

Superquadric Glyphs for Symmetric Second-Order Tensors

Thomas Schultz and Gordon L. Kindlmann

Abstract—Symmetric second-order tensor fields play a central role in scientific and biomedical studies as well as in image analysis and feature-extraction methods. The utility of displaying tensor field samples has driven the development of visualization techniques that encode the tensor shape and orientation into the geometry of a tensor glyph. With some exceptions, these methods work only for positive-definite tensors (i.e. having positive eigenvalues, such as diffusion tensors). We expand the scope of tensor glyphs to all symmetric second-order tensors in two and three dimensions, gracefully and unambiguously depicting any combination of positive and negative eigenvalues. We generalize a previous method of superquadric glyphs for positive-definite tensors by drawing upon a larger portion of the superquadric shape space, supplemented with a coloring that indicates the tensor's quadratic form. We show that encoding arbitrary eigenvalue sign combinations requires design choices that differ fundamentally from those in previous work on traceless tensors (arising in the study of liquid crystals). Our method starts with a design of 2-D tensor glyphs guided by principles of symmetry and continuity, and creates 3-D glyphs that include the 2-D glyphs in their axis-aligned cross-sections. A key ingredient of our method is a novel way of mapping from the shape space of three-dimensional symmetric second-order tensors to the unit square. We apply our new glyphs to stress tensors from mechanics, geometry tensors and Hessians from image analysis, and rate-of-deformation tensors in computational fluid dynamics.

Index Terms—Tensor Glyphs, Stress Tensors, Rate-of-Deformation Tensors, Geometry Tensors, Glyph Design.

1 INTRODUCTION

In tensor visualization, glyphs are the method of choice to locally display the full tensor information at a discrete set of points, by encoding all its degrees of freedom onto the shape and appearance of some base geometry [47]. Even though glyphs alone rarely provide a self-contained answer to a scientific question, they can help build understanding and intuition about scientific data and the patterns within it. When inspecting empirically measured data, glyphs allow one to visually evaluate data quality and detect measurement artifacts. Glyphs can also provide a useful reference point for understanding tensor data when creating new tensor visualization methods. Tensors also arise as ingredients in scalar and vector field analysis algorithms, where glyphs can help to monitor the progress and outcome of the analysis.

The utility of tensor glyphs has led to the development of a variety of glyph-based visualization methods [16, 59, 18, 28, 40, 31], but most of them have either concentrated on the positive-definite case, such as diffusion tensors [2], or resorted to showing eigenvalue sign through color alone. In this work, we apply mathematical principles of symmetry and continuity to evaluate previous methods, and to design an extension of a previous superquadric glyph [28] to symmetric second-order tensors with both positive and negative eigenvalues. Our method allows tensor glyphs to better depict stress tensors [18], rate-of-deformation tensors [34], geometry tensors [32], and Hessians (second derivatives of scalar fields) [38].

These are domains in which eigenvalue signs indicate important qualitative aspects. In geometry tensors and Hessians, eigenvalue sign represents the difference between locally convex and concave surfaces and functions. Compressive and tensile stress are distinguished by eigenvalue sign of stress tensors, and in rate-of-deformation tensors it separates compression from stretch of a volume element. For applications where eigenvalue signs are so crucial to interpreting the data, our method conveys them clearly by using eigenvalue signs to determine glyph *shape*, in addition to glyph color.

Glyph design is a creative process for which no unique “correct” so-

lution exists. However, various constraints are imposed by tensor algebra and generally accepted rules of good visualizations. In Section 2, we aim to make it as transparent as possible what these constraints are and which additional design goals we follow. After reviewing previous work in Section 3, our new method is described in Section 4 and justified carefully with respect to these high-level goals. To confirm that our glyphs can be used in a wide variety of contexts, Section 5 presents results from various applications.

2 TENSOR ALGEBRA AND GLYPH DESIGN

A symmetric second-order tensor \mathbf{D} can be decomposed into real eigenvalues λ_i ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) and corresponding orthonormal eigenvectors \mathbf{e}_i . With respect to a fixed coordinate system, this eigensystem decomposition can be written by stacking the \mathbf{e}_i as columns into a rotation matrix \mathbf{R} and collecting the λ_i in a diagonal matrix Λ :

$$\mathbf{D} = \mathbf{R}\Lambda\mathbf{R}^T \quad (1)$$

A tensor is positive-definite when $\lambda_i > 0$ for all i , negative-definite when $\lambda_i < 0$, and indefinite if it has both positive and negative eigenvalues. An *eigenplane* is a plane spanned by any pair of eigenvectors. The Frobenius norm $\|\mathbf{D}\|$ of the tensor is given by

$$\|\mathbf{D}\| = \sqrt{\sum_i \lambda_i^2} \quad (2)$$

Based on (1) and (2), we distinguish the overall *scale* of the tensor (given by its norm), tensor *orientation* (given by its eigenvectors), and tensor *shape*. In this work, shape denotes the part of the tensor that is invariant under rotation and uniform scaling, and is described by the normalized eigenvalues $\tilde{\lambda}_i = \lambda_i / \|\mathbf{D}\|$. Permutations of eigenvalues are equivalent to rotations that preserve the set of eigenvectors, so *sorted* normalized eigenvalues provide a non-redundant representation of the range of tensor shapes.

Our tensor glyphs are designed around mathematical considerations of the eigensystem and its particular symmetries, as well as more general principles of visualization. The first principle is that of *preservation of symmetry*: tensor glyphs should exhibit the same symmetries as the underlying tensor, no more or less. We view tensor symmetry preservation as a special case of the general goals of visualizing only intrinsic data properties while avoiding misleading artifacts, and using glyphs to completely depict all data properties. If $G(\mathbf{D})$ is the glyph for tensor \mathbf{D} and \mathbf{T} is an isometric linear transformation (such as a rotation or reflection), preservation of symmetry is formalized as

$$\mathbf{D} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1} \iff G(\mathbf{D}) = \mathbf{T}G(\mathbf{D}) \quad (3)$$

- The authors are with the Computer Science Department and the Computation Institute, University of Chicago.
E-mail: {t.schultz, glk}@uchicago.edu.
- Both authors have contributed equally to this work.

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Symmetry preservation has particular implications for how the eigensystem is visualized. Due to their lack of orientation, \mathbf{e}_i and $-\mathbf{e}_i$ are equivalent eigenvectors of a tensor. Consequently, tensor glyphs should exhibit mirror symmetry with respect to all eigenplanes. This condition is fulfilled by all tensor glyphs we are aware of. When two (or three) eigenvalues are equal, the corresponding eigenvectors span a two- (three-)dimensional eigenspace. Rotating them within that space leaves the tensor value unchanged. Tensor glyphs should reflect this continuous rotational symmetry by avoiding to set apart any particular direction within the eigenspace. This condition should only be violated in applications where the case of equal eigenvalues cannot occur. In order to produce glyphs for a wide variety of applications, we include symmetry preservation as a design goal of our work.

Two further principles for the design of visualizations can be summarized as *continuity* and what we term *disambiguity*: Glyphs for nearly equal tensors should look nearly the same, and glyphs for different tensors should look different. Without providing a precise quantification, these principles can be expressed in a formulaic manner as:

$$\mathbf{D}_1 \approx \mathbf{D}_2 \iff \text{appearance}(G(\mathbf{D}_1)) \approx \text{appearance}(G(\mathbf{D}_2)) \quad (4)$$

$$\mathbf{D}_1 \neq \mathbf{D}_2 \iff \text{appearance}(G(\mathbf{D}_1)) \neq \text{appearance}(G(\mathbf{D}_2)) \quad (5)$$

We feel that continuity and disambiguity, even if not formulaically expressed as such, are commonly used visualization guidelines: slight perturbations of the data should not significantly change the visualization (continuity), and visualizations should help discriminate between different data values (disambiguity). We adopt these principles in the interests of making the tensor glyphs as informative as possible.

Our last two glyph design principles, invariance under *scaling* and *eigenplane projection*, are more specific to tensor visualization. Though they may be violated for legitimate reasons in specific contexts [25], these principles are fulfilled by the majority of existing tensor glyphs, and we adopt them here in the interests of generality across applications of tensor visualization. Scale invariance means that a uniform scaling of the tensor \mathbf{D} should result in a uniform scaling (preserving the aspect ratio) of its glyph $G(\mathbf{D})$. In other words, a scale invariant base glyph B is scaled by a function $s(\|\mathbf{D}\|)$ of tensor norm:

$$G(\mathbf{D}) = s(\|\mathbf{D}\|)B\left(\frac{\mathbf{D}}{\|\mathbf{D}\|}\right). \quad (6)$$

In many cases, $s(\|\mathbf{D}\|)$ can be a simple multiple of tensor norm ($s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|$), though any monotonic function will permit qualitative comparisons of tensor scale. When tensor norm varies widely over a dataset, s can effectively compress its range as with $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|^\gamma$; $\gamma < 1$ (cf. Section 5). s can be set to a constant if tensor scale is irrelevant to a visual analysis task [35]. Since the base geometry depends on normalized eigenvalues, it is undefined when $\|\mathbf{D}\| = 0$. When approaching this case, we blend the base geometry with a sphere. The technical details of this interpolation are described in Section 4.3.

A tensor is projected to one of its eigenplanes by setting the remaining third eigenvalue to zero. We call a glyph invariant under eigenplane projection when projecting the tensor in this way has the same effect on the glyph as a corresponding orthogonal projection of the glyph geometry. Let \mathbf{P} denote the projection to one of the eigenplanes:

$$\mathbf{P} = \mathbf{I} - \mathbf{e}_1\mathbf{e}_1^\top \quad (7)$$

where \mathbf{I} is the identity matrix. \mathbf{P} projects to the plane orthogonal to eigenvector \mathbf{e}_1 . Then, invariance under \mathbf{P} can be written as

$$G(\mathbf{PDP}^\top) = \mathbf{P}G(\mathbf{D}). \quad (8)$$

We adopt this invariance as a glyph guideline because it ensures that a consistent set of two-dimensional glyphs are generated by the visualization of three-dimensional tensors with one zero eigenvalue.

