Topological Lines in 3D Tensor Fields and Discriminant Hessian Factorization

Xiaoqiang Zheng, *Student Member*, *IEEE*, Beresford N. Parlett, and Alex Pang, *Senior Member*, *IEEE*

Abstract—This paper addresses several issues related to topological analysis of 3D second order symmetric tensor fields. First, we show that the degenerate features in such data sets form stable topological lines rather than points, as previously thought. Second, the paper presents two different methods for extracting these features by identifying the individual points on these lines and connecting them. Third, this paper proposes an analytical form of obtaining tangents at the degenerate points along these topological lines. The tangents are derived from a Hessian factorization technique on the tensor discriminant and leads to a fast and stable solution. Together, these three advances allow us to extract the backbone topological lines that form the basis for topological analysis of tensor fields.

Index Terms—Hyperstreamlines, real symmetric tensors, degenerate tensors, tensor topology.

1 Introduction

The main motivation and goal of this work is to develop a simple yet powerful representation of 3D real symmetric tensor fields. Topology-based methods can yield simplified yet effective representation in many visualization fields [2], [17], [22]. The topological structures make it simple for users to understand the underlying data fields yet they are sensitive enough to capture important features. Early work on using the topology-based method to visualize tensor fields by Hesselink et al. [7], [9] lays important background for this research project. They define the tensor topology based on degenerate features, discuss its nature for 2D cases in great detail, and provide useful information for 3D cases. However, we find this early work insufficient in studying 3D tensor topology. Not only is the dimension of the degenerate features unknown, but it is also unclear how to numerically extract the topological structures. In their previous work, Hesselink et al. mentioned that the dimension of the degenerate features can be points, lines, surfaces, or subvolumes. This claim is essentially true, but it does not point out the dimension of degenerate features in typical nondegenerate 3D tensor data. By analogy, although the critical features in 3D vector fields can be lines, surfaces, or even subvolumes, we know they are mainly points in a typical nondegenerate 3D vector data. This knowledge is the foundation of the study of topological structure in vector field visualization. All the subsequent study on separatrices and other topological features is based on the extraction of critical *points*. On the other hand, prior to our findings, there were no topological results on 3D real symmetric tensor fields indicating that the degenerate features form *lines*.

During our research on 3D tensor topology, we confirmed that the topological structures in 3D real symmetric tensor fields form feature lines. This can be verified by an early theorem pointed out by Wigner and von Neumann stating that the real symmetric degenerate matrices form a variety of codimension two [15]. This discovery is important in that it tells us that future studies on the topology-based method for 3D nondegenerate real symmetric tensor fields should be based on degenerate *lines*. We can capture and preserve important features when studying the underlying tensor fields by looking at these extracted topological structural lines together with their separatrix surfaces.

The basic strategy for finding these degenerate lines is to first find the degenerate tensors on a 2D patch, i.e., on the face of a hexahedral cell. Next, the feature lines can be obtained by connecting the appropriate set of points. Traditionally, the degenerate features in 3D tensor fields are defined as tensors whose cubic discriminant is equal to zero. Finding roots of discriminants in a stable manner proves to be a challenging task because of their high-orderedness and singularity. To address these shortcomings, we introduce two alternative formulations to extract 3D degenerate tensors. The first formulation decomposes the cubic discriminant into the sum of the squares of seven cubic polynomials referred to as *discriminant constraint functions* [27]. Through this decomposition, the tensors whose discriminant equals zero are equivalent to the tensor whose individual tensor constraint functions are all equal to zero simultaneously. This formulation eliminates the high-orderedness and singularity problems encountered in existing practice and, thus, makes developing stable numerical algorithm to extract degenerate feature lines possible. The second formulation rewrites the degenerate tensors using a geometric approach with four implicit parameters and solves the singularity with a root finding scheme (Section 5).

To ensure that all degenerate tensors are found, a cell and its faces may be subdivided a number of times. Once the degenerate points are found, the feature lines can be

- *X. Zheng and A. Pang are with the Computer Science Department, University of California, Santa Cruz, CA 95064. E-mail: {zhengzx, pang}@cse.ucsc.edu.*
- *B.N. Parlett is with the Department of Mathematics and Graduate School, University of California, Berkeley, CA 94720. E-mail: parlett@math.berkeley.edu.*

Manuscript received 15 Sept. 2004; revised 12 Dec. 2004; accepted 27 Dec. 2004; published online 10 May 2005.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0105-0904.

formed in a relatively straightforward manner by connecting points along lines with minimum angular change. This method of forming feature lines is not mathematically satisfying. Furthermore, the subdivision process introduces significant computational load to the extraction. Hence, in this paper, we also propose a technique using the tangents at the degenerate tensors to help find the topological lines by tracing from the extracted degenerate points. The major contribution in this paper is an efficient analytical method for finding these tangents. It is based on a fundamental theorem on Hessian factorization of the discriminant that allows us to find an analytical expression for the tangent. This approach reduces the false negatives in the number of extracted degeneracies and uses fewer computational resources by tracing an entire feature line from a single extracted point. The method also accurately resolves the connectivity problem when more than two singularities are present in one cell and greatly improves the performance of the extraction. Furthermore, the point connection step is independent of the underlying grid in which the tensor is defined. Although its derivation and reasoning are fairly complicated, the conclusion of the technique is very intuitive and can be generalized to real symmetric second order tensors of any dimension.

In this paper, we describe the key steps in finding the 3D tensor topology. For each step, we show alternative solutions. Since each solution still has the potential to improve, we mainly focus on and discuss their properties instead of comparing and recommending the best one.

The rest of this paper is organized as follows: Section 2 reviews some important facts used in tensor analyses, Section 3 discusses the relevant previous work in tensor field analyses and visualization, Section 4 shows that 3D degenerate tensors form lines in general, Section 5 discusses several strategies to extract 3D degenerate tensors on a 2D patch, Section 6 introduces the discriminant Hessian factorization at degenerate tensors, Section 7 goes over methods to form feature lines from the extracted degenerate points, and Section 8 presents results for both synthetic and practical tensor data sets.

2 Tensor Analysis

Tensor fields, especially second-order tensor fields, are useful in many medical, mechanical, and physical applications such as: fluid dynamics, meteorology, molecular dynamics, biology, geophysics, astrophysics, mechanics, material science, and earth science. Effective tensor visualization methods can enhance research in a wide variety of fields. However, developing an effective algorithm can be difficult because of the large amount of information contained in 3D tensor fields: There are nine independent components in each tensor and six for a symmetric tensor. Users in many research fields are especially interested in real symmetric tensors. In some applications, the data itself is inherently symmetric. In other cases, symmetric tensor data can be obtained through various decomposition techniques. In this paper, the name *tensors* implicitly stands for *second order symmetric tensors* unless mentioned otherwise.

2.1 Degenerate Tensors and Discriminants

Any real symmetric tensor can be decomposed into three orthogonal eigenvectors, each of which has an eigenvalue associated with it. The eigenvectors are labeled as major, medium, and minor, according to the relative magnitudes of their eigenvalues. Using any of these three eigenvector fields, we can define hyperstreamlines [6]. In nondegenerate cases, the hyperstreamlines do not cross each other. The degenerate features are thus defined as those where the hyperstreamlines could cross each other. Hesselink et al. show that the only degenerate features are those having at least two equal eigenvalues [9]. Fortunately, we do not need to conduct the eigendecomposition to find the degenerate points. A tensor has two (or three) equal eigenvalues if and only if its discriminant equals zero. The discriminant D_3 of a 3D tensor T with eigenvalues λ_1, λ_2, and λ_3 and tensor components T_{ij} is defined as:

$$T = \begin{pmatrix} T_{00} & T_{01} & T_{02} \\ T_{01} & T_{11} & T_{12} \\ T_{02} & T_{12} & T_{22} \end{pmatrix}, \tag{1}$$

$$D_3(T) = (\lambda_1 - \lambda_2)^2(\lambda_2 - \lambda_3)^2(\lambda_3 - \lambda_1)^2. \tag{2}$$

This can be reformulated into a form that does not require eigendecomposition yet still explicitly determines eigenvalues as follows:

$$P = T_{00} + T_{11} + T_{22}, \tag{3}$$

$$Q = \begin{vmatrix} T_{00} & T_{01} \\ T_{01} & T_{11} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{vmatrix} + \begin{vmatrix} T_{22} & T_{02} \\ T_{02} & T_{00} \end{vmatrix}, \tag{4}$$

$$R = \begin{vmatrix} T_{00} & T_{01} & T_{02} \\ T_{01} & T_{11} & T_{12} \\ T_{02} & T_{12} & T_{22} \end{vmatrix}, \tag{5}$$

$$D_3(T) = Q^2P^2 - 4RP^3 - 4Q^3 + 18PQR - 27R^2. \tag{6}$$

From (2), we can easily see that a discriminant is 1) always nonnegative and 2) equal to zero if and only if at least two of the eigenvalues are equal. Further, it is ideal for computation and numerical purposes because, although it is defined on eigenvalues, we do not need to carry out an expensive eigendecomposition. Instead, we only need to compute (6), which is a polynomial of order six, to get the discriminant.

