

# Asymmetric Tensor Visualization with Glyph and Hyperstreamline Placement on 2D Manifolds

Darrel Palke, Guoning Chen, Zhongzang Lin, Harry Yeh, Robert S. Laramée, IEEE Member and Eugene Zhang, IEEE Member

**Abstract**—Asymmetric tensor fields present new challenges for visualization techniques such as hyperstreamline placement and glyph packing. This is because the physical behaviors of the tensors are fundamentally different inside *real domains* where eigenvalues are real and *complex domains* where eigenvalues are complex. We present a *hybrid* visualization approach in which hyperstreamlines are used to illustrate the tensors in the real domains while glyphs are employed for complex domains. This enables an effective visualization of the flow patterns everywhere and also provides a more intuitive illustration of elliptical flow patterns in the complex domains. The choice of the types of representation for different types of domains is motivated by the physical interpretation of asymmetric tensors in the context of fluid mechanics, i.e., when the tensor field is the velocity gradient tensor. In addition, we encode the tensor magnitude to the size of the glyphs and density of hyperstreamlines. We demonstrate the effectiveness of our visualization techniques with real-world engine simulation data.

**Index Terms**—Tensor field visualization, asymmetric tensor fields, flow visualization, glyph packing, hyperstreamline placement, non-uniform density.



## 1 INTRODUCTION

ASYMMETRIC tensor fields appear in a wide range of engineering applications such as solid and fluid mechanics, structural engineering, and medical imaging. For example, in flow visualization the velocity gradient tensor, an asymmetric tensor field, describes non-translational motions in fluid parcels such as rotation, stretching, and volume changes that cannot be easily inferred from direct visualization of the velocity vector field [1]. Consequently, effective visualization techniques for asymmetric tensor fields can potentially benefit many applications in the aforementioned domains.

Despite the potential of asymmetric tensor field visualization, there has been relatively little work in this area. Most existing tensor field visualization techniques focus on *symmetric* tensors and use either glyphs or hyperstreamlines that follow either the major or minor eigenvectors. Due to some fundamental differences between symmetric and *asymmetric* tensors, these techniques cannot be easily adapted to the visualization of the latter. For example, symmetric tensors always have

real eigenvalues while asymmetric tensors can have *complex* eigenvalues. Consequently, hyperstreamline-based tensor field visualization techniques [2] do not apply to regions in the domain where eigenvalues are complex (*complex domain*). Furthermore, the major and minor eigenvectors of a symmetric tensor are mutually perpendicular except at the *degenerate points* where they are not well-defined. In contrast, in *real domains* where eigenvectors are real-valued, the major and minor eigenvectors are typically not mutually perpendicular. Consequently, hyperstreamlines following the major eigenvector field cannot be used to infer the minor eigenvector field.

In their pioneering work, Zheng and Pang [3] introduce the concept of *dual-eigenvectors* which can be used to visualize a 2D asymmetric tensor field inside complex domains. They also incorporate into *tensor field topology* for the boundary curves between complex domains and real domains. Along these curves the tensor field has two equal eigenvalues. Consequently, these curves are referred to as *degenerate curves* as they are extensions of degenerate points for symmetric tensor fields. Zheng and Pang also visualize asymmetric tensor fields by using hyperstreamlines that follow either the major or the minor eigenvectors in the real domain and the *major dual-eigenvectors* in the complex domain.

Later, Zhang et al. [1] extend this analysis by introducing the concepts of *eigenvalue manifold* and *eigenvector manifold*. With these manifolds, they develop analysis of 2D asymmetric tensor fields as well as provide physical interpretation of this analysis when the tensor is the gradient of the velocity vector field. In this context, they note that the eigenvectors and the dual-eigenvectors can describe local motions of the fluid parcels which are elliptical in the complex domains and hyperbolic in the real domains. Zhang et al. also introduce the concept of *pseudo-eigenvectors* which they use to infer the elliptical patterns in the complex domains. By intersecting

*D. Palke is with the School of Electrical Engineering and Computer Science, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331. Email: palke@eecs.oregonstate.edu.*

*G. Chen is with the Scientific Computing and Imaging (SCI) institute, the University of Utah, 72 S Central Campus Drive, Room 3750, Salt Lake City, UT 84112. Email: chengu@sci.utah.edu.*

*Z. Lin is with the School of Electrical Engineering and Computer Science, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331. Email: lin@eecs.oregonstate.edu.*

*H. Yeh is a Professor in Fluid Mechanics of the School of Civil Engineering, Oregon State University, 220 Owen Hall, Corvallis, OR 97331. Email: harry@engr.orst.edu.*

*R. S. Laramée is a Lecturer (Assistant Professor) of the Department of Computer Science, Swansea University, SA2 8PP, Wales, UK. Email: R.S.Laramée@swansea.ac.uk.*

*E. Zhang is with the School of Electrical Engineering and Computer Science, Oregon State University, 2111 Kelley Engineering Center, Corvallis, OR 97331. Email: zhang@eecs.oregonstate.edu.*

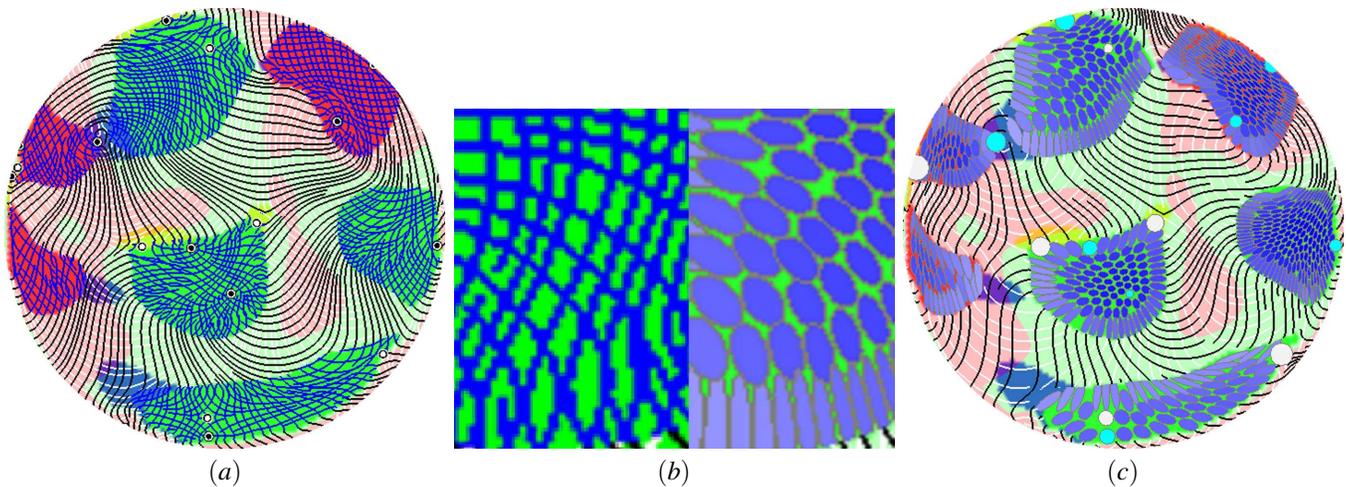


Fig. 1. Comparison of previous and current asymmetric tensor field visualization methods: (a) is based on hyperstreamlines only [1], (b) outlines the shortcomings of the previous method as compared to the hybrid method in the complex domains and (c) is the new hybrid visualization with varying streamline and glyph density in both the real and complex domains. This tensor field stems from the velocity gradient tensor of a flow field confined at a cross section from the cylinder portion of the diesel engine simulation. Regions colored in red indicate the flow inside possesses a counterclockwise rotational component, while green indicates a clockwise one. The blue and yellow areas denote contraction and dilation in the flow respectively. In (b), the flow patterns in the complex domains are difficult to discern using the previous method (a) but relatively simple using the glyphs (c). The semi-axes of the ellipses reflect singular values as well as dual eigenvectors of the asymmetric tensor and the density in both domains reflect the total tensor magnitude.

evenly-spaced hyperstreamlines that follow either the major or minor pseudo-eigenvectors, one obtains diamond-shaped regions whose smallest enclosing ellipses reflect the flow patterns.

Unfortunately, these techniques suffer from a number of problems when visualizing the velocity gradient tensor in the context of fluid mechanics. Firstly, local flow patterns of any given point inside the complex domain are elliptical, which cannot be directly inferred from dual-eigenvectors [1]. While evenly-spaced hyperstreamlines following pseudo-eigenvectors [1] can partially address this problem, it is difficult to achieve perfect evenness in the domain. This leads to difficulty in interpreting the orientations of the elliptical patterns. Furthermore, it is not possible to directly infer tensor magnitude, an important quantity in flow analysis, from pseudo-eigenvectors or evenly-spaced hyperstreamlines.