A previous approach by Kindlmann [28] uses superquadrics [1] to simultaneously pursue similar design goals, though only for *positive-definite* tensors \mathbf{D} , by constructing glyphs $G(\mathbf{D})$ with

$$G(\mathbf{D}) = s(\|\mathbf{D}\|)\mathbf{R}\tilde{\Lambda}B(\tilde{\lambda}_i). \quad (9)$$

Specifically, it defines a superquadric base geometry $B(\tilde{\lambda}_i)$ which is parametrized by the normalized eigenvalues $\tilde{\lambda}_i$, exhibits all required symmetries, and is invariant under projections to eigenplanes. It uses different geometric shapes to disambiguate tensor shape (5), while providing a smooth transition between them to ensure continuity (4). Scale invariance follows from scaling the base geometry with the normalized eigenvalues in $\tilde{\Lambda}$, before rotating it with \mathbf{R} to match tensor orientation and applying a final scale factor $s(\|\mathbf{D}\|) = c\|\mathbf{D}\|$.

Our work extends this framework to the full space of three-dimensional symmetric second-order tensors, by extending the definition of $B(\tilde{\lambda}_i)$ beyond the set of positive-definite tensors, while preserving the design goals followed in the original approach. The previous approach [28] satisfies invariance under eigenplane projection, even though it was never stated as such. Our current work uses eigenplane projection invariance as a technique to create approximations of three-dimensional glyphs (which we call “scaffolds”) from three orthogonal two-dimensional glyphs of the tensor under eigenplane projections.

3 PREVIOUS WORK

Section 3.1 places glyph-based methods into the broader context of tensor visualization. Since an exhaustive overview of the field is outside the scope of this paper, we refer to existing reviews [57] for further references. More specifically, Section 3.2 clarifies our contribution with respect to previous glyphs for indefinite tensors.

3.1 Tensor Visualization

Methods for tensor visualization have been created in the context of various applications, including tensors that arise in computational fluid dynamics [34], geomechanics [26], general relativity [3], diffusion tensor magnetic resonance imaging [56], MRI-based strain rate imaging [51], image processing and computer vision [55], and the study of nematic liquid crystals [25].

A rough overview of previous work on tensor visualization can be given in analogy to an existing classification of techniques in flow visualization [43]: *Direct methods* like color mapping [42], volume rendering [30] or tensor splats [3] continuously depict a large part of the field by mapping local tensor values to attributes like color and opacity. Similarly, *image-based techniques* like brush strokes [35], reaction-diffusion textures [30] or variants of line integral convolution and spot noise [23, 61, 22] produce a dense image of the field.

Geometry-based techniques construct geometric objects whose shape, orientation and color convey tensor attributes. Tensor glyphs belong to this group. Unlike direct and image-based methods, they allow the user to see all attributes of a tensor, but only in discrete locations. Optionally, glyphs can be combined with continuous renderings for context [51], or arranged in a way that emphasizes non-local structures via glyph packing [31] or anisotropic noise sampling [13]. Other important geometry-based methods are hyperstreamlines [7] and hyperstreamsurfaces [26], which concentrate on conveying trajectories in one of the eigenvector fields. Hybrid methods can use both geometry-based and direct methods to combine the benefits of each [51, 8].

Feature-based methods like tensor topology [20], crease extraction [29], segmentation [12] and streamline clustering [9] extract patterns from the data which are meaningful for a specific research problem that involves a specific type of tensor data. Such high-level tensor visualization frequently requires complex algorithms. More basic techniques, including glyphs, play an important part in their debugging and validation [33, 49].

3.2 Glyphs for Indefinite Tensors

Similar to our own work, Jankun-Kelly and Mehta [25] have considered superquadric glyphs for indefinite tensors. Their approach also conforms to the rules of preservation of symmetry, continuity and disambiguity, but it only applies to traceless tensors, a five-dimensional subspace in the six-dimensional space of symmetric tensors. Therefore, our work requires a fundamentally different design. Other work has used superquadric-based glyphs outside of tensor visualization [50, 44], but does not make design decisions relevant to our problem.

Many other tensor glyphs are naturally expressed in terms of (9) with different choices of base geometry B . Various applications [2, 18, 54] have used the unit sphere, which results in ellipsoids whose axes are aligned with the tensor's eigenvectors and scaled with eigenvalues. Other choices are a coordinate cross [51] (sometimes called “hedgehog” [26]), tripod [36], the unit cube [46], a clipped plane [40] or a cylinder [60]. Alternatively, some authors combine different base shapes B , modulated by different aspects of the tensor [16, 59].

Previous work has commonly used color to indicate eigenvalue signs [16, 26, 51, 18, 40]. Alternatively, some authors enforce positive-definiteness for visualization purposes, for example by taking the exponential of eigenvalues [34], by shifting them by the amount of the most negative eigenvalue [52], or other application-specific mappings [22, 21, 33]. An entirely different approach is to define symmetric tensor glyphs via algebraic equations, such as the quadratic surface of the tensor, known as the Cauchy stress quadric in geomechanics [18] and as the Dupin indicatrix in differential geometry [17], or the Reynolds glyph [18]. Section 4.1 describes these methods in comparison with our approach. Glyphs for tensors of order higher than two [41, 48] are outside the scope of our current work.

Closely related to our work are glyph-based methods for visualization of the vector field Jacobian, which in general is not only indefinite, but also non-symmetric. When all eigenvalues are real, Globus et al. [15] display a simple coordinate cross with arrows to indicate eigenvalue signs. The probe by de Leeuw and van Wijk [6] is designed for the simultaneous visualization of the vector field and its Jacobian and does not apply to the visualization of tensor fields alone. Icons proposed by Theisel et al. [54] are based on ellipsoids, which suffer from visual ambiguities addressed by superquadric glyphs [28].

4 METHODS

4.1 Two-dimensional glyphs

Our new tensor glyph is motivated by a review of existing glyphs with respect to the design guidelines from Section 2. Focusing first on the two-dimensional case simplifies the discussion, and provides a scaffold for three-dimensional glyphs by invariance under eigenplane projection (8). Unit-norm tensors are sufficient for considering glyph shape by scale-invariance (6). Since tensor shape is independent of eigenvalue ordering, we restrict ourselves to the unit half-circle of sorted eigenvalue pairs $\lambda_1 \geq \lambda_2$.

Fig. 1 illustrates the unit-norm tensors we use to consider 2-D glyph design. Tick-marks on the lower-right half-circle in Fig. 1(a) indicate the eigenvalue pairs that sample, in clockwise order, the range of two-dimensional tensor shapes visualized in Fig. 1(b) and subsequent figures. The top row of glyphs is positive-semidefinite, the middle row is indefinite, and the third row is negative-semidefinite.

In this work, each point \mathbf{p} of a glyph surface $G(\mathbf{D})$ is colored according to the sign of the quadratic form

$$h(\mathbf{p}) = \mathbf{p}^T \mathbf{D} \mathbf{p}. \quad (10)$$

Due to the one-to-one relation between symmetric tensors and their quadratic forms, this color scheme naturally inherits all required symmetries and continuities of Section 2. The quadratic form evaluates to eigenvalues at eigenvectors ($\mathbf{e}_i^T \mathbf{D} \mathbf{e}_i = \lambda_i$) while creating a coloring that respects the relative magnitudes of the eigenvalues. In particular, the coloring is continuous as per (4) because the angular range around an eigenvector colored by the corresponding eigenvalue's sign goes to zero with the eigenvalue magnitude. Throughout this work, orange indicates positive, and blue shows a negative sign.

We observe that when indicating eigenvalue sign by color alone, the glyph shape may suggest symmetries that the tensor does not actually possess. In the linearly-scaled ellipses (Fig. 1(b)), this becomes apparent with the indefinite tensor $\lambda_1 = -\lambda_2$ (center of middle row), visualized with a circle that differs only in coloring from the isotropic cases ($\lambda_1 = \lambda_2$). With its *continuous* rotational symmetry, this circle violates symmetry preservation (3) and disambiguity (5). Thus, eigenvalue coloring *alone* does not solve the problem of tensor glyph design. More work is required to satisfy the principles of Section 2.

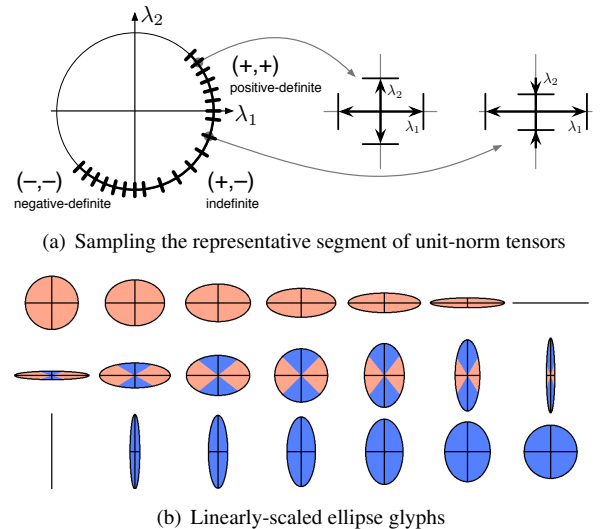


Fig. 1. Our 2-D glyph design starts by sampling the range of unit-norm tensors (a), visualized by standard linear ellipses in (b). Except for coloring by quadratic form, ellipse glyphs misleadingly imply rotational symmetry when there is in fact the least rotational symmetry, at $\lambda_1 = -\lambda_2$.