The justification to define tensors with two or more equal eigenvalues as a degenerate feature is simple. In a vector field, the streamline integration is ambiguous at a point with zero velocity. In a tensor field, the hyperstreamline integration is also ambiguous at a point with two equal eigenvalues because any linear combination of the two eigenvectors is another eigenvector. Since the degenerate tensors are the only places where hyperstreamlines can cross each other, they play an important role in 3D tensor field topological analysis, which divides the space into smaller subspaces wherein local hyperstreamlines exhibit similar patterns. In general, the degenerate tensors can be interpreted in a fashion similar to the critical points in vector field visualization.

2.2 Dimensional Analysis and Transversality

For features A defined on a D-dimensional data that form a subspace of dimension F, *codimension* is defined as the

difference between the dimension of the data and the feature subspace: $\mathrm{codim}(A) = D - F$. Codimension can also be interpreted as the number of independent constraints that reduces the feature dimension. Two smooth submanifolds, A and B, of a smooth manifold M are said to intersect *transversely* if, for any point $x \in A \bigcap B$, we have $tA_x + tB_x = tM_x$, where tA_x denotes the tangent space of A_x. In this case, A and B intersect properly in the sense that $A \bigcap B$ is a submanifold of M, and $\mathrm{codim}(A \cap B) = \mathrm{codim}(A) + \mathrm{codim}(B)$ [1]. Transversality is a sufficient condition for an intersection to be stable after a perturbation. This conclusion is important in analyzing the dimension of degenerate features in 3D tensor fields.

3 Previous Work

Early tensor visualization techniques relied on the tensor ellipsoid, which is a spherical glyph deformed according to the eigenvalues of the tensors. Variations of the basic tensor ellipsoid include drawing eigenvalue scaled axes for the eigenvectors, Haber's disk, and rod glyphs [8], flow probe [5], Kirby et al. glyphs using brush strokes [13]. More recently, Kindlmann proposed new superquadric tensor glyphs to visualize tensors as a combination of spherical, planar, and linear tensors [11]. With few exceptions, particularly for the case of 3D tensor fields, glyphs have been used in a sparing manner because of the clutter and occlusion they produce. Hence, they provide a discrete, rather than a continuous view of the tensor field. To address this problem, tensor splats were introduced by Bhalerao and Westin [3] to provide a global continuous view of the tensor field. Using their approach—a barycentric mapping of linear, planar, and spherical tensors—different parts of the tensor volume can be highlighted. An alternative formulation, also called tensor splats, but with a different way of displaying directional information, was presented by Benger and Hege [21].

A hyperstreamline is basically a streamline defined over an eigenvector field [6]. Typically, the major eigenvector field is used for integrating the hyperstreamline, while the two other eigenvector fields provide local information along the length of the major hyperstreamline and are mapped to its cross section. One of the weaknesses of hyperstreamlines is ambiguity in places where the tensors are degenerate, i.e., where the eigenvalues are nearly equal. In these areas, a sudden change in the direction of the hyperstreamline may arise. Note that this is a common problem with integration algorithms, e.g., fiber tracking algorithms in DT-MRI. To address this problem, tensorlines were introduced by Weinstein et al. [23]. Ambiguities are resolved by taking the anisotropy of the local tensor into account as well as information about orientation of nearby features. This allows the tensorlines to proceed in a relatively smooth path, even in the face of isotropic regions or noise in the data set.

Topology-based tensor visualization techniques represent the tensor fields in a simple yet powerful way. The critical features are extracted to present a simplified version of the underlying data field. They are defined as degenerate tensors where the eigenvalues are identical and are the only places where the two associated hyperstreamlines can intersect each other. In 2D tensor fields, there is only one way to obtain a degenerate point: The two eigenvalues must be equal. Hesselink and Delmarcelle used this concept in 2D and discussed the nature of the degenerate points (wedges and trisectors) in great detail. However, it is less successful in 3D, in part because there are two types of degenerate points in 3D: double degenerate points, where two of the three eigenvalues are equal, and triple, where all three eigenvalues are identical. Furthermore, the double degenerate points may be distinguished by whether the minor and medium eigenvalues are equal or the medium and major eigenvalues are equal. This distinction is important in some applications. Hesselink et al.'s early work does not fully explore the properties of the double degenerate features and, instead, focuses on the triple degenerate tensors, whose properties are closer to their counterparts in 2D. They hint that the triple degenerate points (for the double point load data) are connected by a locus of double degenerate points [9]. The paper fails to point out that the dimension of the stable double degenerate features is, in fact, lines in most of the typical nondegenerate tensor fields. Hence, it did not attempt to find a stable numerical method to extract these feature lines in 3D.

Although triple degenerate features are useful, they are extremely rare and unstable. For triple degenerate features, there must be a scalar times an identity matrix. That means they form a subspace, A, of dimension one, i.e., codimension five in the 6D symmetric tensor space. Further, the 3D tensor data form a 3D subspace, B, in the same tensor space. The feature in the data is the intersection of these two subspaces, $\mathrm{codim}(A \cap B) = \mathrm{codim}(A) + \mathrm{codim}(B) = 5 + 3 = 8$. In other words, the dimension of the feature in a 3D data is $6 - 8 = -2$. A dimension that is less than zero means the feature is unstable. In summary, not only is it extremely rare in real data, but its very existence will also be easily dissolved by small errors introduced by numerical and interpolation methods. This property dramatically limits the usefulness of triple degenerate tensors in practical contexts. Even in time-varying data, which has a dimension of four, the codimension of the feature is $\mathrm{codim}(A \cap B) = \mathrm{codim}(A) + \mathrm{codim}(B) = 5 + 2 = 7$, which leads to a dimension $6 - 7 = -1$, and is still unstable. Actually, in our experiments with many real time-varying stress data sets, we have not found any triple degenerate points. The only data set that contains triple degenerate points is from a synthetic data set known as the Boussinesq double point load stress tensor.

In complex 2D tensor fields, the extracted topology may also be very complex. Tricoche and Scheuermann [20] proposed algorithms to simplify 2D tensor topology as well as track them in time-varying 2D tensor fields [19].

Anisotropy in tensor fields are important features in some fields such as diffusion tensor MRI. In 3D, tensors are classified as being linear or anisotropic where there is a predominant eigenvalue and two other smaller eigenvalues, planar where there are two roughly equal eigenvalues and one smaller one, and spherical or isotropic where there are three roughly identical eigenvalues. Kindlmann and Weinstein [12] use barycentric coordinates to map these properties to color and opacity in volume rendering tensor fields. An alternative approach, called HyperLIC, was proposed by Zheng and Pang [25] to highlight anisotropy using textures. Linear tensors are represented by highly correlated, high contrast textures, while isotropic tensors end up as blurry textures with no preferred orientation.

Using the physical analogy of bending steel beams under load, Boring and Pang [4] used deformation to visualize the effects of tensor fields. Idealized objects such as lines, surfaces, and subvolumes are deformed under tensor transformations. This was further improved to provide a globally consistent deformation based on a collection of local deformations [24]. Calculations were carried out using a system of springs.

Extending the idea of deformation to optics, Zheng and Pang [26] added three alternative ways of visualizing tensors: 1) First, light rays were traced through a tensor volume and bent according to the local tensor properties that they encounter. The bent rays show divergent or convergent regions in the tensor field. 2) Second, the exit points of the rays were collected, as in caustic ray tracing. Different wavelengths were simulated and color separation on the resulting caustic image provides a dense visualization of divergence and convergence from a given viewpoint. 3) Third, the tensor field was treated as a lens that distorts an image. Studying the distortion of a known image, such as a checkerboard pattern, revealed compressive and tensile regions in the tensor field.

4 Dimensionality and Types of Degenerate Tensors

4.1 Dimensionality of Tensor Features

Before we can extract the critical features from 3D tensor fields, we need to know what kind of features we are looking for. Algorithms to locate points, lines, surfaces, and volumes employ very different strategies. As mentioned earlier, we found that, for most nondegenerate 3D tensors, the dimensionality of the critical feature is *one* and, hence, they form feature lines. This can be shown using the theorem by von Neumann and Wigner:

Theorem 4.1. *Real symmetric degenerate matrices form a variety of codimension two [15].*

Since codimension can be interpreted as the number of constraints that reduces the feature dimensionality, this theorem can be interpreted as follows: The first constraint of codimension two is from the definition of degenerate tensors that two eigenvalues are equal. The second constraint is actually a "redundancy" that further reduces the degenerate feature dimension implicitly—when two eigenvalues are equal, the associated eigenvectors are indeterminate up to one degree of freedom since any orthogonal linear combination of the eigenvectors yields another valid pair. For real symmetric 3D tensors, we have six degrees of freedom. Hence, it also follows from this theorem that these tensors form a variety of dimension four. An approach to parameterize the 3D degenerate tensors using four parameters is introduced in [28] and will be discussed in Section 5.

