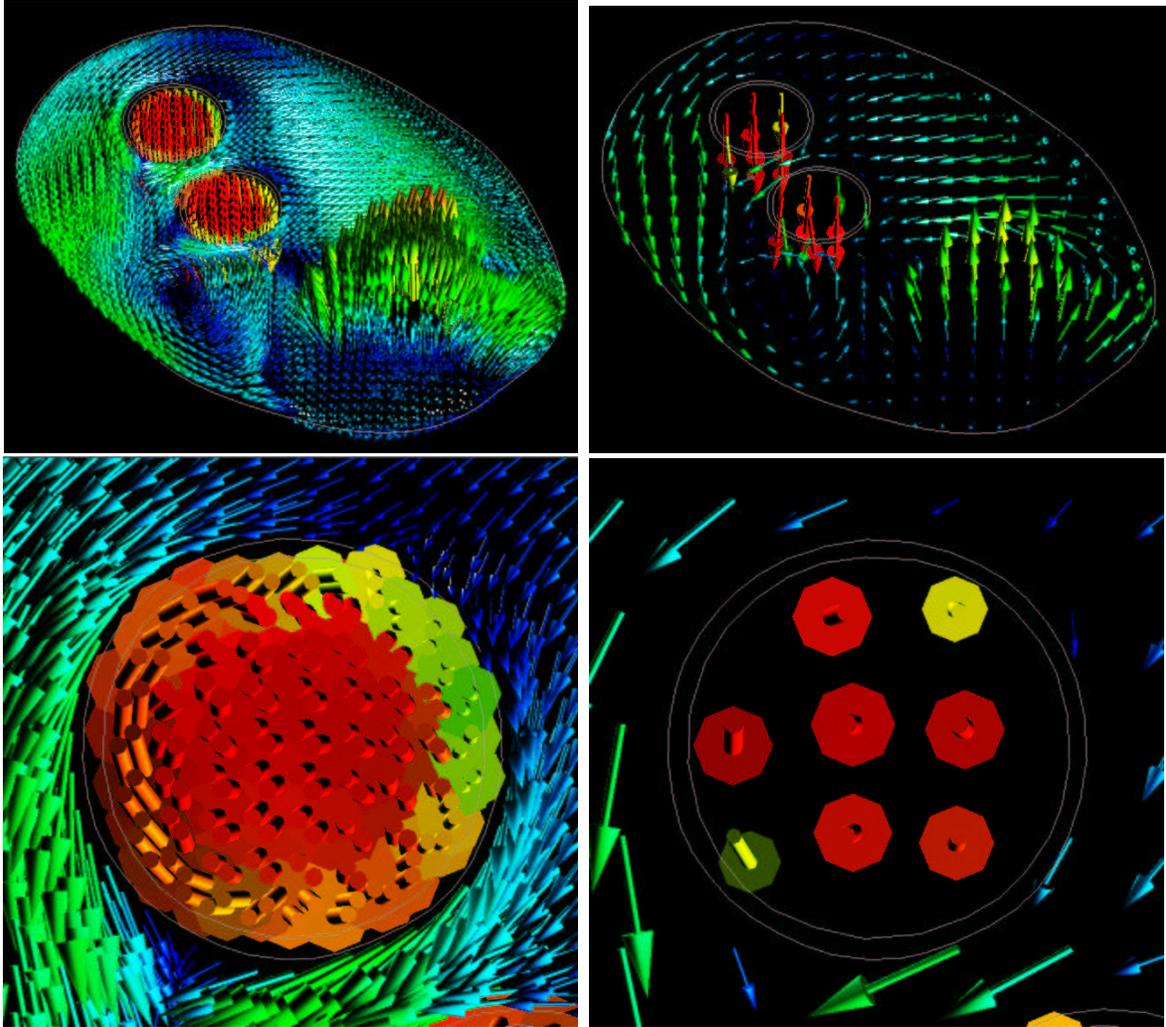


Figure 11: (top, left) A slice mesh with discontinuities -two gaps in the shape of rings, using hedgehog visualization (top, right) The same slice resampled onto a regular grid, (bottom, left) a close-up view of one of the rings causing a discontinuity, with hedgehog visualization (bottom, right) the same close-up view with resampling enabled