Fig. 2 illustrates some previous methods that avoid this misleading symmetry. Fig. 2(a) uses ellipses to show \mathbf{D} by visualizing $\exp(\mathbf{D})$, a method suggested by Kirby et al. [34] for rate-of-deformation tensors that maps all eigenvalues (by $\lambda_i \rightarrow e^{\lambda_i}$) to positive values. A limitation of this approach is that all the glyphs in the middle row (for indefinite tensors) have similar aspect ratios, even though the underlying tensor shape is widely varying, undermining disambiguity (5).

Fig. 2(b) shows the Reynolds glyph [18], defined as $G(\mathbf{D}) = \{(\mathbf{v}^T \mathbf{D} \mathbf{v}) \mathbf{v} \mid \|\mathbf{v}\| = 1\}$. This is equivalent to a polar plot of the quadratic form $h(\mathbf{v})$ over all points \mathbf{v} on the unit-circle. Its four-lobed shape in the indefinite case (middle row) clearly disambiguates it from positive- and negative-definite tensors (5), and since it is based on the quadratic form, the Reynolds glyph has the same symmetries as the tensor (3). Unfortunately, in three-dimensions, the wide lobes of the Reynolds glyph can create ambiguity by self-occlusion (Section 5).

More dramatic disambiguation is given by the Dupin Indicatrix, defined as the positions \mathbf{v} satisfying $\mathbf{v}^T \mathbf{D} \mathbf{v} = \pm 1$. The two distinct solutions are shown in the positive and negative colors in Fig. 2(c). In differential geometry, this indicatrix shows local surface curvature [17]. It has also been suggested (though not commonly adopted) for three-dimensional tensors [10]. A drawback is that the indicatrix width along an eigenvector varies inversely with the eigenvalue magnitude, which may be counter-intuitive. This problem is fixed with an implicit formulation of the ellipse glyph in Fig. 2(d), defined by $\{\mathbf{v} \mid \mathbf{v}^T \mathbf{D}^{-2} \mathbf{v} = \pm 1\}$, wherein \mathbf{D}^{-2} is sign preserving: $\lambda_i(\mathbf{D}^{-2}) = \text{sgn}(\lambda_i(\mathbf{D})) / \lambda_i(\mathbf{D})^2$. Like the Dupin Indicatrix, however, this implicit surface becomes an unbounded hyperbola in the indefinite case.

All presented glyphs use a circle to represent isotropy, no matter if the two equal eigenvalues are positive or negative. A circular glyph for depicting isotropy is in fact a consequence of symmetry preservation (3). Previous work [54] has used a circle with a sawtooth-like profile to indicate complex eigenplanes. Even though such shapes violate rotational symmetry in the strictest sense of (3), they still perceptually suggest it, and could in principle be used to indicate negative-definite isotropy. However, we still believe that circles (and spheres, in three dimensions) provide the best visualization of isotropy. The main reason for this is that isotropic tensors completely lack directional information, making it difficult to consistently orient any other shape or texture in animations or other smooth transitions.

Therefore, glyph shape alone cannot indicate eigenvalue sign. We propose to use glyph shape to show eigenvalue sign *differences*: the shape between two eigenvectors is convex if the corresponding eigenvalues have the same sign, and concave if they are different.

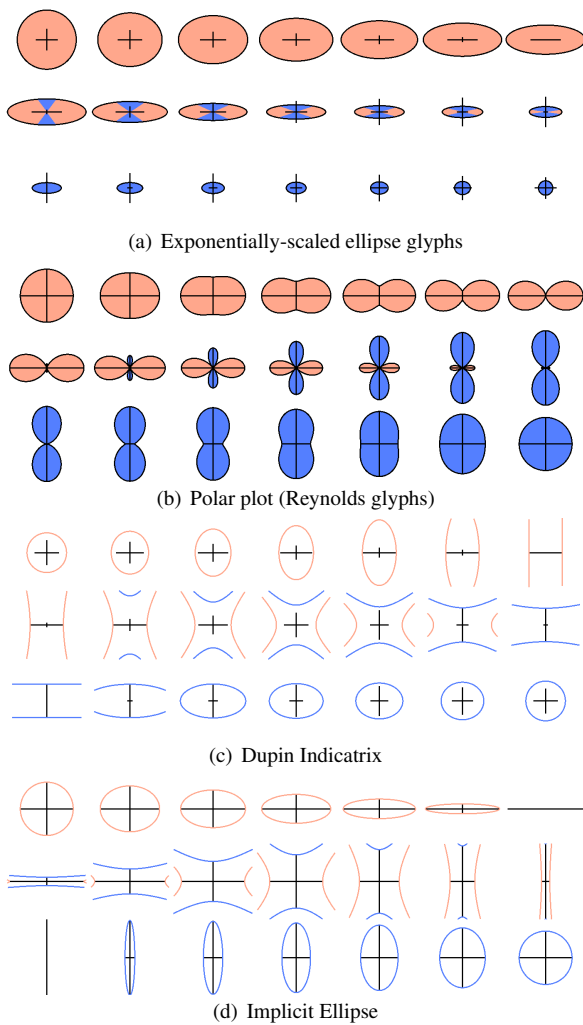


Fig. 2. Other tensor glyphs have benefits and drawbacks that inform our design process. Exponentially-scaled ellipses (a) and polar plots (b) avoid indicating symmetry at $\lambda_1 = -\lambda_2$, but with shapes that vary only slightly when one eigenvalue is near zero. The Dupin Indicatrix and (c) the implicit ellipse (d) produce fundamentally different shapes for indefinite tensors, at the cost of unbounded surfaces.

We propose the superquadric tensor glyphs shown in Fig. 3 as one solution of the design goals in Section 2. For positive- and negative-definite tensors (top and bottom row), it coincides with previous superquadric glyphs [28], with color indicating eigenvalue sign. The novelty is using a four-pointed concave shape to handle mixed eigenvalue signs, inspired by the axis-alignment of the four-lobed polar plot (Fig 2(b)) and the shape of the Dupin Indicatrix (Fig 2(c)). The concavity emphasizes the absence of continuous rotational symmetry, preserves the aspect ratio of the eigenvalue magnitudes, and avoids confusion with the convex shapes for positive- and negative-definite tensors. The concavity diminishes as one eigenvalue tends towards zero, creating narrow diamonds. Though distinct from the narrow rectangles at the nearby positive-definite and negative-definite cases, both shapes degenerate to a line segment when one eigenvalue is zero, ensuring complete continuity (4) across all eigenvalue sign combinations.

This glyph does have drawbacks: as with the polar plot and the Dupin Indicatrix, except for the coloring there is 90-degree rotational symmetry at $\lambda_1 = -\lambda_2$, which weakens symmetry preservation (3). Also, the concavities with mixed eigenvalue signs decrease the visibility of the glyph compared to the positive- and negative-definite tensors with the same norm, which unfortunately reduces the visual effect of scale preservation (6). As the best solution we have yet found, how-

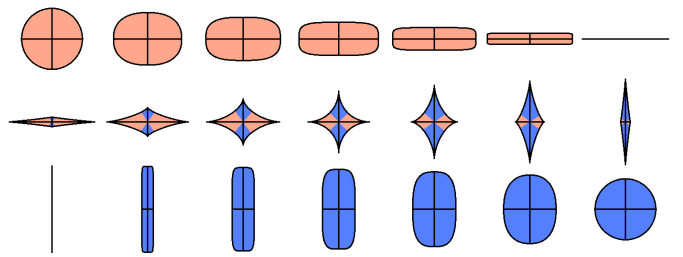


Fig. 3. Our novel glyphs augment an existing superquadric-based approach with concave shapes for indefinite tensors and coloring from the quadratic form.

ever, we use this two-dimensional glyph as the scaffold for creating the three-dimensional tensor glyphs described next.

4.2 Shape Space for 3-D Symmetric Tensors

Our three-dimensional glyph design builds upon the two-dimensional glyphs in Fig. 3. Just like Sect. 4.1 explored two-dimensional tensor shape in terms of sorted eigenvalues $\lambda_1 \geq \lambda_2$ on the unit circle, for three-dimensional tensors we consider sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, on the unit sphere ($\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} = 1$). Fig. 4 visualizes the resulting set of tensor shapes as a two-dimensional surface in $\tilde{\lambda}_i$ eigenvalue space. The three-dimensional analog to the lower half-circle in Fig. 1(a) is the spherical lune formed by the intersection of the two hemispheres in which $\tilde{\lambda}_1 \geq \tilde{\lambda}_2$ and $\tilde{\lambda}_2 \geq \tilde{\lambda}_3$. The resulting lune is shown from two different viewpoints in Fig. 4(a). Considering the isotropic positive ($\tilde{\lambda}_i = 1/\sqrt{3}$) and isotropic negative ($\tilde{\lambda}_i = -1/\sqrt{3}$) points as the north and south poles of a tensor shape globe, the Mercator projection [58] flattens the lune to the shape in Fig. 4(b).

Though not the central contribution of this work, we suggest that the Mercator projection is a novel and natural way to visualize the range of three-dimensional symmetric tensor shapes. A comparable view of the range of 3-D symmetric tensor shapes was illustrated for image analysis by Danielsson and Lin [5], but the space was parameterized with transforms of spherical harmonics, rather than the direct geometric mapping we use. Mercator maps of tensor shape similar to those described in this Section provided the visual context for the original design and evaluation of our new glyphs. The planar layout of the Mercator projection preserves the basic shape of the spherical lune, making it easier to demarcate and reason about the triangular zones (colored here arbitrarily) bounded by the various eigenvalue zero-crossings and equalities. These zones delineate the regions over which our glyph base geometry is defined and interpolated.