Three-dimensional real symmetric tensors have six independent components. Therefore, they form a tensor space of dimension six. A double degenerate tensor where two eigenvalues are equal can be uniquely specified using four parameters. In other words, double degenerate tensors form a subspace A of dimension four in 6D tensor space. In a typical nondegenerate setting, tensor fields defined in a 3D space usually form a subspace B of dimension three in the same 6D tensor space. The degenerate tensors are then the intersection of these two subspaces. Using transversality, we have $\mathrm{codim}(A \cap B) = \mathrm{codim}(A) + \mathrm{codim}(B) = 2 + 3 = 5$. That is, this intersection usually has a dimension one, i.e., double degenerate tensors form feature lines. Using the same line of reasoning, one can show that degenerate tensors are isolated points in most cases if the data is specified in a 2D space.

While the main features are lines in 3D, it is still possible to have features that are points, surfaces, or subvolumes, but those types of features would be considered unstable and do not persist. Those types of unstable features are also less common in most 3D tensor fields. As such, we focus our tensor feature extraction to extract feature lines rather than surfaces or subvolumes. We still need to extract points as these form the basis for the feature lines. Because of this design criterion, features that are surfaces (such as those found in the single point load data) or subvolumes may not be detected as readily as feature lines. This limitation is not insurmountable, but is, rather, based on the effective use of limited resources in finding features that are neither as common nor as stable.

4.2 Categorization of Degenerate Lines

There are two cases where hyperstreamlines cross each other in double degenerate tensors. The first case (type P for planar) is when the major and medium eigenvalues are identical and the second case (type L for linear) is when the medium and minor eigenvalues are identical. We define a quantity, K, that measures the *eigen difference* of a tensor, T, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$:

$$K = 2\lambda_2 - (\lambda_1 + \lambda_3). \tag{7}$$

The eigen difference K measures whether the tensor at a point is closer to type P or type L. It is easy to show that K returns a positive value for a type P degenerate tensor and returns a negative value for a type L degenerate tensor. When K equals zero, the three eigenvalue are equal and the tensor is triple degenerate. The eigen difference is also used to pseudocolor the feature lines (see Figs. 3, 7, and 8). Warm colors are associated with type P and cool colors are associated with type L degenerate points. The closer the color is to pure green, the closer the tensor is to triple degeneracy. Triple degenerate points are the only locations where a type P and type L feature line can cross. This fact, together with the color mapping for K, provides a strong visual clue for finding triple degenerate points, even though these are not explicitly calculated.

4.3 Overview of Extracting Degenerate Lines

Since most numerical algorithms are designed to capture points, we develop the degenerate line extraction algorithm in two stages: In the first stage, we locate 3D degenerate tensors on 2D patches; in the second stage, we connect them to form lines.

Several strategies or solutions will be introduced for each stage. We discuss the direct discriminant minimization, constraint functions, and geometric approach for the point extraction stage. For the line connection stage, we introduce minimum angle connection, feature line tracing, and the prediction-correction algorithm based on an analytical formula of the degenerate line tangents. This formula is derived from the discriminant Hessian factorization on degenerate tensors discussed in Section 6.

5 Extract Degenerate Points on 2D Patches

5.1 Discriminants

To find the critical degenerate tensors, we need to locate those tensors whose discriminants are zero. Although (6) provides an elegant representation for evaluating the discriminant without having to perform eigendecomposition, it is not very suitable for finding roots. In (6), the discriminant of a real symmetric tensor is a polynomial of order six. Since it is always nonnegative, the degenerate tensor also happens to be its minimum. A good method widely used to find the roots of an equation is to detect the change of signs and then to recursively bisect the domain of interest. However, since the degenerate feature is itself a minimum, there is no change of sign at all. Relying on the gradients is also dangerous because the gradients are notoriously unstable unless they are very close to the feature. Due to this high-orderedness and singularity, directly finding the roots of a cubic discriminant in a stable manner is very difficult. Instead, we look for another representation of the discriminant.

5.2 Constraint Functions

In our investigation so far, we found that, while Hilbert [10] pointed out that not all nonnegative polynomials can be broken down into the sum of squares of polynomials, the cubic discriminant can be written as the sum of the squares of *seven* polynomials. We also learned that not only can the discriminant of a second-order tensor of any dimension be expressed as the sum of squares [14], but our solution to the 3D case using *seven* equations is optimal in the number of equations [16]. Therefore, the definition of degenerate tensors can also be expressed as the tensors where the seven *discriminant constraint functions* all equal zero at the same time. We use these seven cubic equations to extract the feature lines from 3D tensor fields. The seven discriminant constraint functions are:

$$\begin{aligned}
f_x(T) &= T_{00}(T_{11}^2 - T_{22}^2) + T_{00}(T_{01}^2 - T_{02}^2) + T_{11}(T_{22}^2 - T_{00}^2) \\
&\quad + T_{11}(T_{12}^2 - T_{01}^2) + T_{22}(T_{00}^2 - T_{11}^2) + T_{22}(T_{02}^2 - T_{12}^2) \\
f_{y1}(T) &= T_{12}(2(T_{12}^2 - T_{00}^2) - (T_{02}^2 + T_{01}^2) + 2(T_{11}T_{00} + T_{22}T_{00} \\
&\quad - T_{11}T_{22})) + T_{01}T_{02}(2T_{00} - T_{22} - T_{11}) \\
f_{y2}(T) &= T_{02}(2(T_{02}^2 - T_{11}^2) - (T_{01}^2 + T_{12}^2) + 2(T_{22}T_{11} + T_{00}T_{11} \\
&\quad - T_{22}T_{00})) + T_{12}T_{01}(2T_{11} - T_{00} - T_{22}) \\
f_{y3}(T) &= T_{01}(2(T_{01}^2 - T_{22}^2) - (T_{12}^2 + T_{02}^2) + 2(T_{00}T_{22} + T_{11}T_{22} \\
&\quad - T_{00}T_{11})) + T_{02}T_{12}(2T_{22} - T_{11} - T_{00}) \\
f_{z1}(T) &= T_{12}(T_{02}^2 - T_{01}^2) + T_{01}T_{02}(T_{11} - T_{22}) \\
f_{z2}(T) &= T_{02}(T_{01}^2 - T_{12}^2) + T_{12}T_{01}(T_{22} - T_{00}) \\
f_{z3}(T) &= T_{01}(T_{12}^2 - T_{02}^2) + T_{02}T_{12}(T_{00} - T_{11}),
\end{aligned}$$

$$\begin{aligned}
D_3(T) = f_x(T)^2 + f_{y1}(T)^2 + f_{y2}(T)^2 + f_{y3}(T)^2 + \\
15f_{z1}(T)^2 + 15f_{z2}(T)^2 + 15f_{z3}(T)^2.
\end{aligned} \tag{8}$$

A tensor is degenerate if and only if all of its seven discriminant constraint functions are zero. We employ this condition to extract the critical features in 3D tensor fields. Its first advantage is that the constraint functions are only cubic polynomials, instead of a polynomial of order of six, which tend to oscillate more. This property leads to a more stable and accurate numerical algorithm. In addition, the requirement that all seven constraint functions be zero at the same time depends on the tensor value only and not on the gradient calculated from adjacent tensors. Hence, the algorithm yields a more accurate result than algorithms that rely on finding critical points where the gradients of the discriminants equal zero. Its second advantage is that the constraint functions can be both positive or negative, as opposed to always being nonnegative. This property allows us to perform a fast and inexpensive check for the existence of features. Finally, the reformulation does not require eigendecomposition.

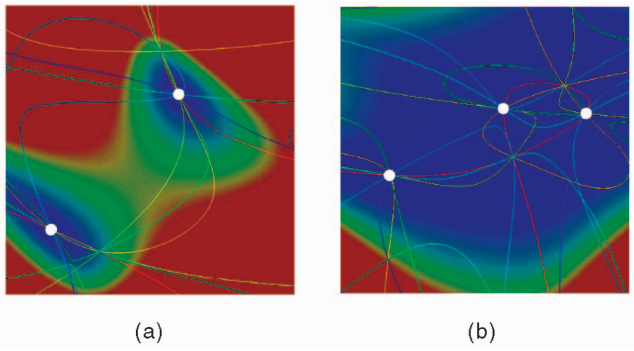


Fig. 1. White dots are degenerate points indicating places where all seven constraint functions are zero. Each colored curve corresponds to a constraint function being equal to zero. Places where multiple curves intersect are where multiple constraint functions are satisfied simultaneously. The background is pseudocolored by the discriminant functions. The data is a 2D slice of a randomly generated 3D tensor field. (a) Slice with two degenerate points. (b) Slice with three degenerate points.

Root Finding. In order to extract smooth and continuous feature lines in 3D tensor fields, we look at each of the six faces of every hexahedral cell. For each face, we extract the intersection point(s) of the feature lines. These points are then connected to generate a continuous feature line.

We know that the degenerate 3D tensors on a 2D slice are mostly points. The only exception is if the feature line lies exactly *on* the face. But, even for that case, that feature line will intersect an adjacent noncoplanar face on the edge or, possibly, corner vertex. To find the feature (intersection) points that satisfy all seven constraints simultaneously, we employ a modified version of the Newton-Raphson algorithm to solve the overspecified system of equations.