To more effectively illustrate the elliptical flow patterns inside the complex domains, we adapt the technique of *glyph packing* from symmetric tensors and apply it to the visualization of asymmetric tensor fields in the complex domains. This is possible because the dual-eigenvectors and their associated singular values [3] can be described as the eigenvectors and eigenvalues of a symmetric tensor [1]. Our basic strategy is to use hyperstreamlines in the real domains and elliptical glyphs in the complex domains. See Figure 1 for a comparison between the method based on using hyperstreamlines everywhere and using glyph packing inside complex domains. This provides a number of benefits over the previous method [1]. First, our method allows for direct visualization of the elliptical

patterns inside the complex domains. Second, strength of tensor field (tensor magnitude), an essential element in tensor understanding, can be directly visualized using the size of the glyphs and the density of the hyperstreamlines in conjunction with other methods such as color coding. Third, by including degenerate points into the glyph packing process, we allow degenerate points with strong rotational strengths to be easily distinguished from ones with weak strengths. Fourth, we demonstrate glyph packing on surfaces as well as in bounded regions. Notice that it is much easier to infer the elliptical flow patterns inside complex domains using elliptical glyphs than diamond formed by hyperstreamlines following pseudo-eigenvectors (Figure 1 (b)). Moreover, by varying the size of glyphs and density of hyperstreamlines, it is easier to differentiate larger features (Figure 1 (c): white and cyan glyphs which indicate degenerate points) from small features.

In this paper, we make the following contributions. First, we provide a novel visualization technique for asymmetric tensor fields motivated by physics. Instead of solely relying on either space-filling glyphs or hyperstreamlines, we make use of both to reflect the underlying physical properties of the tensor fields. To the best of our knowledge, this is the first technique that uses glyphs and hyperstreamlines in different regions of the same tensor visualization. This is also the first time that glyph packing is used in conjunction with asymmetric tensor fields. Note that this hybrid visualization is different from the approach in which glyphs are placed along a set of hyperstreamlines [4]. Second, an improved hyperstreamline placement algorithm is introduced to take into account the tensor magnitude in real domains. Third, packing ellipses

on a 3D surface is a challenging problem. We provide a solution based on the notions of geodesic distance computation and local parameterizations. Our hybrid visualization allows engineers to discriminate typical gyre (energetic) motion at a degenerate point from other degenerate points, such as those characterized by flow stagnation.

The rest of the paper is organized as follows: Section 2 reviews the related work of this paper. Section 3 provides the brief description of a number of basic concepts for asymmetric tensor field analysis. The pipeline of our hybrid visualization framework is presented in Section 4: Section 4.1 depicts the glyph packing technique we use for visualizing the tensor fields inside complex domains and Section 4.2 introduces the improved hyperstreamline placement process. In Section 5, we discuss the range of user controlled options for customizing the desired visualization. In Section 6, we present the visualization results of an engine simulation flow data set using our hybrid framework. Section 7 summarizes this work.

## 2 RELATED WORK

There has been much work in vector field visualization. We refer the readers to the following surveys for comprehensive reviews of these techniques [5], [6], [7]. Many of these techniques have been adapted to second-order symmetric tensor fields. In contrast, there has been relatively little work in the visualization of asymmetric tensor fields. Next, we review past work most relevant to our visualization approach.

### 2.1 Streamline- and Hyperstreamline-based Vector and Tensor Field Visualization

Turk and Banks [8] describe one of the first streamline placement methods for vector fields. In their method, evenly-spaced streamlines following the vector field are obtained through an iterative process. A new streamline is placed where there is a relatively large region free of previously-placed streamlines. This is detected using an image-based metric. Mao et al. [9] extend this technique to streamline placement for parameterized surfaces.

Jobard and Lefer [10] improve the generation of evenly-spaced streamlines with a rather different streamline seeding strategy. Instead of using an image-based metric, they place additional streamlines along existing streamlines with a user specified separating distance. This allows more direct control over the spacing of the streamlines. Spencer et al. [11] extend this to flow visualization on general surfaces.

A number of techniques seek to improve the quality of the streamline placement by improving the various stages in the streamline generation pipeline [12], [13], [14].

Chen et al. [15] incorporate topological features such as separatrices and periodic orbits into the streamline generation process, which enables topology-preservation visualization.

The idea of using evenly-spaced streamlines to visualize vector fields has been extended to tensor fields [16], [17], [18],

[19] and more generally  $N$ -way rotational symmetry ( $N$ -RoSy) fields [20], [21]. Note much of the aforementioned work can handle directional fields defined on curved surfaces.

Zhang et al. [22] propose to use streamtubes and streamsurfaces to visualize diffusion tensor MR images. See McLoughlin et al. [7] for an overview of seeding strategies.

### 2.2 Glyphs and Glyph Packing

Laidlaw et al. [23] stochastically place glyphs to minimize overlap for generating multi-layered diffusion tensor visualization. A similar glyph placement technique is introduced in the work of Kirby et al. [24] in which glyphs represent certain vector and tensor attributes of complex flow fields. The tensor splat method is proposed to convert tensor values into tuned Gabor functions which are encoded into 2D and 3D textures [25], [26]. Reaction-diffusion equations have been adapted by Kindlmann for tensor visualization [27] which is extended to the work on glyph packing [28]. In this study, a tensor-based potential energy is defined to derive the placement of a particle system whose final position will be used to place a glyph. In addition, glyph packing has been applied to create decorative mosaics by Hausner [29].

Note some past work has placed glyphs along hyperstreamlines such as the work by Hlawitschka and Scheuermann on higher-order tensor field analysis [4]. While such work also uses both primitives, it is fundamentally different from our work since they are not placed in complementary regions as in our case. It is not clear how to extend their work to visualize asymmetric tensor fields in a straightforward fashion. To our knowledge, our algorithm is the first to apply glyph packing to asymmetric tensor fields.

## 3 BACKGROUND

In this section we review relative background on asymmetric tensor fields. A *second-order* tensor  $\mathbf{T}$  can be represented by an  $N \times N$  matrix  $T_{ij}$  where  $N$  is the dimension of the field.  $\mathbf{T}$  is *symmetric* when  $T_{ij} = T_{ji}$  for any  $i \neq j$  or *anti-symmetric* when  $T_{ij} = -T_{ji}$  for any  $i \neq j$ . The *trace* of  $\mathbf{T}$  is defined as  $\sum_{1 \leq i \leq N} T_{ii}$ .  $\mathbf{T}$  is *traceless* when the trace of  $\mathbf{T}$  is zero. Note that any anti-symmetric tensor  $\mathbf{T}$  is traceless as  $T_{ii} = 0$  for any  $i$ . A special second-order tensor  $\mathbf{I}$ , the *Kronecker delta*, corresponds to the identity matrix, i.e.,  $I_{ii} = 1$  for any  $i$  and  $I_{ij} = 0$  for any  $i \neq j$ .

Any second-order  $\mathbf{T}$  can be *uniquely* decomposed as follows:

$$\mathbf{T} = \mathbf{D} + \mathbf{S} + \mathbf{R} \quad (1)$$

where  $\mathbf{D}$  is a multiple of the identity matrix,  $\mathbf{S}$  is a symmetric and traceless matrix, and  $\mathbf{R}$  is an anti-symmetric matrix. When  $\mathbf{T}$  is the velocity gradient tensor in fluid mechanics,  $\mathbf{D}$ ,  $\mathbf{S}$ , and  $\mathbf{R}$  represent the amount of volume change, angular deformation, and rotation, respectively [1].

In this paper, we will focus on two-dimensional asymmetric tensors, i.e.,  $N = 2$ . In this case, Equation 1 can be rewritten as

$$\mathbf{T} = \gamma_d I + \gamma_s \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix} + \gamma_r \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (2)$$

where  $\gamma_d$ ,  $\gamma_s$ , and  $\gamma_r$  are the strengths of volume change, angular deformation, and rotation, respectively.  $\theta$  encodes the directions of angular deformation.

For two-dimensional *incompressible* fluids,  $\gamma_d = 0$ , which corresponds to the set of traceless tensors. Tensors in this set can be parameterized as follows:

$$T(\rho, \theta, \varphi) = \rho \cos \varphi \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix} + \rho \sin \varphi \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (3)$$

Notice that the above form is a special case of Equation 2 in which  $\gamma_d = 0$ ,  $\rho = \sqrt{\gamma_s^2 + \gamma_r^2}$  and  $\varphi = \tan^{-1}(\frac{\gamma_r}{\gamma_s}) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ . The eigenvalues of  $T(\rho, \theta, \varphi)$  are:

$$E_{1,2} = \begin{cases} \pm \rho \sqrt{\cos 2\varphi} & \text{if } 0 \leq |\varphi| \leq \frac{\pi}{4} \\ \pm \rho \sqrt{-\cos 2\varphi} \mathbf{i} & \text{if } \frac{\pi}{4} < |\varphi| \leq \frac{\pi}{2} \end{cases} \quad (4)$$

A *tensor field*  $\mathbf{T}(\mathbf{p})$  is a tensor-valued function defined on an  $N$ -dimensional manifold  $D$  ( $N = 2$  in our case). We consider the map  $\tau : D \rightarrow S^2$  defined by:

$$\tau : \mathbf{p} \mapsto (\theta_{\mathbf{p}}, \varphi_{\mathbf{p}}) \quad (5)$$

where  $\theta_{\mathbf{p}}$  and  $\varphi_{\mathbf{p}}$  are the parameters corresponding to  $T(\mathbf{p})$ . The  $S^2$  is so-called *eigenvector manifold* [1] for which the North and South Poles ( $\phi = \pi/2$  and  $-\pi/2$ , respectively) correspond to the pure rotations. The circles  $\varphi = \pm \frac{\pi}{4}$  represent tensors with equal real eigenvalues, and they form the boundaries of tensors with real eigenvalues and with complex eigenvalues. The pre-image of  $\tau$  of these tensors are referred to as the *degenerate curves* [3], which divide the domain into *real domains* ( $\tau^{-1}(\{(\theta, \varphi) : 0 \leq |\varphi| < \frac{\pi}{4}\})$ ) and *complex domains* ( $\tau^{-1}(\{(\theta, \varphi) : \frac{\pi}{4} < |\varphi| \leq \frac{\pi}{2}\})$ ).