To obtain an explicit parameterization of tensor shape better suited to glyph creation, we transform the lune to a unit square, shown in Fig. 4(c) and (d). Normalized eigenvalues are projected to a bi-unit cube that is unfolded and sheared to create a unit (u, v) square:

$$\lambda'_i = \frac{\tilde{\lambda}_i}{\max(|\tilde{\lambda}_1|, |\tilde{\lambda}_2|, |\tilde{\lambda}_3|)}; i = 1, 2, 3 \quad (11)$$

$$u = \frac{1 + \lambda'_2}{2}; v = \begin{cases} \frac{1 + \lambda'_1}{2} - u + 1 & \lambda'_1 > -\lambda'_3 \\ \frac{\lambda'_1 - 1}{2} - u + 1 & \lambda'_1 \leq -\lambda'_3 \end{cases} \quad (12)$$

There are no degeneracies in mapping forwards or backwards between the lune and the (u, v) square, so either parameterization is an equally valid representation. The square makes it more computationally convenient to represent positions inside the space, and to blend between shapes. As functions of the eigenvalues, the (u, v) shape coordinates are tensor invariants (like $\text{tr}(\mathbf{D})$ and $\|\mathbf{D}\|$), though we leave a detailed comparison between (u, v) and established invariants to future work.

For comparison we show in Fig. 4(e) the relationship between the spherical lune and domains associated with two previous variable-geometry glyphs. The barycentric triangle parameterized by the (c_l, c_p, c_s) invariants [59] underlies previous superquadric glyphs for

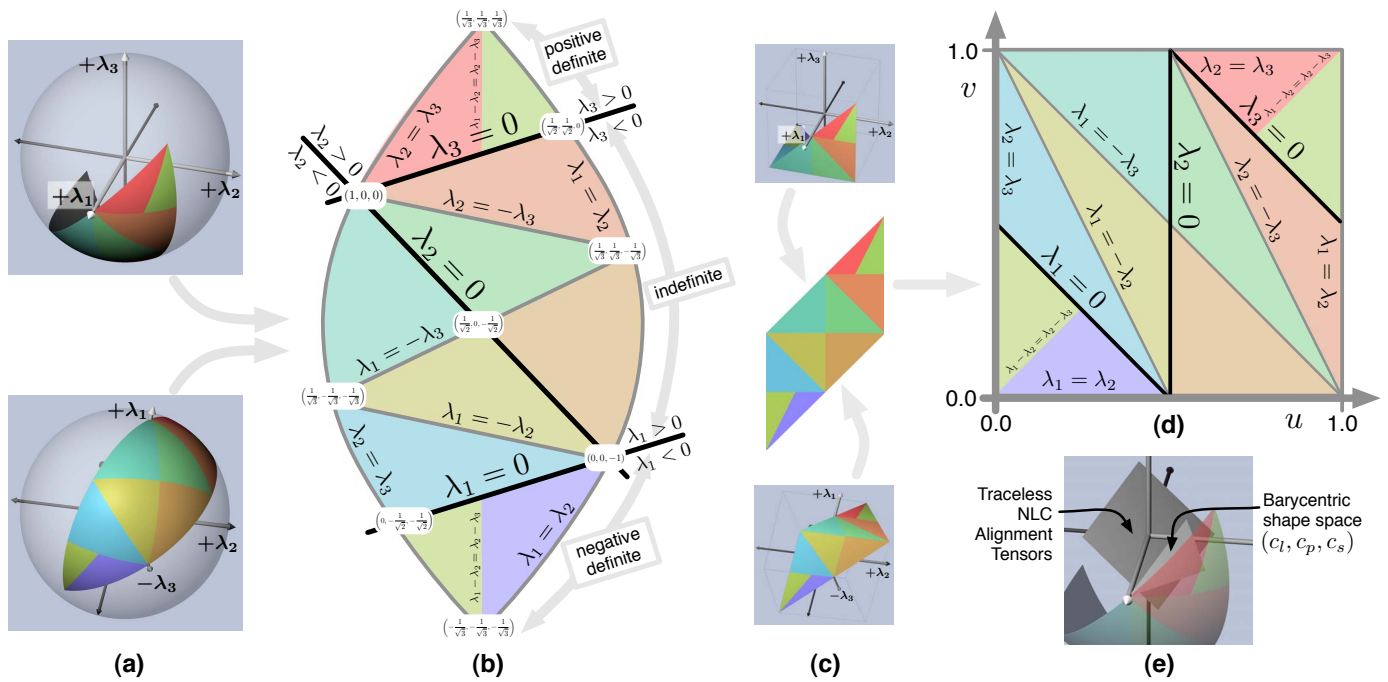


Fig. 4. Normalized and sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ form a spherical lune (a), capturing the range of 3-D symmetric tensor shape. The Mercator projection (b) lays out the space to better document the eigenvalue zero-crossings and equalities that will structure our approach. The result of projecting the lune onto the bi-unit cube and unfolding (c) can be sheared to a novel (u, v) unit square that conveniently represents all relevant tensor shapes (d). Domains of traceless NLC alignment tensors (further visualized in Fig. 11) and barycentric shape coordinates (of positive-definite tensors) are shown relative to the lune for comparison (e).

positive-definite tensors [28]. Normalization by trace ($\lambda_1 + \lambda_2 + \lambda_3$) is built into the definition of (c_l, c_p, c_s) , so all scalings of this triangle non-redundantly cover the volume of positive-definite shapes. This normalization, however, implies that (c_l, c_p, c_s) do not continuously parametrize the larger range of shapes covered by our (u, v) invariants. The square region of traceless tensors in Fig. 4(e) indicates some of the domain of nematic liquid crystal (NLC) alignment tensors studied in previous work [25]. Fig. 11 compares our new method with previous visualizations of the traceless tensors in this square domain.

4.3 Definition of New Superquadric Glyphs

Invariance under eigenplane projection (8) described in Sect. 2 implies that three intersecting orthogonal 2-D glyphs, each visualizing one of the three eigenplane projections of a 3-D tensor, are an initial approximation or scaffold for the desired 3-D glyph base surface. For a range of tensors sampled in the projected lune of tensor shapes, these scaffolds are shown in Fig. 5(a), and the subsequent parts of this figure represent the core contribution of our work. Guided by the orthogonal eigenplane glyphs, we find the vertices and the triangles in the (α, β) superquadric parameter space that span the possible glyph base shapes (Fig. 5(b)). For a given (α, β) value, the superquadric is explicitly parameterized as [1]

$$\mathbf{q}(\alpha, \beta) = \begin{pmatrix} \cos(\theta)^\alpha \sin(\phi)^\beta \\ \sin(\theta)^\alpha \sin(\phi)^\beta \\ \cos(\phi)^\beta \end{pmatrix}, \quad \begin{matrix} 0 \leq \phi \leq \pi \\ 0 \leq \theta \leq 2\pi \end{matrix}, \quad (13)$$

using signed exponentiation $x^\alpha = \text{sgn}(x)|x|^\alpha$.

A time-consuming part of this work was manually finding points in (α, β) parameter space to generate base glyphs satisfying the symmetries and constraints of Section 2. This required finding triangles in the (α, β) space that could be patched together to cover the lune of tensor shape, joined along edges delineated by the eigenvalue equality relationships seen in Fig. 4(a). We found that the glyph scaffolds in Fig. 5(a) provided valuable guidance in this process. For example, $\lambda_1 = \lambda_2$ dictates that $\alpha = 1$, creating continuous rotational symme-

try around the z axis for the entire right edge of the lune. The pattern of concavity and convexity then determines β for the rotationally symmetric cross-sections, made visible in the glyph scaffolds. Mixed eigenvalue signs force $\beta > 2$, which creates concavity, with the base glyph axis perpendicular to the two eigenvectors with the same eigenvalue sign. One eigenvalue tending to zero forces $\beta \rightarrow 0$ for positive- or negative-definite tensors and $\beta \rightarrow 2$ for indefinite tensors.

The final results of our glyph design are shown in Fig. 5(c), which shows both the three-dimensional glyph and (via the green and magenta colormap) the (α, β) parameters that produced it. The sharp (α, β) discontinuities visible within the lune (along the thin black lines in Fig. 5(c)) would seem to violate continuity (4). By design, however, the seams are located exactly where one eigenvalue is zero (cf. Fig. 4(b)), so that eigenvalue scaling recovers continuity of glyph shape. For example, the thin disks for planar positive-definite tensors blend smoothly into the flattened cones as λ_3 changes from positive to negative, even though this involves a jump in β from 0 to 2. Fig 5(d) documents the (α, β) control points at the corners of the (u, v) triangle vertices. The maximal value of β is recorded here as 4.0, but this value can be lowered (e.g. to 3.0) while keeping the other (α, β) control points fixed, to decrease the concavity of indefinite tensor glyphs. Fig 5(d) also identifies the regions (with a yellow inscribed triangle) that require the base glyph to be rotated 90 degrees from their original vertical z axis to a horizontal axis around x (for the minor eigenvector \mathbf{e}_3). This seam between parameterization axes was also present in the earlier superquadric tensor glyph work [28].

When tensor norm $\|\mathbf{D}\| = 0$, (u, v) coordinates are undefined. In this case, we define the base glyph to be the unit sphere. In order to ensure a continuous transition towards this case, we gradually blend the geometry with a sphere when $\|\mathbf{D}\| < \epsilon$, by interpolating the (α, β) parameters that are computed as above with $(\alpha = 1, \beta = 1)$ when approaching zero norm. The parameter ϵ is chosen such that for tensor norm smaller than ϵ , eigenvalue signs start to be dominated by numerical or measurement noise. Its exact value will typically depend on the average tensor norm and the level of noise in the data.