Assume the tensor field is $T(X)$. For the feature points X^*, we have $\overrightarrow{CF}(X^*) = CF_i(X^*) = 0$, for $i = 1, \ldots, 7$, where $\overrightarrow{CF}(X)$ is an assembly of the seven constraint functions into one vector function. Using the modified Newton-Raphson method and an initial guess of X_n, we have the following updating formula:

$$X_{n+1} = X_n - \left(\frac{\partial \overrightarrow{CF}}{\partial X}^T \cdot \frac{\partial \overrightarrow{CF}}{\partial X}\right)^{-1} \left(\frac{\partial \overrightarrow{CF}}{\partial X}^T \cdot \overrightarrow{CF}\right)\Bigg|_{X=X_n} \tag{9}$$

$$\frac{\partial \overrightarrow{CF}}{\partial X} = \frac{\partial \overrightarrow{CF}}{\partial T} \cdot \frac{\partial T}{\partial X}. \tag{10}$$

Note that we calculate $\frac{\partial \overrightarrow{CF}}{\partial X}$ from the chain rule using $\frac{\partial \overrightarrow{CF}}{\partial T}$ and $\frac{\partial T}{\partial X}$ rather than from the interpolated values of $\overrightarrow{CF}$ on

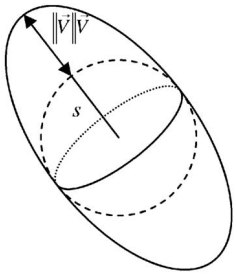


Fig. 2. Relationship of s and V and the degenerate tensor glyph. s is the radius of the spherical tensor.

the grid. $\frac{\partial \overrightarrow{CF}}{\partial T}$ is calculated from the formula of the tensor constraints and $\frac{\partial T}{\partial X}$ is from the interpolated tensor values. We used both the bilinear and bicubic natural spline interpolations.

Using the center of each cell face as the initial guess for an intersection point, we find that this method converges to the actual intersection point within five iterations for most nondegenerate cases with a precision up to 10^{-9} and it rarely misses a feature point if it exists. Additional points are obtained by subdividing the cell face. This modified Newton-Raphson method on constraint functions is superior in speed, accuracy, and precision compared to other methods developed directly based on the cubic discriminants. For comparison purposes, we also implemented an algorithm based on cubic discriminant that searched for its minimum using conjugate gradient methods. Not only is it about 50 times slower, using any precision less than 10^{-6} will yield a false negative rate of over 50 percent.

5.3 Geometric Approach

Since we want to extract the degenerate tensors in a root-finding framework, it is desirable to have a well-defined system of equations with an equal number of equations as there are unknowns. However, neither the discriminant nor the constraint functions satisfy this condition. An equation based on the discriminant is underspecified since there is only one equation for two unknowns. An equation based on constraint functions is overspecified because there are seven equations with two unknowns. The formulation on constraint functions is numerically better than its discriminant counterpart because an overspecified system is easier to solve using the modified Newton-Raphson algorithm and achieves high convergence rates and precision. In this section, we present another extraction algorithm based on the geometric properties of 3D tensors that meets the desired criterion of a well-defined system.

Theorem 5.1. *A tensor T is degenerate if and only if it can be written as the sum of a spherical tensor and a linear tensor (see Fig. 2).*

The sufficiency of this theorem is easy to prove. To show its necessity, we simply subtract the duplicate eigenvalues from the diagonal components of the tensor. It is easy to show that the remaining tensor has two duplicate zero eigenvalues. In other words, the rank of the remaining tensor is at most rank one, i.e., linear. Depending on the sign of the other eigenvalue, a linear real symmetric tensor can always be written as the product of a vector, its transpose, and an extra sign. This gives us a simple way to write a degenerate tensor:

$$T = sI \pm V \cdot V^T, \tag{11}$$

where s is a scalar, I is a 3×3 identity matrix, and V is a 3×1 vector. An advantage of this formula is that it can distinguish between type P and type L double degenerate points: T is type P with equal major and medium eigenvalues if the minus sign holds and T is type L with equal minor and medium eigenvalues if the plus sign holds. In applications where the users are only interested in the major hyperstreamline topology, users only need to retain the minus sign since the major hyperstreamlines are only degenerate at type P features. The three eigenvalues are: $\lambda_1 = \lambda_2 = s$ and $\lambda_3 = s \pm \|V\|^2$. One of the eigenvectors is $e_3 = V/\|V\|$ and the other two eigenvectors are any two orthogonal vectors that are also perpendicular to e_3. Besides its simplicity, this equation also clearly states that all 3D degenerate tensors form a four-parameter family: (s, V). A typical and nondegenerate 3D real symmetric tensor on a 2D patch parameterized by (x, y) is:

$$T(x, y) = sI \pm V \cdot V^T. \tag{12}$$

Since there are six independent components in real symmetric tensors, this system of equations has six equations and six unknowns. Therefore, we expect that it has stable and isolated solutions. If we assume that the tensor patch is obtained through bilinear interpolation, then each equation is quadratic. To solve this formulation, we can employ any standard numerical method to solve a well-defined system of equations such as the Newton-Raphson algorithm or one of its variants.

For the initial guess in the Newton-Raphson method, we use the center of the patch, (x_0, y_0), in place of the position parameters (x, y). Suppose the tensor at (x_0, y_0) is T_0 and suppose that its eigenvalues are $(\lambda_1 \le \lambda_2 \le \lambda_3)$ and its normalized eigenvectors are (e_1, e_2, e_3), respectively. Without loss of generality, we also assume that we are extracting type P degenerate features. The algorithm for extracting type L degenerate features is similar in form. To obtain the initial estimates of the other four parameters (s, V), we use the following heuristic

$$s_0 = \frac{\lambda_2 + \lambda_3}{2}, \tag{13}$$

$$V_0 = \sqrt{s_0 - \lambda_1} \cdot e_1. \tag{14}$$

Using s_0 and V_0 for the initial guess, we iteratively update the six parameters using the Newton-Raphson method until the solution converges. Since each equation is a simple quadratic equation, taking derivatives is trivial. When the algorithm converges, not only do we have the location of the degenerate feature, but we also get the eigenvalues and eigenvectors of the tensor values at that point from s and V. Notwithstanding its simplicity, the disadvantage of this algorithm is also obvious—we need to invert a 6×6 matrix during each iteration of the Newton-Raphson algorithm. A less obvious disadvantage is that, in our experiments, this algorithm shows worse numerical stability than the algorithm built on constraint functions in

situations where the features are very close to triple degeneracy.

Variation on a Theme. A useful form of 3D tensor is the *deviator*. It is simply a 3D tensor whose trace is zero, which implies that the sum of the eigenvalues is also zero. We can obtain the deviator part of any 3D tensor T by subtracting one-third of its trace from its three diagonal components. Since this is a linear operation, the zero-trace property is preserved on a discrete grid using trilinear interpolation.

One variation of the basic geometric algorithm is to consider only the deviator field of the original tensors. For the case of extracting type P degenerate features, it is easy to get:

$$s = \frac{V_x^2 + V_y^2 + V_z^2}{3}. \tag{15}$$

Substituting this term back into (12) and throwing away any redundant diagonal equation, we get a system with five equations and five unknowns. In our experiments, we found that this variation is almost equivalent to the original algorithm in (12) in terms of numerical stability and convergence speed.

5.4 Connecting Feature Points

After the feature points on cell faces have been extracted using any of the methods presented, the task still remains of connecting these points to form feature lines. We note that some cells may have more than one pair of intersection points and, hence, more than one feature line through it. To handle such cases, we use a multipass approach to connect these intersection points. We only examine candidate cells that contain intersection points on at least one of their six faces. In the first pass, all candidate cells containing exactly two intersection points are processed by: 1) simply connecting those two points, 2) recording the orientation of the line segment as tangents at the end points, and 3) marking the cell as processed. In the subsequent passes, their unprocessed neighboring candidate cells are processed by connecting a line segment between each pair of intersection points in such a way as to minimize the angle deviation between the tangent recorded at the end point and the line toward other intersection points within the cell. Each neighboring candidate cell is marked as processed and the procedure continues until there are no more candidate cells.

We use this iterative method to generate the tangent lines from feature points and ultimately resolve the line connections between multiple points. In the next section, we show how to analytically calculate the tangent at any point along a topological feature line.

6 Discriminant Hessian Factorization

6.1 Problem Statement

We are interested in finding the points where the symmetric tensors are degenerate and computing the tangents at those points. The tangents are useful in guiding the search for other degenerate points along topological feature lines and can be obtained given the degenerate tensor value and its gradient. This section describes a formulation obtained through Hessian factorization to produce an analytical form of the tangents. The Hessian factorization is just a theoretical analysis to get the formula. No factorization is actually done during the computation stage.

