Degenerate 3D Tensors

Xiaoqiang Zheng[1], Xavier Tricoche[2], and Alex Pang[1]

[1] University of California, Santa Cruz, CA 95064, USA
[2] University of Utah, Salt Lake City, UT 84112, USA

Abstract. Topological analysis of 3D tensor fields starts with the identification of degeneracies in the tensor field. In this paper, we present a new, intuitive and numerically stable method for finding degenerate tensors in symmetric second order 3D tensor fields. This method is based on a description of a tensor having an isotropic spherical component and a linear or planar component. As such, we refer to this formulation as the *geometric approach*. In this paper, we also show that the stable degenerate features in 3D tensor fields form lines. On the other hand, degenerate features that form points, surfaces or volumes are not stable and either disappear or turn into lines when noise is introduced into the system. These topological feature lines provide a compact representation of the 3D tensor field and are useful in helping scientists and engineers understand their complex nature.

1 Introduction

Tensor fields, especially second-order tensor fields, are useful in many medical, mechanical and physical applications such as: fluid dynamics, meteorology, molecular dynamics, biology, 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 themselves are inherently symmetric. In other cases, symmetric tensor data can be obtained through various decomposition techniques.

The main motivation and goal of this paper is to develop a simple yet powerful representation of 3D real symmetric tensor fields. Topology-based methods can yield simplified and effective depictions in many visualization fields. These methods consist of two parts: identifying the critical features, and their separatrices. Together, they divide the data space into regions with locally similar characteristics. Different types of topology can be extracted from a data field depending on how the critical features are defined. In this paper, we are interested in features defined by the relative magnitudes of the eigenvalues. That is, when some of the eigenvalues are equal, the resulting degeneracy is a critical feature. We assume that the tensor fields are continuous and differentiable. A typical tensor field is one where the tensors are "randomly" distributed in the tensor space, and does not have any inherent constraints such as having two equal eigenvalues at all times. On the other hand, a degenerate tensor field may have, by definition, two repeated eigenvalues everywhere – such as in momentum flux tensor fields defined as: $\Pi = V \cdot V^T$ for a flow field V. In this case, the tensor field is degenerate everywhere. The degenerate features are important in that they are the backbone from which the separatrices are anchored, and they provide a launching point for further analyses into the tensor field. For example, these features may form the basis for seeding hyperstreamlines [7].

Early work on using topology-based method to visualize tensor fields by [1,2] lays an important background for this research project. It defines the tensor topology based on degenerate features and discusses its nature for the 2D case in great detail, and provides useful knowledge for the 3D case. But we find this early work insufficient in studying 3D tensor topology. Not only is the dimensionality of the features unknown, but how to numerically extract the topological structures is also obscure. In their previous work, Hesselink et al. mentioned that the dimension of the degenerate features can be points, lines, surfaces or subvolumes. This claim itself is essentially true, but it does not point out the dimension of features in a typical 3D tensor field. By analogy, although the critical features (defined by locations where the velocity is zero) in 3D vector fields can be lines, surfaces or even subvolumes, we know they are mostly isolated points in a typical vector field. This knowledge is the foundation for the study of topological structure in vector visualization. All the subsequent study on separatrices are based on the extraction of the critical *points*. On the other hand, no topological results on 3D real symmetric tensor fields been published to date indicating that critical features in tensor fields form *lines*.

Our recent research [8], we found that the degenerate features in 3D tensor fields form feature lines that are stable even in the presence of noise. In the next section, we discuss the dimensionality of features in 3D tensor fields. Then, we quickly review the traditional method of finding degenerate features in 3D tensors fields based on *discriminants*, followed by the *constraint function* approach presented in [8]. Both of these methods are considered implicit functional approaches. This is followed by a presentation of our new method based on the geometric interpretation of a tensor.