Fig. 5(e) illustrates one small complication to correctly wrapping

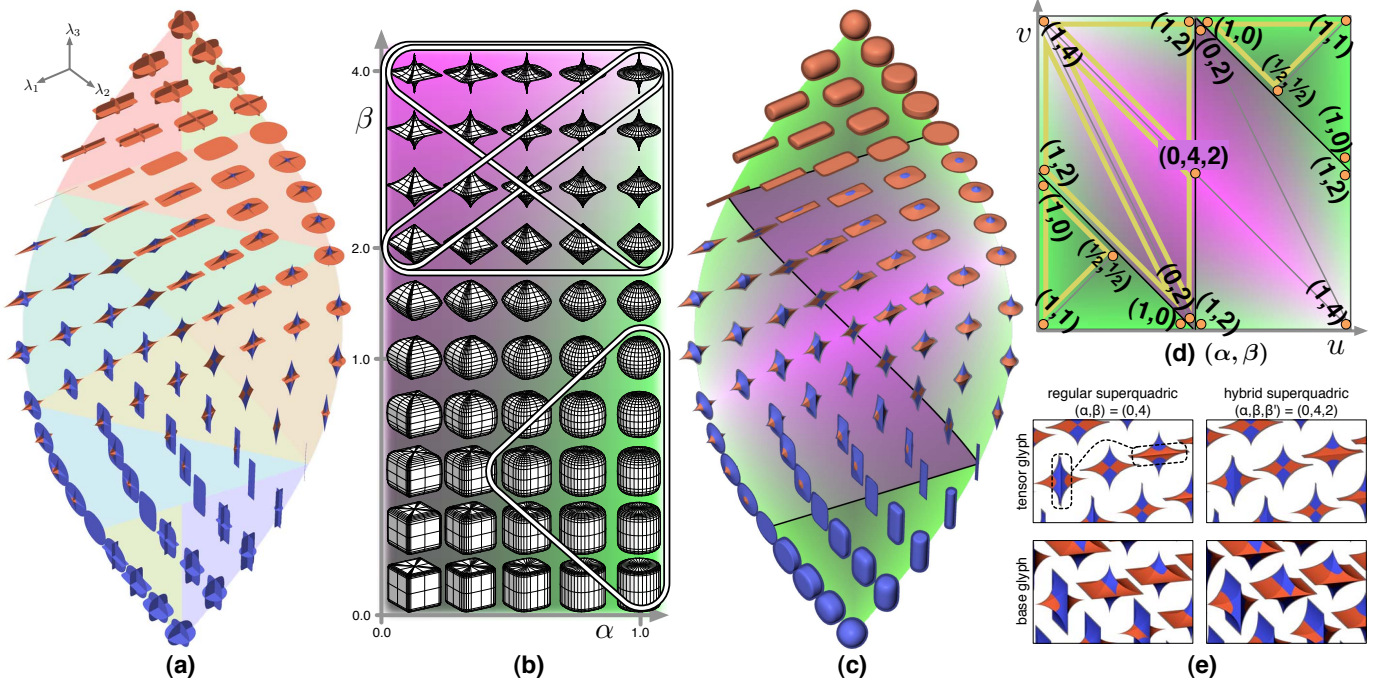


Fig. 5. Creation of the new superquadric tensor glyphs. 3-D tensors sampled on the projected lune (a) are indicated by scaffolds of 2-D glyphs showing the three orthogonal eigenplane projections, each visualized by the method sketched in Fig. 3. To find three-dimensional base glyph shapes conforming to these scaffolds, the range of superquadric shapes is sampled in (b), using green and magenta to indicate the varying α and β parameters respectively. Careful inspection and experimentation led to choosing the triangles indicated in (b), which are mapped to the triangular zones (a), to produce the new 3-D glyphs in (c). The patchwork of triangles in (α, β) parameter space, with vertices labeled for reference, is shown in (d). Yellow outlines show triangles where the base superquadric must be rotated to lie along the major eigenvector \mathbf{e}_1 rather than the minor eigenvector \mathbf{e}_3 . A small adjustment to increase shape continuity at the center of the lune is shown in (e), as described in the text.

the glyph scaffolds (Fig. 5(a)) with superquadrics. Near the center of the palette (where $\lambda_1 \approx -\lambda_3$ and $\lambda_2 \approx 0$), the cross-section across the medium eigenvector \mathbf{e}_2 needs to be the star shape (seen at the center of Fig. 3) created by $\beta = 4$. The cross-sections across the other eigenvectors \mathbf{e}_1 and \mathbf{e}_3 involve an eigenvalue λ_2 near zero, and hence should be either the narrow diamonds or skinny rectangles also seen in Fig. 3. This particular combination of cross-sections, however, does not exist in any available superquadric (Fig. 5(b)). Using $(\alpha, \beta) = (0, 4)$ creates a visibly discontinuous change in seam orientation (Fig. 5(e) upper left), and also fails to have the desired diamond cross-section in the base glyph (Fig. 5(e) lower left). Therefore for this particular tensor shape, at the $(0.5, 0.5)$ center of (u, v) shape space, we define a new *hybrid superquadric* that is parameterized by (α, β, β') :

$$y_\beta = \sin(\theta)^\alpha \sin(\phi)^\beta \quad (14)$$

$$z = \cos(\phi)^\beta \quad (15)$$

$$s_{\beta'} = \sin(\arccos(z^{1/\beta'}))^{B'} \quad (16)$$

$$s_{\max} = \sin(\phi)^\beta \quad (17)$$

$$\mathbf{h}(\alpha, \beta, \beta') = \begin{pmatrix} \cos(\theta)^\alpha \sin(\phi)^\beta \\ \frac{y_\beta s_{\beta'}}{s_{\max}} \\ z \end{pmatrix}, \quad \begin{matrix} 0 \leq \phi \leq \pi \\ 0 \leq \theta \leq 2\pi \end{matrix} \quad (18)$$

The effect of (16) is to recover from z the $\phi = \arccos(z^{1/\beta'})$ due to setting $\beta = \beta'$ in (13), then, to use this ϕ for a $\sin(\phi)^{\beta'}$ factor $s_{\beta'}$ that scales the y coordinate of \mathbf{h} . This forces the (y, z) cross-section of $\mathbf{h}(\alpha, \beta, \beta')$ to match that of $\mathbf{q}(\alpha, \beta')$, while the (x, z) cross-section matches that of $\mathbf{q}(\alpha, \beta)$. Fig 5(d) identifies $\mathbf{h}(0, 4, 2)$ as the base glyph at $(u, v) = (0.5, 0.5)$ (Fig. 5(e) lower right), producing the more continuous transition in Fig. 5(e) upper right.

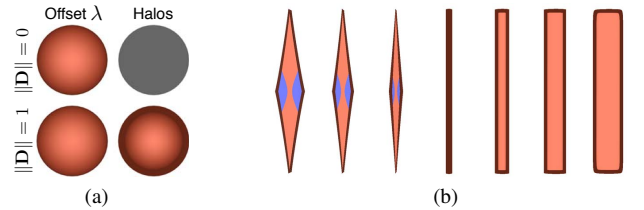


Fig. 6. Halos support glyph perception when one or more eigenvalues are near zero. Subfigure (a) shows the zero tensor (top) and the unit tensor (bottom) at different zoom levels, using offset eigenvalues (left) and halos (right). Subfigure (b) shows a transition from indefinite to positive-definite tensors, with a rank-1 tensor at the center.

4.4 Glyph Halos

A complication with glyphs that follow (9) arises when two or all three eigenvalues are near zero: In this case, the glyph degenerates to a line or a point, so it is no longer visible. In practice, even the case of a single near-zero eigenvalue can be problematic.

A simple solution to avoid this problem is to offset eigenvalues away from zero. However, it requires visual comparison to a reference size to decide how close a displayed glyph is to the zero tensor. This is illustrated in Fig. 6(a), which compares the zero tensor (top) to the unit tensor (bottom), but shows them at different zoom levels. With offset eigenvalues (left), both may result in the same visualization.

Therefore, we propose to render a halo around the tensor glyph. Halos have been used previously in visualization to enhance depth perception [53]; in our case, their goal is to ensure that even skinny structures remain visible. The extent of the halo corresponds to a glyph with offset eigenvalues (resulting in a fixed width in world space), but the visual difference between the glyph itself and its halo clearly indicates whether or not an eigenvalue is near zero (Fig. 6(a), right).

The halo can be colored to encode additional tensor information.

In our examples, a colormap of tensor trace determines halo color. The utility of this is demonstrated in Fig. 6(b), showing tensors varying gradually from indefinite (left) to positive-definite (right). For the rank-1 tensor at the center, the glyph itself vanishes visually, but its halo extent indicates both the magnitude of the non-zero eigenvalue and the direction of the associated eigenvector, while the halo color indicates the eigenvalue sign. For a rank-1 tensor, the sign of the trace coincides with the sign of the only non-zero eigenvalue.

Unlike the glyph shape itself, the shape of our halo is discontinuous with eigenvalue sign changes. In practice, this did not become apparent in our experiments, and it can be considered to reflect the fact that halo shape visualizes eigenvalue signs, which are not continuous mathematical quantities. Image space methods for halo generation [37] could be explored if, for some application, continuous halos are important. A near-zero trace is indicated by a gray halo, which allows for a convenient check of whether or not a tensor is traceless.

4.5 Glyph Generation and Rendering

The process of glyph generation for a given tensor \mathbf{D} is summarized as follows: First, we find eigenvalues λ_i via the spectral decomposition (1), sort and divide them by tensor norm $\|\mathbf{D}\|$ to obtain normalized eigenvalues $\tilde{\lambda}_i$, and map the resulting descriptor of tensor shape to (u, v) coordinates via (11). Next, we find out into which of the triangles shown in Fig. 5 (d) the values of (u, v) fall. Finally, a superquadric base geometry is created as described in Section 4.3, with parameters α and β given by barycentric interpolation within the respective triangle. For reference, Fig. 5 (d) provides the (α, β) at each triangle vertex, as well as the special (α, β, β') value at the center of the (u, v) square. When $\|\mathbf{D}\| = 0$, we set $(\alpha, \beta) = (1, 1)$, with a continuous blending towards this case as described in Section 4.3.