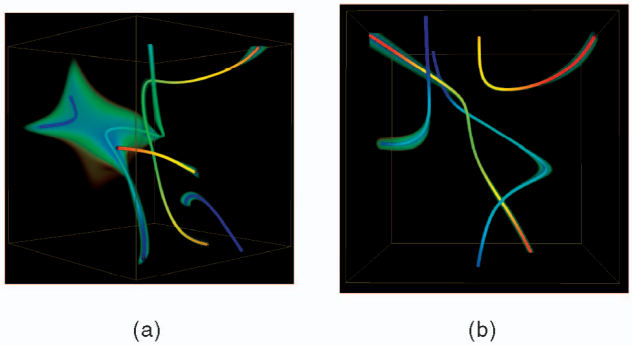


Fig. 3. Two randomly generated 3D tensors. Warmer line colors are closer to type P degenerate points where major and medium hyperstreamlines intersect, while cooler line colors are closer to type L degenerate points where medium and minor hyperstreamlines intersect. The rest of the volume is pseudocolored by the discriminant using cool colors for low discriminant values (closer to feature lines) and warm transparent colors for distant values. The image is exactly the same as the one in [27], although the degenerate points in this image are extracted using the geometric approach instead of the constraint functions.

A real symmetric matrix T of size N has $K = N(N+1)/2$ free components. All of the components together can be written as a single vector of size K. This vector can be used to obtain a $K \times K$ Hessian matrix of its discriminant. A better formulation is to use high-order tensors, but, since most practitioners in the visualization field are more familiar with vectors and matrices, we will make no distinction between an $N \times N$ matrix, T, and its K dimensional vector form, $\vec{T}$. We define a multiplication-sum operation on two symmetric tensors: $C = A \circ B = \sum_{i \le j} A_{ij} B_{ij}$. This is also the dot product of the vector forms of the two matrices.

We mentioned in Section 4 that an early theorem by Wigner and von Neumann points out that degenerate real symmetric matrices form a variety of codimension two. We derived the following lemma based on this theorem:

Lemma 6.1. *The Hessians of the discriminants on degenerate tensors has a rank of at most two.*

Because the Hessian matrix, $H = \nabla_T^2 D$, is positive semi-definite at degenerate tensors, it follows from Lemma 6.1 that we can factorize H as a product of two matrices:

$$H = F \cdot F^T, \tag{16}$$

where F is a $K \times 2$ matrix. Our task is to look for an analytical solution of F in terms of tensor values T and its local gradient $\nabla T = (T_x, T_y, T_z)$. The reason we are interested in this factorization is to study the local behavior of degenerate features, especially the tangents along the degenerate line. Given a tensor field $T(X)$ and a degenerate point X^*, the Taylor's expansion of the discriminant D shows:

$$\begin{aligned} & D(X^* + \Delta X) \\ & \approx D(T(X^*)) + \nabla D(X^*) \cdot \Delta X + \frac{1}{2} \Delta X^T \cdot \nabla_X^2 D(X^*) \cdot \Delta X \\ & = \frac{1}{2} \Delta X^T \cdot H_X(X^*) \cdot \Delta X \\ & = \frac{1}{2} (\nabla T \Delta X)^T \cdot H(T^*) \cdot (\nabla T \Delta X), \end{aligned} \tag{17}$$

where $H_X = \nabla_X^2 D$ is the Hessian of discriminant with respect to X and, similarly, $H = \nabla_T^2 D$ is that with respect to T. This is because D and its gradient ∇D both vanish at degenerate points. Hence, the Hessians are very useful in studying the local behavior around a degenerate point. If the factorization F is found, we can further rewrite (17) as:

$$\begin{aligned} D(X^* + \Delta X) &\approx \frac{1}{2}(\nabla T \Delta X)^T \cdot H(T^*) \cdot (\nabla T \Delta X) \\ &= \frac{1}{2}(F^T \cdot \nabla T \cdot \Delta X)^T (F^T \cdot \nabla T \cdot \Delta X) \\ &= \frac{1}{2}\|F^T \cdot \nabla T \cdot \Delta X\|^2. \end{aligned} \quad (18)$$

Equation (18) equals zero only if:

$$(F^T \cdot \nabla T) \cdot \Delta X = 0. \quad (19)$$

Equation (19) results in two equations and three variables. The tangent of a degenerate line is just its null space. If we have the analytical form factorization of F from the tensor value T, then getting its tangent by solving (19) is simply a cross product operation.

6.2 Analytical Factorization

Our solution to the factorization problem in (16) involves two stages. In the first stage, we consider the factorization in the local eigenvector coordinate. In the second stage, we rotate the results from the first stage back into the physical space. Note that the results can also be extended to discriminant Hessian factorization on time-varying tensor data.

6.2.1 Hessian Factorization in the Eigenvector Space

Given a degenerate tensor T^* at X^* and its eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$ and their associated eigenvectors $e_1, e_2, \ldots e_n$, a diagonal matrix, Λ, can be formed from the eigenvalues and an orthonormal eigen matrix, E, can be formed by tiling all the eigenvectors as column vectors. Without loss of generality, we assume $\lambda_1 = \lambda_2$ and there are no other equal roots. If there is, both the Hessian and its factorization will be zeros. The subblock of the matrix $\{T_{ij} : i \in \{1,2\}, j \in \{1,2\}\}$ forms a 2×2 matrix and is denoted as T_s.

We can study the Hessian matrix in the eigenvector coordinate E by transforming all the tensors using: $T^+ = E^T \cdot T \cdot E$. In this paper, all variables in the transformed coordinate have $^+$ to distinguish from their physical space counterparts. Note that this transformation is applied to every point in the space using the same E from T^*. T^+ is transformed into Λ at X^* and might not be diagonal elsewhere. It is also obvious that this transformation is a linear transformation and can be written as: $\vec{T^+} = R \cdot \vec{T}$ in their vector forms. It is not difficult to prove that R is a $K \times K$ unitary matrix such that $R^T \cdot R = I$ for any valid E. We now present the following theorem:

Theorem 6.2.

$$\left.\frac{\partial^2 D}{\partial T_{ij}^+ \partial T_{i'j'}^+}\right|_{T^+=\Lambda} = 0, i > 2 \text{ or } j > 2 \text{ or } i' > 2 \text{ or } j' > 2. \quad (20)$$

Theorem 6.2 states that the components outside T_s^+ do not contribute to the Hessian and therefore do not affect the degenerate line tangent. This theorem is the basis for the discriminant Hessian factorization technique.

Further analyses show that:

$$\begin{aligned} &\frac{\partial^2 D(\Lambda)}{\partial T_{11}^{+2}} = q^2, \ \frac{\partial^2 D(\Lambda)}{\partial T_{22}^{+2}} = q^2, \ \frac{\partial^2 D(\Lambda)}{\partial T_{11}^+ \partial T_{22}^+} = -q^2, \\ &\frac{\partial^2 D(\Lambda)}{\partial T_{12}^{+2}} = 4q^2, \ \frac{\partial^2 D(\Lambda)}{\partial T_{11}^+ \partial T_{12}^+} = 0, \ \frac{\partial^2 D(\Lambda)}{\partial T_{12}^+ \partial T_{22}^+} = 0, \\ &q = \sqrt{2} \prod_{2<i<j} |\lambda_i - \lambda_j| \prod_{i=3}^{N} (\lambda_i - \lambda_1)^2. \end{aligned} \quad (21)$$

Equation (21) shows that the Hessian of D has only four nonzero components. We denote $F = (F_1, F_2)$, where F_1 and F_2 are the two column vectors of F which can also be expressed as two real symmetric matrices. F^+ can be obtained from (21) analytically,

$$F_1^+ = \begin{pmatrix} q & 0 & \cdots & 0 \\ 0 & -q & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{pmatrix}, \quad (22)$$

$$F_2^+ = \begin{pmatrix} 0 & q & \cdots & 0 \\ q & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{pmatrix}. \quad (23)$$

Noting that $\nabla T \Delta X = \Delta T$ and substituting them back into (19), we have $F_1^+ \circ \Delta T^+ = q(\Delta T_{11}^+ - \Delta T_{22}^+) = 0$ and $F_2^+ \circ \Delta T^+ = q(2\Delta T_{12}^+) = 0$. This is the condition that keeps ΔT_s^+ degenerate. Therefore, in the eigenvector coordinate space, the line tangent at a degenerate point, which is the direction that keeps the *entire tensor* T^+ degenerate, is simply the direction that keeps *its submatrix* T_s^+ degenerate.

6.2.2 Hessian Factorization in the Physical Space

We have shown that the Hessian factorization in the eigenvector coordinate space is $H^+ = F^+ \cdot F^{+T}$. Since the transformation between the two coordinate spaces, expressed in vector form, is $\vec{T^+} = R \cdot \vec{T}$, we get:

$$\begin{aligned} H &= F \cdot F^T = \nabla_T^2 D = R^T \cdot \nabla_{T^+}^2 D \cdot R \\ &= R^T \cdot H^+ \cdot R = R^T \cdot F^+ \cdot F^{+T} \cdot R \\ \Longrightarrow & F = R^T \cdot F^+ = R^{-1} \cdot F^+ \\ \Longrightarrow & \vec{F_1} = R^{-1} \cdot \vec{F_1^+}, \vec{F_2} = R^{-1} \cdot \vec{F_2^+} \\ \Longrightarrow & F_1 = E \cdot F_1^+ \cdot E^T, F_2 = E \cdot F_2^+ \cdot E^T. \end{aligned} \quad (24)$$

The second equation comes from R being a unitary matrix and $R^T = R^{-1}$. Equation (24) gives out the formulation of the factorization F in terms of E and the constant F^+. However, F is not unique since the definition of E has one degree of freedom due to the two equal eigenvalues. It corresponds to the fact that if F is a valid factorization of H, $F \cdot S$ for any 2D unitary matrix S is another valid solution. Although the choice of S affects the value of F, it does not change H and the null space of $F^T \cdot \nabla T$.