In the context of fluid, i.e., when the tensor is the gradient of the velocity vector field, the following interpretation is applicable [1]. First, in the real domains, a local linearized flow pattern at a point resembles a distorted hyperbola. The (real) eigenvectors indicate the direction of stretching and compression of fluid parcels. In the complex domains, local linearized flow patterns are elliptical whose eccentricity is given by:

$$e = \sqrt{\frac{2 \sin(2|\phi|)}{1 + 2 \sin(|\phi|)}} \quad (6)$$

for  $\frac{\pi}{4} < |\phi| \leq \frac{\pi}{2}$ . The major and minor axes of the ellipses are given by the *dual-eigenvectors* of the tensor, which are

the major and minor eigenvectors of the following symmetric matrix:

$$P_T = \frac{\gamma_r}{|\gamma_r|} \gamma_s \begin{pmatrix} \cos(\theta + \frac{\pi}{2}) & \sin(\theta + \frac{\pi}{2}) \\ \sin(\theta + \frac{\pi}{2}) & -\cos(\theta + \frac{\pi}{2}) \end{pmatrix} \quad (7)$$

Notice the dual-eigenvectors are not well-defined at the *degenerate points* that are the pre-image of the Poles in the eigenvector manifold.

Zhang et al. [1] point out the need to visualize the local flow patterns for effective flow visualizations. While it is straightforward to visualize such patterns in the real domains using hyperstreamlines following the major and minor eigenvector fields, using hyperstreamlines in the complex domains becomes less effective (Figure 1).

Zheng and Pang have used hyperstreamlines following the dual-eigenvectors in the complex domain [3]. However, it cannot convey the eccentricity of elliptical patterns. To overcome this difficulty, Zhang et al. [1] define *pseudo-eigenvectors* which they use to *infer* the elliptical patterns (Figure 1 (a): blue curves). However, due to challenges with placing evenly-spaced hyperstreamlines, it is difficult to infer the elliptical patterns in this fashion. More importantly, the strength of the tensor is not reflected in the visualization.

Our goal is to provide a more natural way of illustrating the local flow patterns inside the complex domains which can seamlessly integrate with efficient hyperstreamline-based visualizations in the real domain. Moreover, we wish to incorporate tensor magnitude into the visualization.

## 4 HYBRID VISUALIZATION PIPELINE

Our hybrid visualization pipeline can be described as follows. First, the real domains and complex domains are extracted and the background coloring scheme applied [1] (Figure 2 (a)). Second, glyph packing is conducted inside the complex domains with degenerate points included in the packing process (Section 4.1) (Figure 2 (b)). Special care is given to the treatment of the boundaries between these regions. Third, the hyperstreamlines are placed inside the real domains (Figure 2 (c)). Finally, we combine the results for real domains and complex domains to obtain the complete visualization and highlight the degenerate points (Figure 2 (d)).

### 4.1 Glyph Packing In Complex Domains

Our glyph packing technique follows the work by Kindlmann et al. [28] except that we incorporate degenerate points into the packing process, take into account the boundaries of the disjoint complex regions, and allow packing on manifold surfaces. We will first review their framework. Kindlmann et al. describe controlling a particle system whose density varies over space according to the tensor determinants. For visualization, each particle is assigned an ellipse whose axes coincide with the eigenvectors of the tensor defined at that

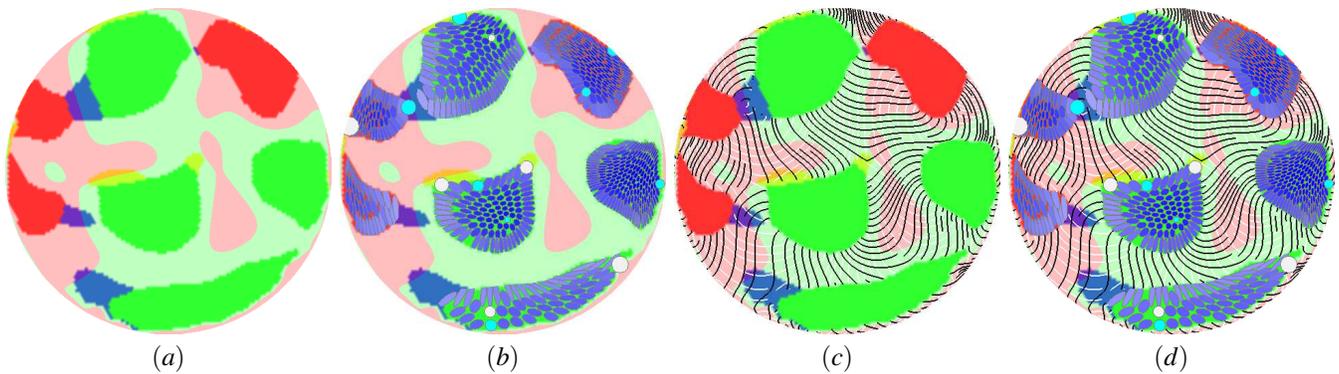


Fig. 2. This figure illustrates the basic constituents of our hybrid visualization of an asymmetric tensor field. (a) the real domains (light yellow = positive scaling + clockwise rotation, orange = negative scaling + counterclockwise rotation, blue = negative scaling + clockwise rotation, purple = negative scaling + counterclockwise rotation, light green = stretching + clockwise rotation, and pink = stretching + counterclockwise rotation) and complex domains (red=counterclockwise, green=clockwise) are identified by the eigenvector manifold, respectively; (b) the glyph packing is conducted inside the complex domains; (c) the hyperstreamlines are computed inside the real domains; (d) finally, the results of (b) and (c) are combined to produce the hybrid visualization.

point and area is determined by the eigenvalues. The basic steps of their algorithm can be described as follows. First, a number of particles (seeds) are initialized in the domain based on a stochastic strategy determined by the tensor determinants. Second, an iterative process is conducted to move the particles in proper locations to minimize the potential energy [28] of each particle. This is realized by repelling the particles according to the forces exerted by their nearby particles. These forces are derived from the potential energy between pairs of particles. To accelerate this, only particles within a small neighborhood are considered when accumulating forces for a particle. This size of the neighborhood is determined by a user specified glyph size parameter  $\alpha$ , an adjustable parameter  $\gamma$ , and the maximum eigenvalue in the domain [28]. These parameters provide control over the shape, size, and spacing of the glyphs. Refer to Kindlmann's work for more detail on calculating these forces.

Different from Kindlmann et al.'s work [28], our visualization technique can accept both tensor and vector fields as input. It is noted that the tensor fields in the latter case stem from the gradient computation of the input vector fields. These derived tensor fields are asymmetric. The field is defined at the vertices of a triangular mesh which serves as a discretization of domain. The field values on the edges and inside the triangles of the mesh are computed using a piecewise linear interpolation scheme in the plane. On curved surfaces, the interpolation schemes proposed by Zhang et al. for tensor fields [19] and vector fields [30] are employed to ensure the continuity of field values across edges and vertices of the mesh. In particular, given a point  $p$  inside a triangle  $K = (v_1, v_2, v_3)$  with barycentric coordinates  $(w_1, w_2, w_3)$ , let  $T(v_i) = D(v_i) + S(v_i) + R(v_i)$  be the isotropic scaling, anisotropic stretching, and rotation at  $v_i$ , respectively. Then  $T(p) = D(p) + S(p) + R(p)$  where  $S(p)$  follows the non-linear interpolation scheme of [3] while  $D(p) = w_1D(v_1) + w_2D(v_2) + w_3D(v_3)$  and  $R(p) = w_1R(v_1) + w_2R(v_2) + w_3R(v_3)$ .

In order to apply glyph packing which requires symmetric tensor fields as input, we compute the following symmetric tensor:

$$T = (u_1 \ u_2) \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix} (u_1 \ u_2)^T \quad (8)$$

where  $u_1$  and  $u_2$  are the two eigenvectors of the symmetric matrix given by equation 7, and

$$\begin{cases} J_1 = \max\{|\gamma_s + \gamma_r|, |\gamma_s - \gamma_r|\} \\ J_2 = \min\{|\gamma_s + \gamma_r|, |\gamma_s - \gamma_r|\} \end{cases}$$

Note  $J_1$  and  $J_2$  computed this way are the singular values for the asymmetric tensor by subtracting the trace. This treatment has the following physical meanings. For 2D incompressible fluids, the local linearization at any point inside the complex domain is an elliptic pattern whose eccentricity and semi-axes are defined by equation 8. In this case, eccentricity indicates the relative strength between stretching and rotation. The smaller the eccentricity, the stronger the rotation. When eccentricity is minimum ( $= 0$ ), there is no stretching but only rotation. That is where degenerate points occur (Poles in the eigenvector manifold). When the eccentricity equals 1, we are on the degenerate curves, i.e., stretching equals rotation. The directions of this ellipse are the dual- eigenvectors of the tensor [1]. The singular values of the matrix (the long and short semi-axes) represent the overall tensor strength. Note that for 2D incompressible flows there is no volume change, i.e., no isotropic scaling.