Our OpenGL-based implementation stores vertex positions \mathbf{p} of the glyph base geometry in the tensor's eigenframe; non-uniform scaling, rotation and global scaling (as with $s(\|\mathbf{D}\|)\mathbf{R}\tilde{\Lambda}$ in (9)) are achieved via a suitable modelview matrix. The untransformed position \mathbf{p} is passed from an OpenGL vertex shader to a fragment shader in a “varying” variable [45], where it is used to determine color. \mathbf{D} is diagonal in its eigenframe, so it suffices to transfer the three eigenvalues to the GPU (once per glyph, in a “uniform” variable) for computing the quadratic form (10) in the fragment shader. Based on the resulting sign, orange (positive) or blue (negative) is selected as the fragment color, modulated by Phong shading.

Glyph halos are rendered in a two-pass approach: The extent of the glyphs rendered in the first pass is recorded in the stencil buffer. For the second pass, glyphs are made slightly larger by offsetting the eigenvalues by a constant amount, away from zero. To achieve the desired halo effect, lighting is turned off in this second pass, and we only render to pixels for which the stencil bit is not set.

4.6 Optimized Implementation

To accelerate glyph generation, we pre-compute a palette of representative base glyph shapes $B(\tilde{\lambda}_i)$ by a uniform sampling of the (α, β) domain shown in Fig. 5 (b). To render a new tensor, we can then compute its (α, β) and simply pick the nearest base shape from the palette. Note that the final glyphs seen in the tensor visualization are not merely uniform scalings and reorientations of the shapes sampled in the palette. Rather, the palette provides the axis-aligned *base* glyphs B . These are transformed by (9) to glyphs G that accurately indicate tensor orientation, shape, and scale. Since shapes in superquadric (α, β) space vary smoothly, the differences in shape of adjacent samples will no longer be apparent visually when the discretization is fine enough. In our experience, a palette resolution around 15×45 is sufficient to make the result visually indistinguishable from the use of base geometry that is computed from scratch for each individual tensor.

To increase rendering performance, the shape palette can be kept in video memory, e.g., via OpenGL vertex buffer objects. Even with a tessellation that produces high-quality renderings (we use a single triangle strip with 801 vertices per glyph), vertex positions and normals ($6 \times 4 = 24$ bytes per vertex) for a 15×45 palette consume $801 \times 24 \times 15 \times 45$ bytes (≈ 12.7 MB) of video memory, which is very

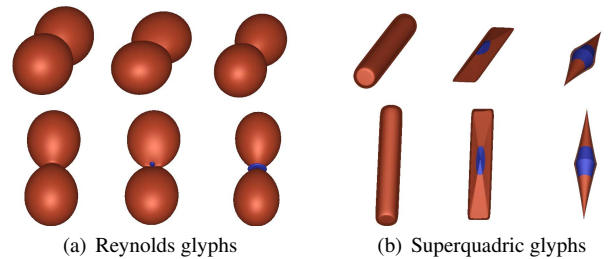


Fig. 7. The existing general glyphs for symmetric tensors, Reynolds glyphs (a), suffer significantly from ambiguity problems due to self-occlusion by the wide lobes. Both the magnitude and sign of the remaining eigenvalues can be hard to discern (top row), while from the same viewpoint, the new superquadrics (b) show these clearly.

moderate with respect to modern hardware resources and leaves space for other visualization techniques that may complement the glyphs. The same index array can be used for all shapes.

With this implementation, only a pointer into the palette, a modelview matrix, and the set of eigenvalues (for coloring) is transferred to the graphics card per tensor. Despite the two-pass rendering for the halos, we achieve interactive results (25 fps) when rendering more than 3000 glyphs simultaneously in full mesh resolution on a large viewport (1800×1000 pixels).

5 RESULTS

The polar plot seen in Fig. 2(b), sometimes referred to as a Reynolds glyph, naturally generalizes to three dimensions and is an interesting possibility for three-dimensional symmetric tensor visualization. As Fig. 7 shows, Reynolds glyphs suffer from ambiguity problems. At certain view angles (top row), the wide lobe associated with a single positive eigenvalue can hide basic information about the remaining eigenvalue magnitudes and signs. The superquadric glyphs clearly reveal linearly shaped tensors with (left to right) zero, one, or two negative eigenvalues. The same distinctions can only be inferred from the Reynolds glyph when changing the viewpoint (bottom row).

In demonstrating the new glyphs on real-world data from different contexts, our hope is to highlight possible benefits of being able to fully “see” individual symmetric tensors. A simulation of the stress tensor field resulting from a two downward compressive point loads is an especially common dataset for hyperstreamline demonstrations [7], but it is less often visualized with glyphs, in part because, as others have noted [26], existing glyphs are unsatisfactory. Fig. 8 visualizes this canonical dataset with hyperstreamlines and our new glyphs. Tensor norms are significantly compressed, by setting $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|^{1/10}$ in (6), due to the very wide variation of eigenvalues. The longest axis of glyphs in Fig. 8(a) indicates the minor eigenvector (direction of maximal compression) typically traced with hyperstreamlines. The glyphs additionally reveal the variation of stress patterns near the center of the dataset, and halos clarify the orientation of the narrowest glyphs elsewhere. The horizontal slice in Fig. 8(b) shows the radial structure of the two compressive loads, as well as locations and directions along which there is expansion (positive eigenvalues in orange). A previous glyph for stress tensor visualization in geomechanics depicts the orientation of the plane spanned by the two eigenvectors with the largest eigenvalue magnitudes [40]. By symmetry preservation (3), our glyphs also depict this orientation whenever the tensor shape strongly determines it, while also showing the principle eigenvector directions traced by hyperstreamlines.

Another tensor field more often depicted by streamlines or other methods than by individual glyphs are *geometry tensors* \mathbf{G} on smooth three-dimensional implicit surfaces:

$$\mathbf{G} = (\mathbf{I} - \mathbf{nn}^T)\mathbf{H}(\mathbf{I} - \mathbf{nn}^T)/\|\mathbf{g}\| \quad (19)$$

where $\mathbf{n} = \mathbf{g}/\|\mathbf{g}\|$ is the normalized gradient and \mathbf{H} is the Hessian of a scalar function $f(\mathbf{x})$. The eigenvalues of \mathbf{G} are zero and the principal curvatures (of the isosurfaces of f) κ_1 and κ_2 . Invariants of \mathbf{G} can be

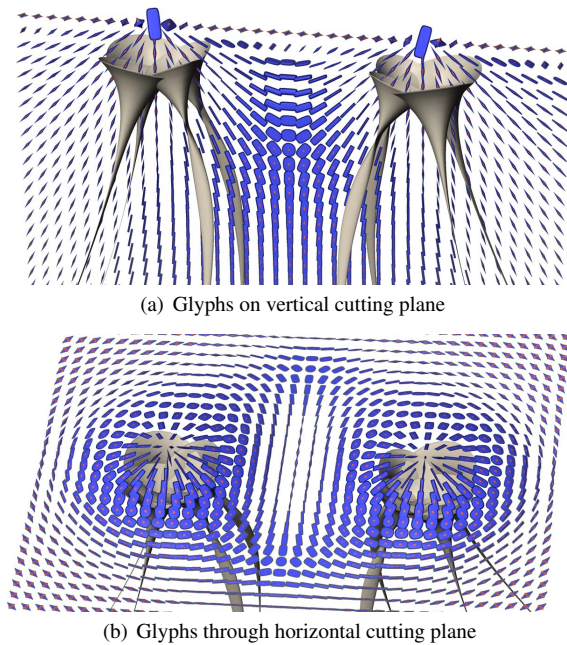


Fig. 8. Glyphs in the double point load stress tensor field reveal the minor eigenvector along which hyperstreamlines [7] are traced (a), and the variation in stress with distance from the load (b).

volume rendered [32], but its eigenvectors are commonly used in non-photo-realistic rendering, e.g. curvature-based strokes [11, 14, 19]. Inspecting geometry tensors could help debug an NPR method giving unexpected results in an unfamiliar dataset. Fig. 9(a) visualizes geometry tensors G on an isosurface (sampled by a particle system [39]) of an ear from the Visible Human male CT scan. Variations in surface curvature are reflected in the new glyphs: convex (blue circles), concave (orange circles), and saddles (orange and blue stars). For comparison, Fig. 9(b) shows the full Hessian H from which G was computed.

The new glyphs may also have a role in visualizing the tensor ingredients of image analysis methods such as edge detection. One edge definition is zero-crossing on the second directional derivative along the gradient direction, $f'' = \mathbf{n}^T \mathbf{H} \mathbf{n}$. This edge surface is sampled by a particle system [33] in Fig. 9(c), showing the Hessians at the edge locations, and revealing close similarities with the geometry tensors on the isosurface in Fig. 9(a), indicating that one of the Hessian eigenvalues is near zero even though this is not part of the edge definition. Another edge definition is the zero-crossing of the Laplacian $\nabla^2 f = \text{tr}(\mathbf{H})$, and Fig. 9(d) illustrates the difference between the Hessians on this surface and those in Fig. 9(c). The consistently gray glyph halos in Fig. 9(d) indicate that these are traceless tensors.