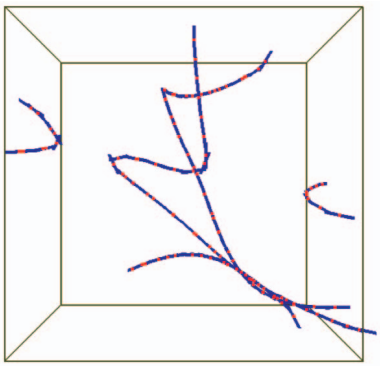


Fig. 4. Degenerate points in red extracted using iterative methods. Analytical tangents are calculated for each point and shown as blue arrows. We see that the tangents accurately match the degenerate lines and can be used to trace the feature lines.

For 3D tensors, the result is:

$$F_1 = \begin{pmatrix} E_{1x}^2 - E_{2x}^2 & E_{1x}E_{1y} - E_{2x}E_{2y} & E_{1x}E_{1z} - E_{2x}E_{2z} \\ E_{1x}E_{1y} - E_{2x}E_{2y} & E_{1y}^2 - E_{2y}^2 & E_{1y}E_{1z} - E_{2y}E_{2z} \\ E_{1x}E_{1z} - E_{2x}E_{2z} & E_{1y}E_{1z} - E_{2y}E_{2z} & E_{1z}^2 - E_{2z}^2 \end{pmatrix}, \tag{25}$$

$$F_2 = \begin{pmatrix} 2E_{1x}E_{2x} & E_{1x}E_{2y} + E_{1y}E_{2x} & E_{1x}E_{2z} + E_{1z}E_{2x} \\ E_{1x}E_{2y} + E_{1y}E_{2x} & 2E_{1y}E_{2y} & E_{1y}E_{2z} + E_{1z}E_{2y} \\ E_{1x}E_{2z} + E_{1z}E_{2x} & E_{1y}E_{2z} + E_{1z}E_{2y} & 2E_{1z}E_{2z} \end{pmatrix}, \tag{26}$$

$$N_1 = < F_1 \circ T_x, F_1 \circ T_y, F_1 \circ T_z >, \tag{27}$$

$$N_2 = < F_2 \circ T_x, F_2 \circ T_y, F_2 \circ T_z >, \tag{28}$$

$$N = N_1 \times N_2, \tag{29}$$

where N is the tangent of the degenerate line at the degenerate tensor T with tensor gradient $< T_x, T_y, T_z >$. It is worth noting that E_1 and E_2 are indeterminate and E_3 is the only determinate eigenvector. So, theoretically, it is enough to determine N only with E_3 and ∇T. But, E_3 does not show up in (25) and (26). We verify that any valid orthogonal combination of E_1 and E_2 will lead to the same N.

Another implication of this fact is that we will obtain different formulas for calculating tangents at degenerate tensors if we use different combinations of E_1 and E_2. Although they all lead to the same result at the degenerate tensor, the results are different when the point is away from degeneracy. Therefore, the choice of a proper formula does affect the quality of our numerical algorithm.

Note that (25) to (29) are enough to determine the tangent of a 3D degenerate line given a degenerate point and its tensor gradient. Hessian factorization is used to derive these equations, although it is not needed in the calculation of tangents. Fig. 4 uses these formulas to calculate the tangents of degenerate lines on extracted points. It also visually verifies the correctness of these formulas so that they can be used to further improve the extraction algorithm.

7 Topological Lines from Degenerate Points

A brute force degenerate point extraction and connection algorithm has several serious drawbacks. First, it is very inefficient. To make the extracted feature lines smoother, e.g., to double its resolution, one must double the spatial resolution of the data and thus make the whole computation eight times slower. Second, although algorithms built on constraint functions and the geometric approach are much more stable than direct discriminant minimization, they still may not capture all the features due to the possibility of multiple degenerate points in one cell. With analytically computed line tangents, we can address the connectivity problem by simply tracing the feature points using the tangent in a similar fashion as in feature flow field [18]. In addition, we can also capture the whole feature line from a single extracted point. We can further improve the efficiency by using a coarser extraction grid.

In this section, we introduce several strategies to connect the extracted feature points into feature lines.

7.1 Improved Postprocessing

An improvement over the brute force connection algorithm is to replace the estimated tangent with the analytically calculated tangents. When there are more than two features in a cell, the connectivity problem is addressed by minimizing the angular deviation of the tangent and the candidate features. This simple improvement eliminates the need for the multipass algorithm.

7.2 Tracing Degenerate Lines Using Tangents

With the analytically calculated tangents, the degenerate lines can simply be traced in the same manner as integrating a streamline from a seed point. This is also similar to the approach of tracing critical points over time with feature flow fields [18]. From each point, we can guess the next point using the tangent at the current point. Different integration algorithms, such as Euler and fourth order Runge-Kutta algorithms, are all applicable. During tracing, when the feature line hits a cell face, we should check for matches with the degenerate points on that face. If the error falls within a predefined threshold, the feature line will be connected back to the closest feature point. If the error is outside the threshold and the discriminant is small enough, then we have found a new degenerate point (and a new feature line trace is started). This new degenerate point will be added to the face and the cell on the opposite side of the face will be marked as unprocessed.

It is worth noting that, although (29) is defined on degenerate tensors, it can also be evaluated at nondegenerate tensors, although this may be physically meaningless. Different choices of E_1 and E_2 will lead to algorithms that are equivalent on degenerate tensors, but could behave differently when the tracing is off the degenerate lines.

A drawback of this algorithm is that the tangents will be zero at points with triple degeneracy or at crossings of two degenerate lines of the same type. In our experiments, the line orientation becomes fairly complicated in these cases. Therefore, we switch back to the brute force extraction-connection algorithm when we detect that we are near

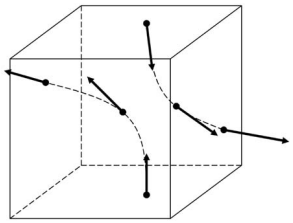


Fig. 5. Tracing degenerate lines using the tangents.

points with triple degeneracy or when the calculated tangent is too small. Since this rarely happens, we can afford to use a fairly high resolution of features without greatly reducing the performance of the entire system.

An advantage of this simple tracing scheme is that it gives users control over quality and efficiency. For better quality, users simply choose smaller integration steps or employ the fourth order Runge-Kutta method. For a quicker result, the Euler method with relatively large integration steps can be used, although it is not advisable. If only a rough overview is needed, the improved postprocessing algorithm using the analytical tangents is sufficient. Fig. 5 shows how the tracing algorithm addresses the multiple features problem in one cell.

7.3 Extraction Grid

Since the feature lines can be traced independently of the point extraction and the extraction algorithm is much more expensive than the tracing step, we can use a grid with a lower resolution than the expected feature resolution during extraction. In our implementation, we typically choose the original grid or a grid with half the original resolution and an integration step that is one eighth of the size of the grid spacing. This produces high quality features with relatively low computational expense for point extraction. A bolder choice of low quality extraction grid is possible. This is simply a trade-off between accuracy and efficiency.

7.4 Prediction-Correction

A drawback of simply tracing the feature as streamlines is that integration errors are accumulated. One can choose a better integration scheme, such as the Runge-Kutta algorithm, for better quality. Still, the accuracy of the feature line is not guaranteed. Therefore, we develop other algorithms that use the tangent to predict the next point along the feature line and then resort back to the iterative method to correct the prediction.

The Newton-Raphson algorithm and its variants are known for their superior convergence speed when the estimate is sufficiently close to the real solution. In our algorithm, the next point along the feature line is predicted as the current point plus an offset defined with the analytical tangent. Next, a slice on one of the orthogonal planes most perpendicular to the tangent direction at the predicted point is chosen. The algorithm then uses the iterative method built on constraint functions or the geometric approach to refine the result. Since the predicted point is very close to the true feature, it usually takes only one or two iterations to converge.

The refined point is then used as the starting point in the next step. This algorithm is slower than the algorithms built directly on tracing, but its advantage is that the errors are not accumulated. Still, it is much faster than the brute force extraction-connection algorithm because it does not waste any computational resources on cells that are too far from the features during the extraction stage.

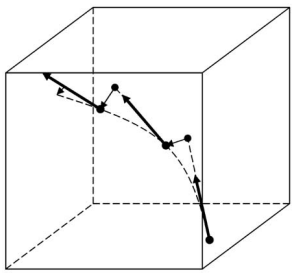


Fig. 6. Tracing degenerate lines with analytical tangents using prediction-correction.

Fig. 6 shows how the tracing algorithm combines the prediction and correction stages. With this method, the errors are not accumulated.