In addition, our packing process is carried out inside bounded regions. For better visualization, no glyphs are placed on or near boundaries. Due to the unstructured mesh and the aforementioned additional requirements, some of the techniques in [28] need to be augmented such as the neighborhood calculation for accumulating forces and seeding strategy. Boundary handling is also included in the glyph packing pipeline. The neighborhood calculation on a mesh can be implemented using a fast marching technique [31]. In this section, we will describe our seeding strategy and boundary handling.

### 4.1.1 Seeding Strategy

The criterion of seeding for the purpose of glyph packing is to determine an approximate number of seeds in the region based on the sum of the areas of the glyphs to be packed and the total area of the region. Ideally, the area of the region should be slightly larger than the total area of the glyphs. When these two values are equal, glyph overlap can be expected which can be aesthetically displeasing. In the visualization, we expect to see minimal overlap as well as empty space between glyphs. The seeding must also incorporate the dynamic area of glyphs as they move into their final positions. In practice, we introduce a user modified density parameter  $d$  to help control the overall number of glyphs seeded in the region.

Another issue that we have to consider is to how to incorporate degenerate points, which are important features in fluid mechanics [1]. A degenerate point in the velocity gradient tensor field represents the location of zero angular strain and coincides with the location of pure circular rotation of fluid. To better visualize the tensor behaviors near the degenerate points of an asymmetric tensor field, we assign a fixed seed point at the center of each degenerate point. Figure 1 compares the glyph packing results using the previous method of placing a filled circle to denote degenerate points (a) to the fixed glyphs colored in white and cyan (c). Note how the sizes of the degenerate points visually convey the strength of the rotating flow in this area.

Our seeding algorithm can be described as follows. First, we compute the total area  $A$  of a complex region by accumulating the areas of the triangles inside the region. Secondly, we seed degenerate points in the region and initialize the total area  $A_s$  of the seeded points with the sum of their respective areas. Finally, we iteratively seed points until the total area  $A_s$  is larger than  $d \times A$  where  $d$  controls the density of glyphs in the region being seeded.

For a given seed, we randomly choose a triangle within the region and compute a probability

$$Pr = (1 - |det|/|det_{max}|) \times tr \quad (9)$$

where  $tr$  denotes ratio between the area of the triangle and glyph being considered ( $A_{tri}/A_{glyph}$ ),  $det_{max}$  is the maximum determinant of the tensor over the region, and  $det$  represents the determinant of tensor value at current seeding position. For regions with large differences in the size of the triangles, glyphs are more likely to be seeded in smaller triangles as there are potentially more of them. Also since a seed near a degenerate point has a large determinant,  $1 - |det|/|det_{max}|$  is small. This prevents seeding many points near the degenerate points. With the probability  $Pr$  we accept a seed and update the total area of the seeds. Note that this algorithm is conducted inside each complex region individually. Also, the area of each seed point is evaluated as the area of the ellipse with the seed point as the center (i.e.  $\alpha^2 \pi J_1 J_2$  where  $J_1$  and  $J_2$  are the two eigenvalues of the symmetric tensor  $T$  from Equation 8 at the seed).

---

### Seeding algorithm

---

**Input:**  $T$ : the triangle list of the region

$DL$ : the list of degenerate points in the region

**Output:**  $S$ : the seed point list

**Local variables:**  $det_{max}$ : the maximum determinant

$det$ : current determinant

$A$ : total area of the domain

$A_s$ : the current seed area

$Pr$ : the probability of accepting a seed

$tr$ : the ratio of triangle and glyph area

$q$ : a location in the region

$r$ : a random number between 0 and 1

$d$ : percentage of total area to be seeded

**Begin**

$A = \text{area}(T)$ ;  $A_s = 0$ ;

$det_{max} = \text{get\_max\_determinant}(T)$ ;

$S \leftarrow \text{add\_to\_seeds}(DL)$ ;

$A_s = \text{compute\_seed\_area}(DL)$ ;

While  $A_s < d \times A$  Do

$q \leftarrow \text{random\_triangle\_center}(T)$ ;

$det = \text{compute\_determinant}(q)$ ;

$tr = \text{compute\_tri\_area} / \text{compute\_seed\_area}(q)$ ;

$Pr = (1 - |det|/|det_{max}|) \times tr$ ;

$r = \text{random\_number\_generator}()$ ;

If  $r < Pr$

$S \leftarrow \text{add\_to\_seeds}(q)$ ;

$A_s = A_s + \text{compute\_seed\_area}(q)$ ;

EndWhile

**End**

---

In our original implementation, seeds at degenerate points are fixed during the repelling stage. This potentially causes problems during the movement of other seeds. More specifically, these fixed seeds can block other seeds from reaching optimal locations, which can lead to a number of holes in the result. To handle this, we modify the seeding scheme at degenerate points as follows. Initially, we place seeds at degenerate points with zero area (i.e. having zero influence on the other seeds). After a certain number of repulsion iterations, we linearly increase the size of the glyphs located at the degenerate points until they achieve their true sizes. The repelling process continues as normal at this point until the user is satisfied with the resulting packing. This modification eases the other glyphs into their proper locations while allowing the degenerate points to be fixed in place. The seeding algorithm can be described in the **seeding algorithm** 4.1.1.

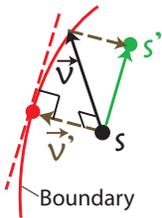
### 4.1.2 Boundary Handling

To prevent the hyperstreamlines from entering the complex domains and glyphs from overlapping the real domains, we need to handle the boundaries between different domains. For glyph packing we start placing scaffolding points on these boundaries to repel other seeds inside the complex domains and prevent them from crossing the boundary. The scaffolding points are fixed in their locations during the repelling process and not visualized in the output. These points allow the the

glyphs inside the boundaries to be pushed along the boundaries if necessary without allowing them to roam across degenerate curves into real domains.

We observe that the scaffolding points were not entirely effective in keeping the glyphs inside a complex domain. In certain cases, the repulsion forces from the actual seeds overwhelmed the scaffolding points allowing seeds to cross the boundaries. Instead of increasing the number of scaffolding points, which could adversely affect the visualization, we made the following adjustment. Consider a seed  $s$ , let its current direction of motion to be  $\vec{v}$ . When the next location of  $s$  crosses outside of the complex region, we compute the point on the boundary  $s$  is crossing which has the shortest distance to  $s$ , denoted by  $\vec{v}'$ .

To further help push this point from the boundary, we subtract  $\vec{v}'$  from  $\vec{v}$  to compute the new position of  $s$  near the boundary. If this new position is still outside the boundary,  $s$  remains fixed in its original position. The figure to the right illustrates such an idea (green arrow is the updated direction).



## 4.2 Hyperstreamline Placement With Spatially-Varying Density In Real Domains

We adapt Jobard and Lefer [10]'s algorithm for evenly-spaced streamline placement with sign ambiguity being considered. This will typically generate evenly placed hyperstreamlines. In order to visualize the tensor magnitude with hyperstreamlines, we make use of it to control the hyperstreamline density. More specifically, given a current integration point  $P$ , we determine whether it is close to an existing sample point  $S$  as follows. Let  $T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}$  be the average tensor between  $P$  and  $S$  (e.g. the tensor at the middle point of the line segment between  $P$  and  $S$ ). The tensor magnitude of  $T$  is

$$L = \sqrt{\gamma_s^2 + \gamma_r^2}$$

where  $\gamma_s$  and  $\gamma_r$  are defined in Section 3. We determine a scaling factor  $k$  which allows the program to determine when seeds are too close together. Then,  $P$  is said to be too close to  $S$  if its distance to  $S$  is smaller than  $kd_{sep}$  where  $d_{sep}$  is the globally-defined unscaled separation distance between hyperstreamlines as defined in [10]. In that case, we reject  $P$  and stop the tracing of current hyperstreamline. A similar process is conducted for rejecting seeds. Figure 1 (c) provides the visualization result of such a hyperstreamline placement using a spatially-varying density. Notice that high tensor magnitude is reflected by high densities. Note that similar idea has also been applied in the street placement by Chen et al. [32] while a density map is input to control the density of the streets during the tracing of hyperstreamlines.