As a demonstration of the glyphs in a 2-D visualization, Fig. 10 visualizes a cross-section of a simulation of jet flow rightward into a steady medium, causing turbulence. Glyphs of rate-of-deformations tensors document how an infinitesimal volume is stretched or compressed as it moves along the flow. A backdrop of line integral convolution [4] (with contrast modulated by velocity) provides visual context. Fig. 10(a) uses the exponentially-scaled ellipses of [34] to map tensors with negative eigenvalues to positive-definite tensors suitable for ellipsoid visualization. When the absolute difference between eigenvalues becomes too large, these glyphs can become so stretched that they overlap each other and extend over a significant portion of the domain, undermining the locality normally enjoyed by glyphs. Such stretching also *reduces* the visual presence of the needle-like glyphs for tensors with larger norms, contrary to scale preservation (6). Fig. 10(b) uses our superquadric glyphs with $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|$. The aspect ratio reflects the relative eigenvalue magnitudes, the size correctly indicates the tensor norm, and pointed glyph shapes clearly communicate eigenvector directions. With compression of scale variation

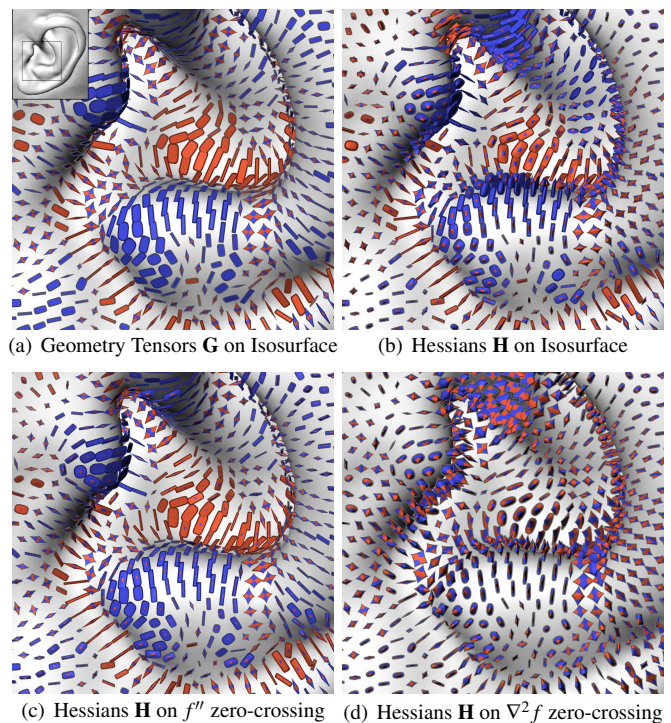


Fig. 9. Visualization of geometry tensors (a) and Hessians associated with isosurfaces (b) and two different definitions of edges, zero-crossings of the second-directional derivative (c) and the Laplacian (d). These results use $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|^{1/2}$ in (6).

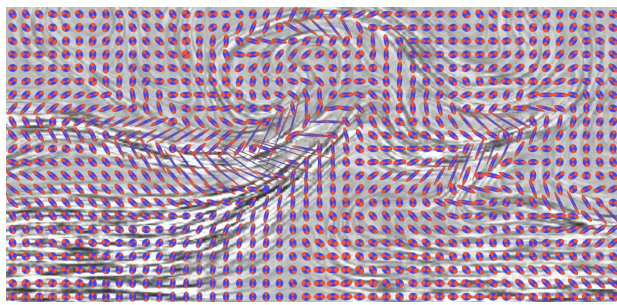
($s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|^{1/2}$), Fig. 10(c) better shows the directional patterns where the tensor norm is low. Colormapping the rate-of-deformation tensor trace with glyph halos highlights the regions of over-all stretching or compression, especially along the bottom edge of the domain.

Finally, Fig. 11 demonstrates how our new glyph performs traceless tensor visualization, in a side-by-side comparison to the dedicated traceless NLC tensor glyphs by Jankun-Kelly et al. [25]. Traceless tensors form a plane in eigenvalue space, and we are visualizing samples from a square within this plane, centered around the zero tensor (cf. Fig. 4(e)). Unlike the traceless glyph, which maps tensor norm to glyph *sharpness*, our glyph expresses norm by its overall *scale* $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|$. Consequently, the traceless glyph requires pre-specification of maximum eigenvalue magnitudes (which are mapped to perfect sharpness), while our glyph can be used without such prior information. Another notable difference is that limiting their glyph to traceless tensors allows Jankun-Kelly et al. to make use of parts of the superquadric shape space – including cylinders and boxes – that our approach sets aside for positive- or negative-definite tensors.

6 CONCLUSION

Visualization research has made significant progress in visualizing second-order tensor fields, but has mostly concentrated on the positive-definite case. Faced with indefinite tensors, a frequent strategy is to map them to positive-definite tensors prior to visualization [34, 22, 21, 52, 33]. Even when bijective mappings are used (so mathematically, no information is lost), such mappings still visually obscure the difference between positive and negative eigenvalues, which is a fundamental qualitative aspect in various applications.

Therefore, we propose an extension of a previous positive-definite tensor glyph [28] to the full space of symmetric second-order tensors. Our glyph emphasizes differences in eigenvalue sign in a way that, unlike the Reynolds glyph [18], prevents small eigenvalues from being occluded by larger ones. We also propose to use halos to ensure tensor glyph visibility even when one or more eigenvalues are near zero. Finally, we present a time- and memory-efficient implementa-



(a) Exponentially-scaled ellipses

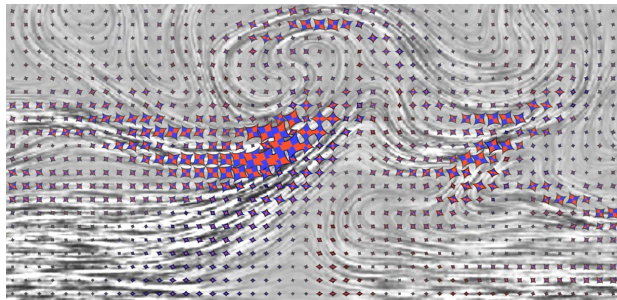
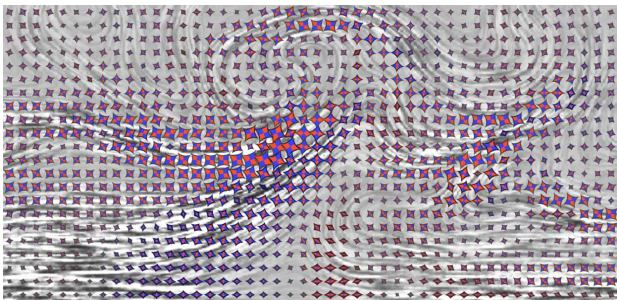
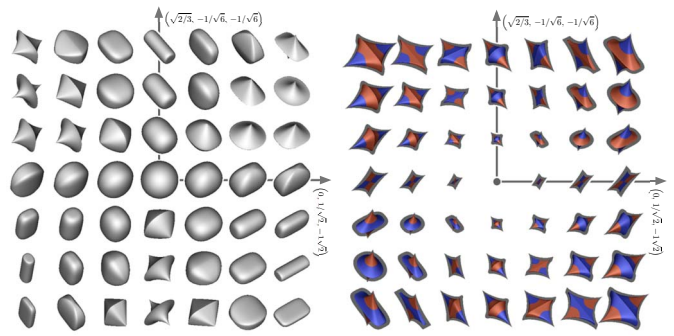
(b) Superquadric tensor glyphs; $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|$ (c) Superquadric tensor glyphs; $s(\|\mathbf{D}\|) \propto \|\mathbf{D}\|^{1/2}$

Fig. 10. Rate-of-deformation tensors visualized in a computational fluid dynamics simulation with exponentially-scaled ellipsoids (a) and our superquadric glyphs (b), (c).

tion, based on a small pre-computed palette of glyph shapes which are represented by a single triangle strip each. They are kept in GPU memory and instantiated as needed, combined with a simple and effective shader-based coloring to indicate the quadratic form.

Tensor glyph design is ultimately a creative process that balances aesthetic, computational, and mathematical considerations, and optionally some level of specialization to a particular visualization application. An interesting counterpart to our approach is the work of Keefe et al. [27] that fostered the design of multi-variate fluid flow visualizations through artistic collaborations, which started with visual media and then moved to computer graphics algorithms and the constraints they impose. Our initial work in glyph design began with hand-drawn experimental versions of Figs. 3, 4, and 5, but the mathematical guidelines of Section 2 were adopted from the outset as constructive constraints in the design process. Our glyph is designed to preserve symmetry, continuity, and invariance under scaling and eigenplane projection, while minimizing visual ambiguity. By adhering to these constraints, our glyph unambiguously reflects the mathematical properties of the underlying tensors, and produces useful visualizations in a range of applications.

Following on recent work by Jankun-Kelly et al. [24] for perceptually evaluating traceless tensor glyphs, future work may quantitatively investigate the effectiveness of our glyph for general symmetric tensors, perhaps in the context of particular application areas. Special-purpose glyphs may offer advantages for particular applications in-



(a) Traceless superquadric tensor glyphs proposed in [25] (b) Our symmetric tensor glyphs, applied to the same traceless plane

Fig. 11. When applied to traceless tensors, our glyph (b) uses a smaller part of superquadric shape space than a previous approach that specialized on the traceless case (a), but supplements it with color.

volving tensor fields, such as mapping glyph shape to important physically meaningful properties of the field. In contrast, our approach has aimed for generality across applications, and is most likely not a unique solution to the chosen mathematical constraints. We invite other researchers to explore new ways of satisfying some or all of the design constraints according to their own needs, perhaps starting with Fig. 1 as a framework for sketching new glyph shapes.