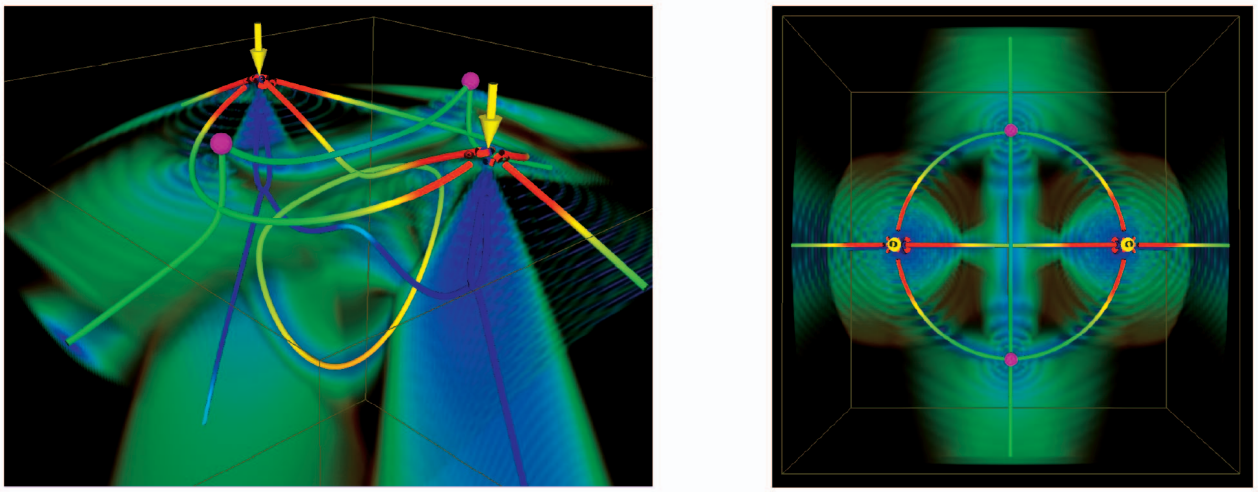
7.5 Coherent Tangents Formula

Any formula from (29) that produces valid eigenvectors on degenerate tensors will yield valid tangents of degenerate lines. However, their performance may differ when tracing points that are further away from the degenerate lines. The first choice of E_1 and E_2 is the eigenvectors of the tensor at that point. But, the value of E_1 and E_2 can vary dramatically around the degenerate lines, thus introducing errors. Another choice is to start from the outer eigenvector E_3 to consistently produce E_1 and E_2. This algorithm produces line tangents with more consistent behavior around degeneracies and is much friendlier to the fourth order Runge-Kutta algorithm, which relies heavily on the continuity assumption. However, this algorithm is not rotation invariant and may yield a biased result.

7.6 Higher Order Degeneracy

Previous researchers have pointed out that tensor features may have higher order degeneracy than just points and may include lines, surfaces, and subvolumes. It can be proven that not only is the discriminant zero at the degenerate tensors, but also the first order gradient of the discriminant is also equal to zero. We hypothesize that the higher order degeneracy happens only where the second or higher order of discriminant gradients are zeros. One can find this kind of phenomenon only under very special conditions. For example, we can show that, for the single point load data, there are two types of degenerate features: first, a feature line directly below the point load direction; second, a surface that spreads symmetrically down from the point load. The higher order surface degeneracy happens in this particular data set because of a delicate equilibrium achievable only in a purely synthetic data. This equilibrium can be easily disturbed by noise or other forces and the feature will fall back to the more stable form of lines. This is confirmed in the double point load data. In the double point load data, although the discriminants on the degenerate surface are still very small, as indicated by the almost transparent surface below each point load in Fig. 7, the stable tensor features are lines.

In the same figure, we notice that, near the apex of these two weak surfaces, the feature lines seem to break apart. We think that, in the vicinity of those two points, the appropriate tensor feature is a small feature surface. Since



(a) (b)

Fig. 7. Double point load data. Yellow arrows indicate point load directions, while the two magenta spheres show the location of the triple degenerate points. Feature lines are colored by the eigen difference, while the volume is colored by the discriminant. (a) Oblique view. (b) Top view.

our current algorithm is not designed for finding such features, it appears as an artifact. This is an area that we plan on addressing in the future, even though such features are generally unstable.

Another generally unstable higher order feature is feature subvolumes. We think that, in some types of tensor data sets, such as diffusion tensor data sets, it may be more likely that such features will appear. The locations of such features can be found in purely isotropic regions and, while they may not be of too much interest, we also plan to investigate this aspect for the sake of completeness.

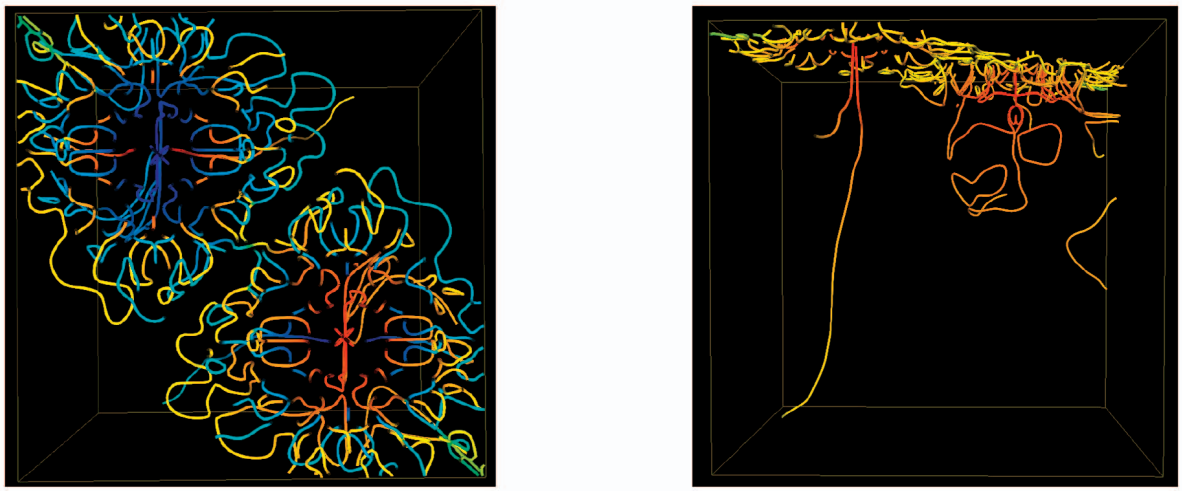
8 Results

We experimented with four data sets to test out our degenerate tensor extraction algorithm. The first set is a 2D rectangular patch with symmetric 3D tensors at the four corners that have been set randomly (see Fig. 1). The tensor values within the patch are obtained through linear interpolation. This synthetic data corresponds to tensors on a face of a 3D cell. The second is a 3D cell with symmetric 3D tensors on its eight corners which are also set randomly (see Fig. 3). It is resampled into a higher resolution for smoother feature lines. The third is the stress tensor data in a semi-infinite volume with two point loads (see Fig. 7). The fourth contains stress tensors from a numerical simulation of a box with a compressive and a tensile force on its top face. The forces are aligned along a diagonal of the top face (see Fig. 8). For Figs. 3 and 7, the colors of the volumes are mapped to the tensor discriminant (6) with more transparent cooler colors mapped to lower values and more opaque warmer colors mapped to higher values. Degenerate tensors can be found in the cool blue regions. Digital images can be accessed online at: www.cse.ucsc.edu/research/avis/hessian.html.

Fig. 3 shows degenerate tensors in a 3D cell form feature lines (rendered as tubes). Note that the feature lines are *not* hyperstreamlines, rather they are places where the major and medium or the medium and minor or all three hyperstreamlines intersect each other. The color of the tubes are mapped to the eigen difference, where the type P lines are mapped to warmer colors and the type L lines are mapped to cooler colors. Only the faint green is visible in the vicinity of the tubes because the tubes are in the blue regions. We see that complex feature lines can form even from a simple linearly interpolated random tensor field. In Fig. 3a, the type P and L lines swirl around each other, while, in Fig. 3b, the two types of lines form a complicated structure.

Fig. 7 shows the double point load stress tensors. The yellow arrows indicate the two point loads and the two magenta spheres are the triple degenerate points. We can see the line of double degeneracy connecting these two stress-free points to which Hesselink et al. alluded in [9]. Other very interesting feature lines are also extracted. The first is a vertical loop that lies directly under the double degenerate feature line connecting the two triple degenerate points. This feature is not present in the single point load data. The loop feature is also stable in the sense that it persists even as the magnitudes of the two point loads are varied. Another visible feature is how the blue lines below each of the point loads bifurcate and then reconnect. These two structures and the vertical loop are connected together by a type P feature line running between the two point loads. Looking from the top view in Fig. 7b, we see a third interesting feature, which is the circular feature line that connects the two point loads and the two triple degenerate points. We need to further investigate the physical significance of these features that have not been seen in previous visualizations of the data. It is worth noting that the stress tensor is dominated by only one single load in the vicinity of the load point. Therefore, it is locally similar to the single point load stress tensor where the degenerate tensor form a surface symmetrically spreading away from the load point. Since our algorithm is designed for extracting features lines, it produces artifacts when the features form a surface or subvolume.