## 4.3 Visualization on Surfaces

Almost all of the techniques we introduced for visualizing asymmetric tensor fields defined in a plane also ap-

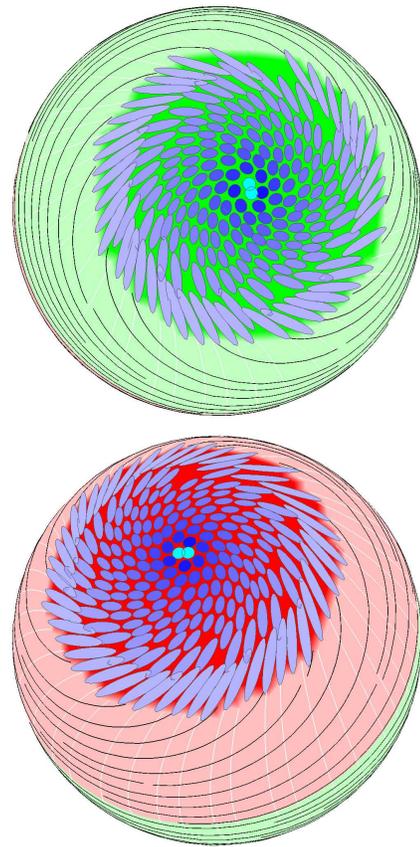
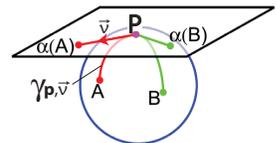


Fig. 3. An example asymmetric tensor field on the unit sphere is illustrated using our hybrid approach. The major and minor eigenvectors of the symmetric component of this field follow the latitude and longitude, respectively, everywhere on the surface except the Poles. The tensor magnitude is the weakest at the degenerate curves and strongest at the Equator and Poles. Notice that the tensor magnitude is mapped to the density of hyperstreamlines (higher tensor magnitude = higher density) and size of glyphs (higher tensor magnitude = smaller glyphs).

ply to surfaces with one exception. One primary concern of visualization on surfaces is the ability to efficiently calculate the distance between points for repelling glyph seeds and the rejecting of streamline seeds. To address this challenge, we compute a *geodesic polar map* [30] for the curved surface and then approximate these quantities based on this map. A geodesic polar map is the generalized polar coordinate system (in a plane) on surfaces. The geodesic polar map at a point  $\mathbf{p}$  on a curved surface is computed based on the geodesics emanating from  $\mathbf{p}$ . A geodesic on a curved surface is a locally shortest and straightest curve. Starting from  $\mathbf{p}$ , there is a geodesic in every tangent direction  $\vec{v}$ . Denote this geodesic by  $\gamma_{\mathbf{p},\vec{v}}$ . A point  $\mathbf{q}$  on  $\gamma_{\mathbf{p},\vec{v}}$  with a distance  $\rho$  from  $\mathbf{p}$  can be identified by the coordinates  $(\rho, \theta)$  where  $\theta$  is the angular coordinate of  $\vec{v}$  with respect



to surfaces with one exception. One primary concern of visualization on surfaces is the ability to efficiently calculate the distance between points for repelling glyph seeds and the rejecting of streamline seeds. To address this challenge, we compute a *geodesic polar map* [30] for the curved surface and then approximate these quantities based on this map. A geodesic polar map is the generalized polar coordinate system (in a plane) on surfaces. The geodesic polar map at a point  $\mathbf{p}$  on a curved surface is computed based on the geodesics emanating from  $\mathbf{p}$ . A geodesic on a curved surface is a locally shortest and straightest curve. Starting from  $\mathbf{p}$ , there is a geodesic in every tangent direction  $\vec{v}$ . Denote this geodesic by  $\gamma_{\mathbf{p},\vec{v}}$ . A point  $\mathbf{q}$  on  $\gamma_{\mathbf{p},\vec{v}}$  with a distance  $\rho$  from  $\mathbf{p}$  can be identified by the coordinates  $(\rho, \theta)$  where  $\theta$  is the angular coordinate of  $\vec{v}$  with respect

to some local frame at  $\mathbf{p}$  [30]. See figure in this section for an illustration. In our implementation of glyph rendering, an individual glyph is drawn in the tangent plane. Drawing glyphs directly on the surface is an area of future work.

The example shown in Figure 3 is based on the following asymmetric tensor field. Given a point on the unit sphere, we project the vector  $(0, 1, 0)$  into the local frame at this point and rotate 90deg. From this projection, we calculate the angular component of the vector  $\theta$ . We then calculate  $\phi$  as the angle between the x-z plane and the vector created by the point on the sphere and the origin. Then the field  $T$  is defined as:

$$T = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix} + \tan \phi \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (10)$$

The tensor field  $T$  is purely anisotropic stretching on the equator  $\phi = 0$  and purely rotational at the poles  $\phi = \pm \frac{\pi}{2}$ . The degenerate curves occur at  $\phi = \pm \frac{\pi}{4}$ , where the tensor magnitude  $\sqrt{1 + \tan^2 \phi}$  achieves its minimum. Figure 3 is another example of our method applied to surfaces.

## 5 VISUALIZATION OPTIONS

In addition to the basic algorithm behind this hybrid visualization technique, we propose a set of visualization options that allow the user to control the quality of visualization. The general pipeline stays the same but different problem domains provide different challenges. These visualization options allow users to customize the resulting images to a specific problem domain. The paper discusses the applicable options for fluid flow but does not rule out other options for more general applications of asymmetric tensor visualization.

### 5.1 Glyph Scaling

We control the glyph scaling using two different methods. Firstly, the  $\alpha$  parameter can be used to change the size of the glyphs globally. As alpha increases, each glyph increases in size and area and the overall number of glyphs seeded decreases. The purpose of this parameter is to allow the user to scale the glyphs as needed for different levels of zoom. As the visualization is zoomed in, a smaller glyph could help better identify features of the complex region. See Figure 5 for an example of the use of the alpha parameter.

In our experiments, we noticed another need for glyph scaling. In some applications, the determinants in different complex domains of an asymmetric tensor field stemming from the flow field vary greatly. So do their eigenvalues,  $J_1$  and  $J_2$ , which typically leads to large variations in the sizes of the glyphs. Figure 4 (a) provides such an example where some glyphs cover most of the complex domains (e.g. glyphs in the right-most region). In addition, large eigenvalues require a large neighborhood computation when accumulating forces exerted on a seed. This increases the computation expense. To overcome this, we provide two options. The choice of these methods is more suited for the individual application. Both

methods involve using the eigenvalues,  $J_1$  and  $J_2$ , of the tensor to control the scaling.

The first method introduces thresholds to the eigenvalues of the tensor. This scaling method results in visually pleasing images however, information is lost as every eigenvalue above the threshold produces the same size glyph. This may be acceptable for certain applications. To overcome this, we first normalize the glyph sizes and then multiply them by a scaling factor based on the eigenvalues of the tensor. Refer to Figure 4 (b) for an example of this method.

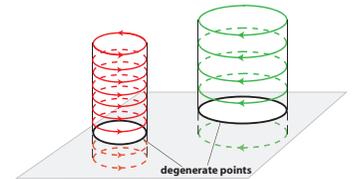
The second method involves scaling the glyphs based on some properties. After experimenting with a few different ways of realizing this, we adopted the log of the area of the glyph which amounted to the log of the product of the eigenvalues using the following process where  $J_1$  and  $J_2$  are the existing eigenvalues of the tensor and  $N_1$  and  $N_2$  are the new eigenvalues. We want to have the new area  $A = N_1 N_2 = \log(J_1 J_2 + 1)$  while also having  $N_1 / N_2 = J_1 / J_2 = \text{ratio}$ . We can solve the second equation for  $N_1$ , substitute that into the first equation and solve for  $N_2$ . This becomes:

$$\begin{aligned} N_2 &= \sqrt{\log(J_1 J_2 + 1) / \text{ratio}} \\ N_1 &= \text{ratio} \times N_2 \end{aligned}$$

We took this quantity and used it to preserve the eccentricity of the glyph (the ratio between the eigenvalues). Overall, the glyphs were scaled down to a visible level while the respect sizes and eccentricity are maintained (See Figure 4 (c)).

### 5.2 Glyph and Streamline Density

The notion of glyph and streamline density can be different amongst different applications. For instance when visualizing flow fields using the velocity gradient field, large tensor magnitudes relate to high stretching and rotational components of the tensor. In the flow, this behavior creates a vortex shape which becomes more concentrated as the rotational and stretching component becomes larger. This means that larger tensor magnitudes actually relate more effectively to larger glyph and streamline densities and smaller glyph sizes in the context of flow visualization. See the illustrated figure for an example of this.



However, this is contrary to the notion that large values relate better to large visualization structures. To an untrained eye, the smaller objects in the image would naturally correlate to small values and conversely, large objects relate to large values.