ACKNOWLEDGMENTS

We thank the reviewers for their constructive comments. The presentation of the glyph design principles benefited from discussions at the 2009 Dagstuhl Scientific Visualization Seminar 09251. Flow data courtesy of Wolfgang Kollmann, MAE Department of the University of California Davis. This work was supported by a fellowship within the Postdoc Program of the German Academic Exchange Service (DAAD). Our open-source implementation is described at <http://people.cs.uchicago.edu/~glk/sqd/>

REFERENCES

- [1] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 18(1):11–23, 1981.
- [2] P. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysics Journal*, 66(1):259–267, 1994.
- [3] W. Bengler and H.-C. Hege. Tensor splats. In *Visualization and Data Analysis (Proc. SPIE)*, volume 5295, pages 151–162, 2004.
- [4] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Computer Graphics (Proc. ACM SIGGRAPH)*, pages 263–270, 1993.
- [5] P. Danielsson, Q. Lin, and Q. Ye. Efficient detection of Second-Degree variations in 2D and 3D images. *Journal of Visual Communication and Image Representation*, 12(3):255–305, 2001.
- [6] W. C. de Leeuw and J. J. van Wijk. A probe for local flow field visualization. In *Proc. IEEE Visualization*, pages 39–45, 1993.
- [7] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
- [8] C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.
- [9] Z. Ding, J. C. Gore, and A. W. Anderson. Classification and quantification of neuronal fiber pathways using diffusion tensor MRI. *Magnetic Resonance in Medicine*, 49(4):716–721, 2003.
- [10] R. K. Dodd. A new approach to the visualization of tensor fields. *Graphical Models and Image Processing*, 60(4):286–303, 1998.
- [11] G. Elber. Line art illustrations of parametric and implicit forms. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):71–81, 1998.
- [12] C. Feddern, J. Weickert, and B. Burgeth. Level-set methods for tensor-valued images. In O. D. Faugeras and N. Paragios, editors, *Proc. Second IEEE Workshop on Geometric and Level Set Methods in Computer Vision*, pages 65–72, 2003.

- [13] L. Feng, I. Hotz, B. Hamann, and K. Joy. Anisotropic noise samples. *IEEE Trans. on Visualization and Computer Graphics*, 14(2):342–354, 2008.
- [14] A. Girshick, V. Interrante, S. Haker, and T. Lemoine. Line direction matters: An argument for the use of principal directions in 3D line drawings. In *Proc. 1st Intl. Symp. Non-Photorealistic Animation and Rendering*, pages 43–52. ACM, 2000.
- [15] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proc. IEEE Visualization*, pages 33–40, 1991.
- [16] R. Haber. Visualization techniques for engineering mechanics. *Computing Systems in Engineering*, 1(1):37–50, 1990.
- [17] H. Hagen, S. Hahmann, T. Schreiber, Y. Nakajima, B. Wordenweber, and P. Hollemann-Grundstedt. Surface interrogation algorithms. *IEEE Computer Graphics and Applications*, 12(5):53–60, 1992.
- [18] Y. M. A. Hashash, J. I.-C. Yao, and D. C. Wotring. Glyph and hyperstreamline representation of stress and strain tensors and material constitutive response. *Intl. Journal for Numerical and Analytical Methods in Geomechanics*, 27:603–626, 2003.
- [19] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *Computer Graphics (Proc. ACM SIGGRAPH)*, pages 517–526, 2000.
- [20] L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Trans. on Visualization and Computer Graphics*, 3(1):1–11, 1997.
- [21] I. Hotz, L. Feng, H. Hagen, B. Hamann, and K. Joy. Tensor field visualization using a metric interpretation. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, chapter 16, pages 269–280. Springer, 2006.
- [22] I. Hotz, L. Feng, H. Hagen, B. Hamann, K. Joy, and B. Jeremic. Physically based methods for tensor field visualization. In *Proc. IEEE Visualization*, pages 123–130, 2004.
- [23] E. Hsu. Generalized line integral convolution rendering of diffusion tensor fields. In *Proc. International Society of Magnetic Resonance in Medicine (ISMRM)*, page 790, 2001.
- [24] T. J. Jankun-Kelly, Y. S. Lanka, and J. E. Swan II. An evaluation of glyph perception for real symmetric traceless tensor properties. *Computer Graphics Forum (Special Issue on EuroVis 2010)*, 29(3):1133–1142, 2010.
- [25] T. J. Jankun-Kelly and K. Mehta. Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualization. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 12(5):1197–1204, 2006.
- [26] B. Jeremić, G. Scheuermann, J. Frey, Z. Yang, B. Hamann, K. I. Joy, and H. Hagen. Tensor visualizations in computational geomechanics. *Intl. Journal for Numerical and Analytical Methods in Geomechanics*, 26(10):925–944, 2002.
- [27] D. Keefe, D. Karelitz, E. Vote, and D. H. Laidlaw. Artistic collaboration in designing VR visualizations. *IEEE Computer Graphics and Applications*, 25(2):18–23, 2005.
- [28] G. Kindlmann. Superquadric tensor glyphs. In *Proc. EG/IEEE TCVG Symposium on Visualization*, pages 147–154, May 2004.
- [29] G. Kindlmann, X. Tricoche, and C.-F. Westin. Delineating white matter structure in diffusion tensor MRI with anisotropy creases. *Medical Image Analysis*, 11(5):492–502, 2007.
- [30] G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Trans. on Visualization and Computer Graphics*, 6(2):124–138, 2000.
- [31] G. Kindlmann and C.-F. Westin. Diffusion tensor visualization with glyph packing. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1329–1335, 2006.
- [32] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proc. IEEE Visualization 2003*, pages 513–520, 2003.
- [33] G. L. Kindlmann, R. S. J. Estépar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1415–1424, 2009.
- [34] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proc. IEEE Visualization*, pages 333–340, 1999.
- [35] 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. In *Proc. IEEE Visualization*, pages 127–134, 1998.
- [36] M. A. Livingston. Visualization of rotation fields. In *Proc. IEEE Visualization*, pages 491–494, 1997.
- [37] T. Luft, C. Colditz, and O. Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, 25(3):1206–1213, 2006.
- [38] J. E. Marsden and A. J. Tromba. *Vector Calculus*. W.H. Freeman and Company, New York, New York, 1996.
- [39] M. D. Meyer, P. Georgel, and R. T. Whitaker. Robust particle systems for curvature dependent sampling of implicit surfaces. In *Proc. Shape Modeling and Applications (SMI)*, pages 124–133, June 2005.
- [40] A. Neeman, B. Jeremic, and A. Pang. Visualizing tensor fields in geomechanics. In *Proc. IEEE Visualization*, pages 35–42, 2005.
- [41] E. Özarslan and T. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50(5):955–965, 2003.
- [42] S. Pajevic and C. Pierpaoli. Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data: Application to white matter fiber tract mapping in the human brain. *Magnetic Resonance in Medicine*, 42(3):526–540, 1999.
- [43] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramée, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.
- [44] T. Ropinski, J. Meyer-Spradow, M. Specht, K. H. Hinrichs, and B. Preim. Surface glyphs for visualizing multimodal volume data. In *Proc. Vision, Modeling, and Visualization*, pages 3–12, 2007.
- [45] R. J. Rost. *OpenGL shading manual*. Addison-Wesley, 2nd edition, 2006.
- [46] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object Oriented Approach to Graphics*, chapter 6. Kitware, 2003.
- [47] W. J. Schroeder and K. M. Martin. *The Visualization Handbook*, chapter 1: Overview of Visualization, pages 3–38. Academic Press, 2004.
- [48] T. Schultz and G. Kindlmann. A maximum enhancing higher-order tensor glyph. *Computer Graphics Forum (Proc. EuroVis)*, 29(3):1143–1152, 2010.
- [49] T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Trans. on Visualization and Computer Graphics*, 16(1):109–119, 2010.
- [50] C. D. Shaw, D. S. Ebert, J. M. Kukla, A. Zwa, I. Soboroff, and D. A. Roberts. Data visualization using automatic, perceptually-motivated shapes. In *Visual Data Exploration and Analysis (Proc. SPIE)*, volume 3298, pages 208–213, 1998.
- [51] A. Sigfridsson, T. Ebbens, E. Heiberg, and L. Wigström. Tensor field visualisation using adaptive filtering of noise fields combined with glyph rendering. In *Proc. IEEE Visualization*, pages 371–378, 2002.
- [52] V. Slavin, R. Pelcovits, G. Lorient, A. Callan-Jones, and D. Laidlaw. Techniques for the visualization of topological defect behavior in nematic liquid crystals. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1323–1328, 2006.
- [53] M. Tarini, P. Cignoni, and C. Montani. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006.
- [54] H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *Proc. IEEE Visualization*, pages 225–232, 2003.
- [55] W.-S. Tong, C.-K. Tang, P. Mordohai, and G. Medioni. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3D. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):594–611, 2004.
- [56] A. Vilanova, S. Zhang, G. Kindlmann, and D. H. Laidlaw. An introduction to visualization of diffusion tensor imaging and its applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 121–153. Springer, 2006.
- [57] J. Weickert and H. Hagen, editors. *Visualization and Processing of Tensor Fields*. Springer, 2006.
- [58] E. W. Weisstein. *CRC Concise Encyclopedia of Mathematics*, page 1894. CRC Press, 2003.
- [59] C.-F. Westin, S. E. Maier, B. Khidir, P. Everett, F. A. Jolesz, and R. Kikinis. Image processing for diffusion tensor magnetic resonance imaging. In *Proceedings MICCAI*, volume 1679 of LNCS, pages 441–452, 1999.
- [60] M. R. Wiegell, H. B. W. Larsson, and V. J. Wedeen. Fiber crossing in human brain depicted with diffusion tensor MR imaging. *Radiology*, 217(3):897–903, 2000.
- [61] X. Zheng and A. Pang. HyperLIC. In *Proc. IEEE Visualization*, pages 249–256, 2003.