Fig. 8 shows degenerate lines in the stress tensor field with two opposing forces on the top surface of a box. The



(a) (b)

Fig. 8. Degenerate lines in a box with two forces on top, one compressive and the other tensile. The forces are aligned along a diagnoal of the top face. Feature lines are colored by the eigendifference. (a) Top view with type-P and type-L lines. (b) Front view with only type-P lines.

tensors are calculated at Gauss points in a rectilinearly gridded field. The degenerate lines are much more complex than the double point load tensor field. We have separated the type-P from the type-L lines to reduce the clutter. Notice how the short lines tend to cluster near the top surface. The current hypothesis is that these are due to numerical artifacts as the values of the stress tensors near the surface are expected to be small. These small numbers may possibly be similar due to the limited precision in the simulation. The most important lines are those beneath the two main loads. Similarly, one can notice some loop structures similar to those found in Fig. 7. Further analyses are needed to determine their significance.

9 Conclusion

We pointed out that the degenerate tensors form stable feature lines in 3D real symmetric tensor field. This knowledge lays the foundation for future research on topology-based methods to visualize 3D tensor fields. Finding these feature lines is divided into two independent steps: extracting the degenerate points and forming feature lines from these points. Aside from the classic cubic discriminant for finding degenerate points, two new, more efficient, and computationally accurate methods were presented. The first is a reformulation of the discriminant into seven constraint functions, while the second one is based on a more intuitive geometric interpretation of degenerate tensors. The task of finding the feature lines given the degenerate points is also addressed and several algorithms are presented. One of the key contributions here is finding an analytical form for calculating tangents at degenerate points. The tangents are used to trace the rest of the feature lines.

We tested our algorithms on several data sets including randomly generated tensor fields, which allowed us to stress test our algorithms, several analytical data sets, such as the single and double point load data sets, to validate our results, and several computational data sets, such as the flow past cylinder with hemispherical cap and the stress tensor data in Fig. 8, to test its practical use. The results reveal new information—for the case of the double point load data set, as well as additional areas of investigation such as studying the correlation between the interesting patterns we saw in the real data sets and the underlying physics. These new insights will be useful in seeding hyperstreamlines, topology simplification, and tracking topology in time-varying data.

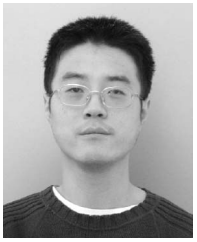
Acknowledgments

This work is supported by US National Science Foundation grant ACI-9908881. The stress tensor data in Fig. 8 is courtesy of Boris Jeremic. The authors would like to thank Peter Lax, Allen van Gelder, Joel Yellin, and Xavier Tricoche for their correspondence and suggestions. The comments by the anonymous reviewers also helped improve this paper.

References

[1] Transversality, planetmath.org/encyclopedia/Transversality.html, 2003.
[2] C.L. Bajaj, V. Pascucci, and D.R. Schikore, "Visualization of Scalar Topology for Structural Enhancement," *Proc. Visualization '98,* pp. 51-58, 1998.
[3] A. Bhalerao and C.-F. Westin, "Tensor Splats: Visualising Tensor Fields by Texture Mapped Volume Rendering," *Proc. Sixth Int'l Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI '03),* pp. 294-901, Nov. 2003.
[4] E. Boring and A. Pang, "Interactive Deformations from Tensor Fields," *Proc. IEEE Visualization '98,* D. Ebert, H. Hagen, and H. Rushmeier, eds., pp. 297-304, 1998.
[5] W.C. de Leeuw and J.J. van Wijk, "A Probe for Local Flow Field Visualization," *Proc. IEEE Visualization '93,* G.M. Nielson and D. Bergeron, eds., pp. 39-45, 1993.
[6] T. Delmarcelle and L. Hesselink, "Visualizing Second-Order Tensor Fields with Hyperstreamlines," *IEEE Computer Graphics and Applications,* vol. 13, no. 4, pp. 25-33, July 1993.
[7] T. Delmarcelle and L. Hesselink, "The Topology of Second-Order Tensor Fields," *Proc. IEEE Visualization '94,* R.D. Bergeron and A.E. Kaufman, eds., pp. 140-148, 1994.
[8] R.B. Haber, "Visualization Techniques for Engineering Mechanics," *Computing Systems in Eng.,* vol. 1, no. 1, pp. 37-50, 1990.
[9] L. Hesselink, Y. Levy, and Y. Lavin, "The Topology of Symmetric, Second-Order 3D Tensor Fields," *IEEE Trans. Visualization and Computer Graphics,* vol. 3, no. 1, pp. 1-11, Jan.-Mar. 1997.

[10] D. Hilbert, "Über die Darstellung definiter Formen als Summen von Formenquadraten," *Math. Annalen,* vol. 32, pp. 342-350, 1888.
[11] G. Kindlmann, "Superquadric Tensor Glyph," *Proc. Vissym '04,* pp. 147-154, 2004, http://www.cs.utah.edu/gk/papers/vissym04.
[12] G.L. Kindlmann and D.M. Weinstein, "Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields," *Proc. IEEE Visualization,* pp. 183-189, 1999.
[13] R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. Visualization '99,* D. Ebert, M. Gross, and B. Hamann, eds., pp. 333-340, 1999.
[14] P.D. Lax, "On the Discriminant of Real Symmetric Matrices," *Comm. Pure and Applied Math.,* vol. 51, nos. 11-12, pp. 1387-1396, 1998.
[15] P. Lax, *Linear Algebra.* Wiley, 1996.
[16] B.N. Parlett, "The (Matrix) Discriminant as a Determinant," *Linear Algebra and Its Applications,* vol. 355, pp. 85-101, 2002.
[17] G. Scheuermann and X. Tricoche, "Topological Methods for Flow Visualization," *Visualization Handbook,* C. Johnson and C. Hansen, eds., chapter 17, pp. 341-356, Elsevier, 2005.
[18] H. Theisel and H.-P. Seidel, "Feature Flow Fields," *Proc. Joint Eurographics-IEEE TCVG Symp. Visualization (VisSym '03),* 2003.
[19] X. Tricoche, G. Scheuermann, and H. Hagen, "Tensor Topology Tracking: A Visualization Method for Time-Dependent 2D Symmetric Tensor Fields," *Computer Graphics Forum,* vol. 20, no. 3, pp. 461-470, 2001.
[20] X. Tricoche and G. Scheuermannn, "Topology Simplification of Symmetric, Second-Order 2D Tensor Fields," *Geometric Modeling for Scientific Visualization,* G. Brunnet, B. Hamann, H. Müller, and L. Linsen, eds., pp. 171-184, Springer, 2003.
[21] W. Benger and H.-C. Hege, "Tensor Splats," *Visualization and Data Analysis,* pp. 151-162, 2004.
[22] T. Weinkauf, H. Theisel, H.C. Hege, and H.P. Seidel, "Boundary Switch Connectors for Topological Visualization of Complex 3D Vector Fields," *Proc. Joint Eurographics-IEEE TCVG Symp. Visualization (VisSym '04),* pp. 183-192, 2004.
[23] D.M. Weinstein, G.L Kindlmann, and E.C. Lundberg, "Tensorlines: Advection-Diffusion Based Propagation through Diffusion Tensor Fields," *Proc. IEEE Visualization '99,* D. Ebert, M. Gross, and B. Hamann, eds., pp. 249-254, 1999.
[24] X. Zheng and A. Pang, "Volume Deformation for Tensor Visualization," *Proc. Visualization '02,* pp. 379-386, 2002.
[25] X. Zheng and A. Pang, "HyperLIC," *Proc. Visualization '03,* pp. 249-256, 2003.
[26] X. Zheng and A. Pang, "Interaction of Light and Tensor Fields," *Proc. VisSym '03,* pp. 157-166, 295, May 2003, www.cse.ucsc. edu/research/avis/tensorray.html.
[27] X. Zheng and A. Pang, "Topological Lines in 3D Tensor Fields," *Proc. Visualization '04,* pp. 313-320, 2004.
[28] X. Zheng, X. Tricoche, and A. Pang, "Degenerate 3D Tensors," *Visualization and Processing of Tensor Fields,* J. Weickert and H. Hagen, eds., chapter 14, Springer, 2005.



Xiaoqiang Zheng received the BS degree from Tsinghua University in Beijing, People's Republic of China. He is currently a PhD student in the Computer Science Department at the University of California at Santa Cruz. His research interests include flow and tensor visualization, computer graphics, and image processing. He is a student member of the IEEE.

Beresford N. Parlett received the BA degree in mathematics from Oxford University in 1955 and the PhD degree, also in mathematics, from Stanford University in 1962. He taught in the Mathematics Department at the University of California, Berkeley, from 1965 until 1994, when he became a professor in the Graduate School. He was the chair of the Computer Science Department at the University of California, Berkeley, from 1967 until 1970. He is interested in matrix eigenvalue computations.

Alex Pang obtained the BS degree in industrial engineering from the University of the Philippines, and the MS and PhD degrees in computer science from the University of California at Los Angeles in 1984 and 1990, respectively. He is a professor in the Computer Science Department at the University of California at Santa Cruz and has worked in the areas of comparative and uncertainty visualization and flow and tensor visualization. He is a senior member of the IEEE and a member of the IEEE Computer Society.