To get an idea of which method would be suitable, a user study would have to be conducted to determine the effectiveness of both styles of representing this information. Because this method is not restricted to only flow visualization, we leave

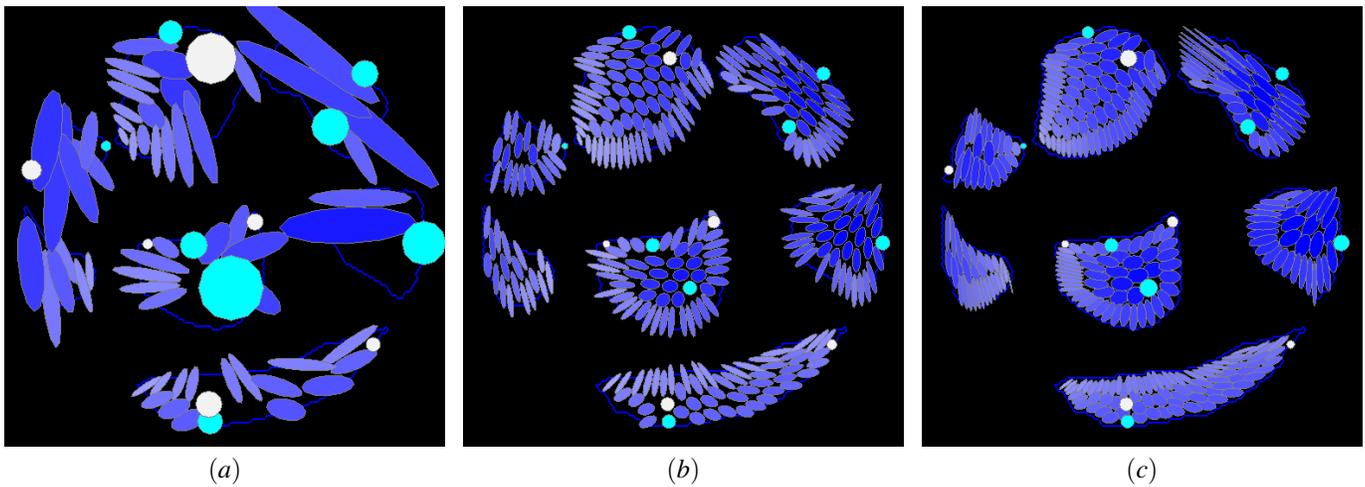


Fig. 4. This figure shows the results of glyph packing without scaling the glyphs (a), with a finite range for the tensor magnitude (b) and using the log scaling of the area of the glyph (c). In (a), the difference in size of the glyphs causes problems with the glyph seeding, the ability of glyphs to be drawn in the complex regions, as well as the repulsion process during glyph packing.

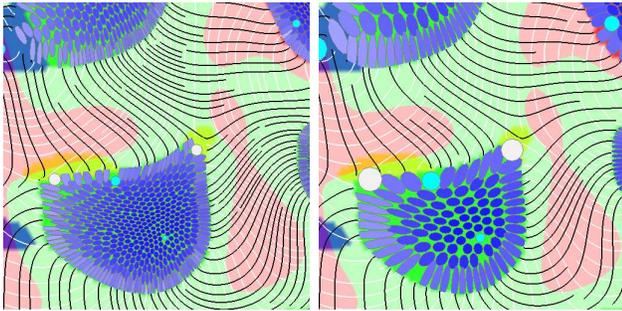


Fig. 5. This figure shows the effect of the  $\alpha$  parameter on the overall size of the glyphs. The images on the left and right use alpha values of 0.01 and 0.02 respectively. The seeding method adaptively changes the number of initial glyphs based on the  $\alpha$  value.

the choice of how to visualize the density of streamlines and glyphs to the user. This allows the user to decide which visualization technique is more appropriate for the application as opposed to settling on a method that simply applies to the majority of applications.

## 6 APPLICATIONS - ENGINE SIMULATION DATA

We have applied our hybrid visualization technique to a diesel engine simulation data. An ideal flow pattern strived for in a diesel engine [33] resembles a helix spiral about an imaginary axis aligned with the combustion chamber as illustrated in Figure 7. Achieving this ideal motion results in an optimal mixing of air and fuel and thus a more efficient combustion process. The flow is evidently highly three-dimensional and transient. A number of vector field visualization techniques have been applied to a simulated flow inside the diesel engine [1], [33], [34], [35]. It is traditional and standard in practice that two-dimensional slices through a 3D domain are analyzed

by fluid and simulation engineers [33]. This practice offers the advantage of reduced dimensionality. In other words, visualization of planar data does not suffer from occlusion and is generally characterized by lower visual complexity than that of 3D. Practitioners often rely on dimension reducing techniques precisely to avoid these challenges and simplify analysis and presentation of high-dimensional data. Slices also allow the engineer to focus on the most important subsets of the flow. Furthermore, general time-dependent visualization often relies on animation to depict transient behavior. This can be cognitively challenging for the viewer due to the short lifespan of interesting flow features. Therefore, practitioners need static imagery in order to analyze and present visualization results regardless of the temporal dimensionality of the simulation data.

Figure 6 shows our hybrid visualization technique on the flows in the transverse slices at 230 mm (a), 530 mm (b), and 830 mm (c) from the top of the diesel engine cylinder. The density of the hyperstreamlines and sizes of the glyphs represent the tensor magnitude: the denser the hyperstreamlines and the smaller the glyphs, the larger the tensor magnitudes. This representation is aligned with that traditionally used for flow vector fields: i.e. the denser the streamlines and the vortex lines (or the smaller the stream-tubes), the larger the magnitudes. There are several salient features that can be observed in Figure 6. First, we recognize decrease in tensor magnitude from the upper plane (230 mm) to the lower plane (830 mm), which reflects the flow condition being at the end of the intake process, i.e., the piston head is near the bottom. The flow from the intake port expands to the cylinder chamber, creating intense velocity gradients via flow separation near the top. The velocity gradients tend to diminish near the bottom where the piston head decelerates at the end of the intake process and the vortex tubes expand; consequently, fluid rotation is reduced near the bottom. A careful observation in the 230 mm plane also reveals that the tensor magnitude is stronger in the

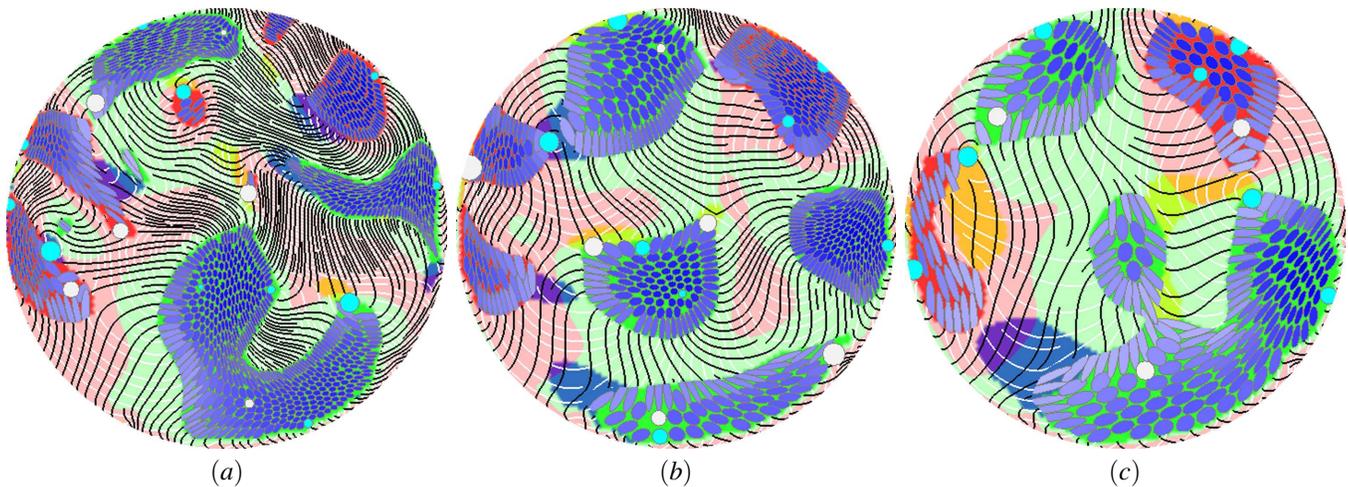


Fig. 6. This figure compares three planar slices of the diesel engine simulation data. (a) is 230 mm, (b) is 530 mm and (c) is 830 mm from the top of the diesel engine cylinder.

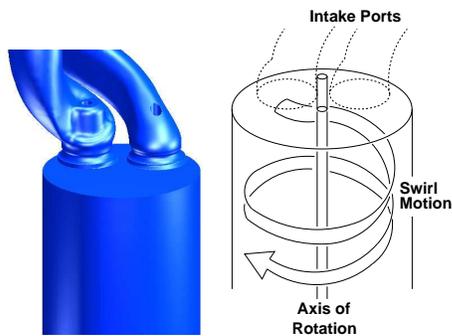


Fig. 7. The swirling motion of flow in the combustion chamber of a diesel engine. *Swirl* is used to describe circulation about the cylinder axis. The intake ports at the top provide the tangential component of the flow necessary for swirl. The data set consists of 776,000 unstructured, adaptive resolution grid cells.

first quadrant and weaker in the third quadrant, reflecting the location of the intake port that induces the clockwise spiral as illustrated in Figure 7. It is interesting to point out that, in spite of the induced spiral, the general pattern of where the clockwise (green) and counterclockwise (red) motions dominate remains similar in each plane. Hence, the spiral motion must be weaker than that as illustrated in Figure 7.

In each plot, the locations of irrotational motion of fluid elements can be identified explicitly at the interface of light green and light red regions. Stretching and rotation of fluid-elements can be detected explicitly by tracing the direction of hyperstreamlines and the shape of glyphs. The major and minor eigenvectors in the real domain merge tangentially at the boundary of the complex domains. The glyph presentation shows how the stretching of a fluid particle continues into the complex domain, forming the elongated swirl motions near the boundaries. The interface of the real and complex domains represents the transition from linear stretching motion (stretching with rotation) to curved swirl motion (rotation with

stretching) of a fluid element. A degenerate point represents the location of pure rotation; hence it can be interpreted where a vortex line intersects perpendicularly to the plane without stretching in the transverse plane. Also a degenerate point that is located in the isotropic-scaling dominated region (dilation (yellow) and compression (blue), respectively) implies the spiral flow along the vortex line from the third dimension.

Comparing the three planes shown in Figure 6, the transformations of glyph patches from one plane to the other provide substantial insights to fluid element motions. For example, the glyph packing in the complex domain near the center (clockwise rotation - green) shows the change in strength and elongation from the upper to lower planes: the dominant elongation in the up-down direction in the 230 mm plane becomes the side direction in the 530 mm plane, and then back to the up-down direction but weaker tensor strength in the 830 mm plane. Also observed in the center glyph packing are degenerate points well inside the complex domain that represent the locations of pure rotation. It is noted that the weak degenerate point near 10 O'clock in the 530 mm plane, that appears at the edge of complex domain, could represent flow separation or attachment. It is located between a pair of counter-rotating flow regions, in the close neighborhood of irrotational flow (at the interface of light red and light green) and within a scaling dominated region, in this case negative isotropic scaling (compression). Note that flow attachment/separation occurs at the transition of fluid rotations in a two-dimensional flow. There are many revealing analyses can be made from Figure 6. Those enlightening observations are now possible with our new hybrid visualization technique.

In addition to the plane-surface analysis, more information can be extracted from the visualization presented in Figure 8. This figure shows the present hybrid visualization (a) on the curved surface of the diesel engine - the same data set used for Figure 6 - together with a combination of the eigenvector visualization in the real domain and the pseudo eigenvector visualization in the complex domain (b), as well

as the visualization of vector field texture (c). Note that (b) was previously presented in [1]. Again, in Figure 8 (a), the density of hyperstreamlines and the size of glyphs represent the tensor magnitude: the denser the hyperstreamlines and the smaller the glyphs, the larger the tensor magnitudes. The hyperstreamline density and the glyph size in (a) demonstrate that the tensor magnitude decreases from the top to the bottom, consistent to the foregoing discussion for the transverse slices in Figure 5. No such magnitude information can be obtained from the other visualizations: (b) and (c). Fluid deformation patterns on the surface can be gained by tracing the major and minor hyperstreamlines and the elongation of glyphs in (a). For example, the hyperstreamlines in the upper-left area exhibit nearly pure stretching in the slightly slant vertical direction: the major and the minor are nearly perpendicular to each other. The major hyperstreamlines change the direction rather abruptly to the horizontal in the small region next to the complex domain (red = counterclockwise rotation), and the angle between the major and the minor becomes small, indicating that rotation is imposed. This motion in the real domain is smoothly transformed into the complex domain. While the elongated swirl-like motion of fluid elements (rotation with stretching) is evident in our glyph representation in (a), detection of such motion is implicit and formidable in the previous visualization in (b). The hyperstreamlines behave differently in the relatively large area right side of the complex domain, about one-quarter down from the top of the cylinder. The angle between the major and the minor hyperstreamlines remains small, and the fluid elements in the real domain stretch with rotation in the horizontal direction. This demonstrates that our hybrid visualization is capable of tracing such fluid-element patterns seamlessly in an integrated fashion.

The glyph size in the complex domain (red) is fairly uniform, except near the bottom where the glyph size is much larger (i.e. weak tensor magnitude). There appear four degenerating points: the two are of trisectors and the other two are of wedges. Considering that they are located at the edge of the complex domain, within the isotropic scaling dominated region (dilation (yellow)), and near the irrotational flow (at the interface of light red and light green), those weak degenerating points must represent flow stagnation. Such intriguing flow behaviors can be detected effectively with the present hybrid tensor field visualization (a), but not the previous visualizations (b) or (c). There is another degenerating point at the top right edge of the view close to the intake port. Note that there is no other degenerating point on the surface, which indicates angular deformation of fluid elements dominates on the cylinder surface everywhere, except near the bottom stagnation region.

## 7 CONCLUSION

Asymmetric tensor field visualization is an important topic in the visualization community for which more work is needed. In this paper, we highlight the challenges faced by state-of-art techniques for asymmetric tensor field visualization including the loss of magnitude information and the lack of

effectiveness for conveying the eccentricity information in the complex domains. In order to address these, we introduce a hybrid visualization technique for asymmetric tensor field visualization in which hyperstreamlines and glyphs are used to represent the tensor patterns in the real and complex domains, respectively. The sizes of the glyphs and density of the hyperstreamlines are used to convey tensor magnitude. In addition, degenerate points are maintained by the visualization as they are incorporated in the glyph packing stage. This is the first time glyph packing is used in conjunction with asymmetric tensor fields. The combination of these techniques generate the hybrid visualization results that are capable of delivering the underlying physical characteristics of the data more effectively and efficiently.

For future work, one major area of interest is the color scheme used for visualization. We are interested in adding level-of-detail display of glyphs and hyperstreamlines. Incorporating the volume change component ( $\gamma_d$ ) is also desirable. Notice the density of hyperstreamlines and glyphs can be mapped to other tensor-related quantities and colors can be used for this purpose as well. We plan to use this idea to experiment with other visualizations in which valuable information can be conveyed. We also want to explore some problems in the glyph packing domain. Firstly, we want to refine glyph packing in non-uniform bounded regions as in Figure 6 (c). Secondly, we also want to draw glyphs directly on curved surfaces by way of the polar maps discussed in this paper. Drawing glyphs in the tangent plane at a point causes problems when drawing glyph near each other as in Figure 8 (a) toward the bottom. Solving this problem would greatly improve the quality of the visualization but it turns out to be very challenging.

## REFERENCES

- [1] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramée, "Asymmetric Tensor Analysis for Flow Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 1, pp. 106–122, 2009.
- [2] X. Tricoche, G. Scheuermann, and H. Hagen, "Tensor Topology Tracking: A Visualization Method for Time-Dependent 2D Symmetric Tensor Fields," in *Computer Graphics Forum 20(3) (Eurographics 2001)*, Sep. 2001, pp. 461–470.
- [3] X. Zheng and A. Pang, "2D Asymmetric Tensor Analysis," *IEEE Proceedings on Visualization*, pp. 3–10, Oct 2005.
- [4] M. Hlawitschka and G. Scheuermann, "HOT Lines: Tracking Lines in Higher Order Tensor Fields," in *Proceedings IEEE Visualization 2005*, 2005, pp. 27–34.
- [5] R. S. Laramée, 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*, vol. 23, no. 2, pp. 203–221, June 2004.
- [6] R. S. Laramée, H. Hauser, L. Zhao, and F. H. Post, "Topology-Based Flow Visualization: The State of the Art," in *The Topology-Based Methods in Visualization Workshop (TopoInVis 2005)*, *Visualization and Mathematics*, 2007, pp. 1–19.
- [7] T. McLoughlin, R. S. Laramée, R. Peikert, F. H. Post, and M. Chen, "Over Two Decades of Integration-Based, Geometric Flow Visualization," in *Eurographics 2009, State of the Art Reports*, 2009.
- [8] G. Turk and D. Banks, "Image-Guided Streamline Placement," in *ACM SIGGRAPH 96 Conference Proceedings*, Aug. 1996, pp. 453–460.

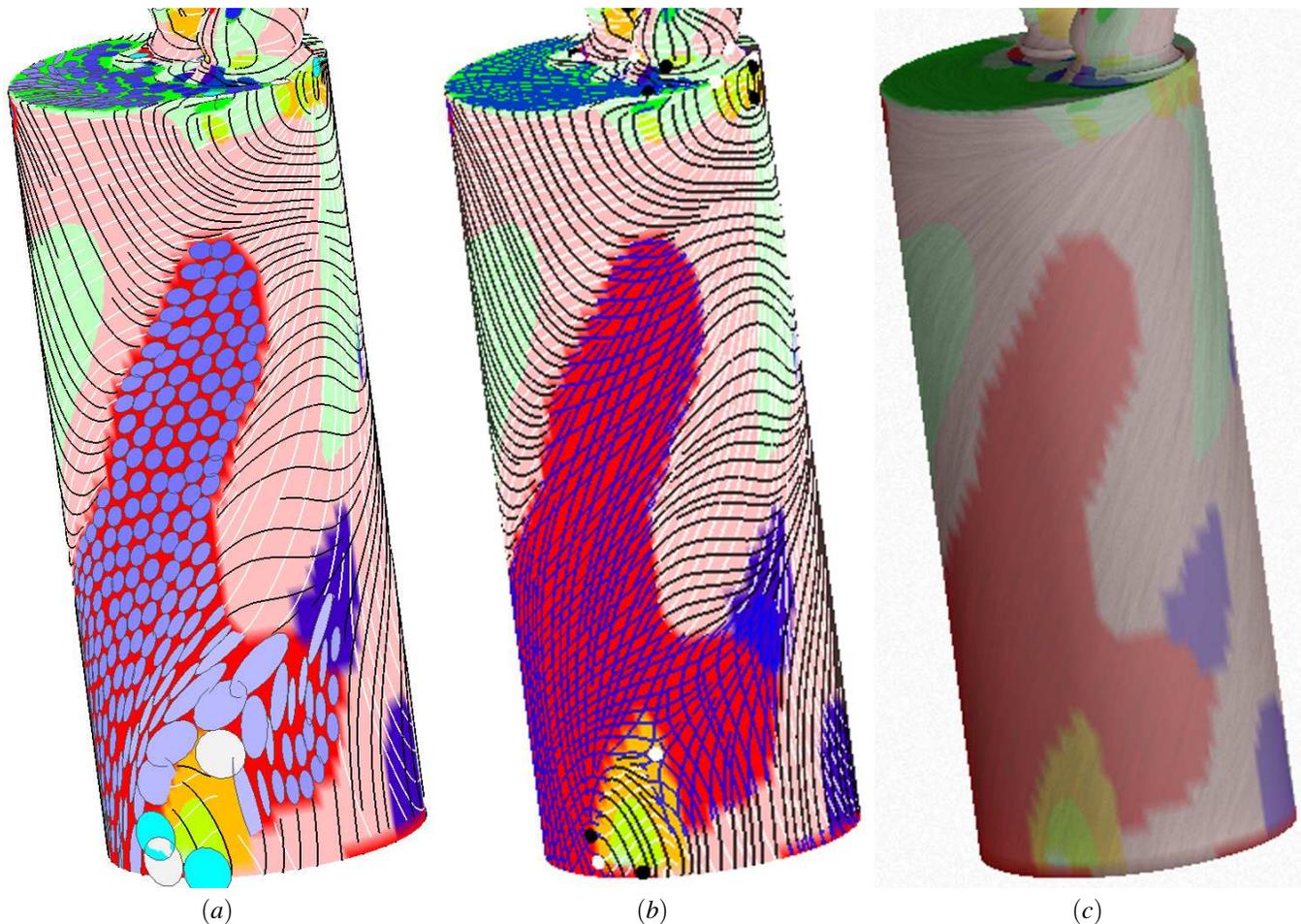


Fig. 8. Visualization of a diesel engine data set on the cylinder surface: (a) the present hybrid visualization of the velocity gradient tensor with variable hyperstreamline density and glyph size, (b) the previous visualization of the gradient tensor with eigenvectors in the real domains and pseudo eigenvectors in the complex domains this image was presented in [1], (c) the visualization of the velocity vector with textures. The background colors represent dominant eigenvalues as described in Figure 2.

- [9] X. Mao, Y. Hatanaka, H. Higashida, and A. Imamiya, "Image-Guided Streamline Placement on Curvilinear Grid Surfaces," in *Proceedings IEEE Visualization '98*, 1998, pp. 135–142.
- [10] B. Jobard and W. Lefer, "Creating Evenly-Spaced Streamlines of Arbitrary Density," in *Proceedings of the Eurographics Workshop on Visualization in Scientific Computing '97*, vol. 7, 1997, pp. 45–55.
- [11] B. Spencer, R. S. Laramée, G. Chen, and E. Zhang, "Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach," *Computer Graphics Forum*, vol. 28, 2009, forthcoming.
- [12] Z. P. Liu and R. J. Moorhead, II, "An Advanced Evenly-Spaced Streamline Placement Algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 965–972, 2006.
- [13] A. Mebarki, P. Alliez, and O. Devillers, "Farthest Point Seeding for Efficient Placement of Streamlines," in *Proceedings IEEE Visualization 2005*. IEEE Computer Society, 2005, pp. 479–486.
- [14] V. Verma, D. Kao, and A. Pang, "A Flow-guided Streamline Seeding Strategy," in *Proceedings IEEE Visualization 2000*, 2000, pp. 163–170.
- [15] G. Chen, K. Mischaikow, R. S. Laramée, P. Pilarczyk, and E. Zhang, "Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 769–785, 2007.
- [16] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, "Anisotropic Polygonal Remeshing," *ACM Transactions on Graphics (SIGGRAPH 2003)*, vol. 22, no. 3, pp. 485–493, Jul. 2003.
- [17] A. Hertzmann and D. Zorin, "Illustrating Smooth Surfaces," *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 2000)*, pp. 517–526, Aug. 2000.
- [18] M. Marinov and L. Kobbelt, "Direct Anisotropic Quad-Dominant Remeshing," *Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pp. 207–216, 2004.
- [19] E. Zhang, J. Hays, and G. Turk, "Interactive Tensor Field Design and Visualization on Surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 94–107, 2007.
- [20] J. Palacios and E. Zhang, "Rotational Symmetry Field Design on Surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, p. 55, 2007.
- [21] N. Ray, B. Vallet, W. C. Li, and B. Lévy, "N-symmetry Direction Field Design," *ACM Trans. Graph.*, vol. 27, no. 2, pp. 1–13, 2008.
- [22] S. Zhang, C. Demiralp, and D. Laidlaw, "Visualizing Diffusion Tensor MR Images using Streamtubes and Streamsurfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 4, pp. 454–462, Oct.-Dec. 2003.
- [23] D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead, "Visualizing Diffusion Tensor Images of the Mouse Spinal Cord," *Visualization Conference, IEEE*, pp. 127–134, 1998.
- [24] R. M. Kirby, H. Marmanis, and D. H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," in *Proceedings IEEE Visualization '99*. ACM Press, Oct. 25–29 1999, pp. 333–340.

- [25] W. Benger and H.-C. Hege, "Tensor Splats," in *Visualization and Data Analysis 2004, Proc. of SPIE*, vol. 5295, June 2004, pp. 151–162.
- [26] A. Bhalerao and C.-F. Westin, "Tensor Splats: Visualising Tensor Fields by Texture Mapped Volume Rendering," in *Sixth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'03)*, Montreal, Canada, November 2003, pp. 294–301.
- [27] G. Kindlmann, "Superquadric Tensor Glyphs," in *Proceedings of IEEE TVCG/EG Symposium on Visualization 2004*, May 2004, pp. 147–154.
- [28] G. Kindlmann and C.-F. Westin, "Diffusion Tensor Visualization with Glyph Packing," *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2006)*, vol. 12, no. 5, pp. 1329–1335, September–October 2006.
- [29] A. Hausner, "Simulating Decorative Mosaics," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2001, pp. 573–580.
- [30] E. Zhang, K. Mischaikow, and G. Turk, "Vector Field Design on Surfaces," *ACM Transactions on Graphics*, vol. 25, no. 4, pp. 1294–1326, 2006.
- [31] J. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts," in *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, 1996, pp. 1591–1595.
- [32] G. Chen, G. Esch, P. Wonka, P. Mueller, and E. Zhang, "Interactive Procedural Street Modeling," *ACM Trans. Graph.*, vol. 27, no. 3, pp. Article 103: (1–10), 2008.
- [33] R. S. Laramée, D. Weiskopf, J. Schneider, and H. Hauser, "Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques," in *Proceedings IEEE Visualization 2004*, 2004, pp. 51–58.
- [34] C. Garth, R. Laramée, X. Tricoche, J. Schneider, and H. Hagen, "Extraction and Visualization of Swirl and Tumble Motion from Engine Simulation Data," in *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)*, ser. Mathematics and Visualization. Springer, 2007, pp. 121–135.
- [35] G. Chen, K. Mischaikow, R. S. Laramée, P. Pilarczyk, and E. Zhang, "Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 769–785, jul–aug 2007.



**Darrel Palke** received a BS degree in computer science and math, summa cum laude, from Western Oregon University in 2008. He is currently attending Oregon State University as a Masters student and working with Eugene Zhang on tensor field visualization. His research interests include computer graphics, scientific visualization, geometric modeling and rendering.



**Guoning Chen** received a B.S. degree in Information and Communication in 1999 from Xi'an Jiaotong University and a M.S. degree in Control Theory and Engineering/Computer Application in 2002 from Guangxi University in China. He worked as an instructor in the Department of Computer Science at Guangxi University from 2002 to 2004. He joined Oregon State University in 2004 and graduated with a PhD degree in Computer Science in 2009. He is currently a postdoctoral fellow at the Scientific Computing and Imaging (SCI) institute at the University of Utah. His research interests include scientific visualization and computer graphics.



**Zhongzang Lin** received his BS degree from Zhejiang University in 2006. He is pursuing a PhD degree in computer science at Oregon State University. He is currently working with Eugene Zhang on tensor field analysis and visualization. His research interests include scientific visualization, computer graphics and geometric modeling.



**Harry Yeh** received his PhD degree in Civil Engineering in 1983 from the University of California, Berkeley. His research interests are in the areas of environmental fluid mechanics, water wave phenomena, wind turbulence, and tsunami hazard mitigation. He was a Hydraulic Engineer at Bechtel Inc., San Francisco from 1977 to 1983. From 1983 to 2002, he was a professor at the University of Washington, Seattle. He is currently the Edwards Chair Professor in Engineering at Oregon State University.



**Robert S. Laramée** received a bachelors degree in physics, cum laude, from the University of Massachusetts, Amherst in 1997. In 2000, he received a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics and Algorithms in 2005. From 2001 to 2006 he was a researcher at the VRVis Research Center ([www.vrvis.at](http://www.vrvis.at)) and a software engineer at AVL ([www.avl.com](http://www.avl.com)) in the department of Advanced Simulation Technologies. Currently he is a Lecturer (Assistant Professor) at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science. His research interests are in the areas of scientific visualization, computer graphics, and human-computer interaction.



**Eugene Zhang** received the PhD degree in computer science in 2004 from Georgia Institute of Technology. He is currently an assistant professor at Oregon State University, where he is a member of the School of Electrical Engineering and Computer Science. His research interests include computer graphics, scientific visualization, and geometric modeling. He received an National Science Foundation (NSF) CAREER award in 2006. He is a member of the IEEE and ACM.