2 Dimensionality of Degenerate Features

Similar to the 2D case described by Trichoche et al. in Chapter 13, a 3D real symmetric tensor can be decomposed into three orthogonal eigenvectors, each of which has a real eigenvalue associated with it. They are labeled as major, medium and minor eigenvectors according to the relative order of their eigenvalues. A tensor is degenerate when two or more of the eigenvalues are equal. The corresponding position in the tensor field is called a degenerate point. It follows that degenerate points are the only places where hyperstreamlines can cross each other, and therefore they are critical features in the tensor fields. The collection of these degenerate points constitutes the topological features of interest. Although the experience in flow visualization shows that a visualization restricted to topology alone may be incomplete and ignore essential features like vortex core lines, this analysis remains an important step towards better understanding of the complicated nature of 3D tensor data.

Before we can extract these topological features from 3D tensor fields, we need to know their dimension. Algorithms to locate points, lines, surfaces and volumes employ very different strategies. During our earlier work [8], we discovered that for most typical 3D tensor fields, the dimension of the topological feature is *one*, i.e. the collection of degenerate points form lines. This conclusion can be shown using an early theorem by von Neumann and Wigner which states that the real symmetric degenerate matrices form a variety of codimension two [5]. Codimension is defined as the difference between the dimension of a space and the dimension of a subspace contained in it. Read symmetric tensors in 3D 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 setting, tensor fields defined in 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. It can be shown by transversality that these dimensions satisfy the following formula: $\mathrm{codim}(A \cap B) = \mathrm{codim}(A) + \mathrm{codim}(B)$, which yields $\mathrm{codim}(A \cap B) = 2+3 = 5$, that is this intersection usually has a dimension one, i.e. forms lines. From the same line of reasoning, we know that degenerate tensors are isolated points in most cases if the data is specified in a 2D space. Since most numerical algorithms are designed to capture points, the basic block of our feature extraction algorithm is to locate 3D degenerate tensors on a 2D patch and then to connect them into lines afterwards.

While the main features are lines, it is still possible to obtain features that are points, surfaces or subvolumes. Features that form points, surfaces or subvolumes are less common in most 3D tensor fields and are usually induced by symmetry constraints. Such features are considered unstable and do not persist under perturbation. For example, a triple degenerate point where three eigenvalues are equal can be uniquely specified using one parameter (scaling of an identity matrix). In this case, previous computation yields a codimension $5 + 3 = 8 > 6$, which results in an unstable feature. Hence we focus our tensor feature extraction on lines rather than surfaces or subvolumes. Having said that, our extraction algorithm still needs to extract points first as these form the basis for finding the lines. Because of this design criterion, features that are surfaces (e.g. 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 not as common nor as stable.

3 Implicit Function Approach

The first family of methods to analyze degenerate tensors is through implicit functions. In this family, a tensor is degenerate if and only if its value makes an implicit function equal zero. Here we introduce two formulations: *discriminant* and *constraint function*. Note that since degenerate tensors form a variety of codimension two, an ideal formulation of the implicit constraint defining degenerate tensors should have two implicit functions. But neither of the two formula shown below has this property.

3.1 Discriminants

Hesselink et al. [2] define degenerate points as those tensors having at least two equal eigenvalues. Fortunately, we do not need to conduct the eigen-decomposition 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 tensor T with eigenvalues λ_1, λ_2 and λ_3 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 eigen-decomposition to determine 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 Equation 2, we can easily find that a discriminant is (a) always non-negative; (b) equal to zero if and only if at least two of the eigenvalues are equal. And it is ideal for computation and numerical purposes because although it is defined on eigenvalues, we do not really need to carry out an expensive eigen-decomposition. Instead, we only need to compute Equation 6 which is a polynomial of order six to get the discriminant.

An interesting geometric mapping of the three real eigenvalues is the Cardano circle or the eigenwheel (see Chapter 12 by Kindlmann). This is illustrated in Figure 1 where the three roots are the x-intercepts of the three axes that are 120 degrees apart. Note that the roots λ_1, λ_2 and λ_3 are increasing from left to right. The angle of the axes associated with the largest eigenvalue and the positive X axis is labeled as α. It is obvious that a double degeneracy occurs when $\alpha = 0$ resulting in $\lambda_1 = \lambda_2$, and $\alpha = 180$ resulting in $\lambda_2 = \lambda_3$. We refer to the first type of double degeneracy as Type L for linear, and the second type of double degeneracy as Type P for planar. A triple degenerate point occurs when all three eigenvalues are equal, and the radius of the circle reduces to zero. A very rare event indeed.

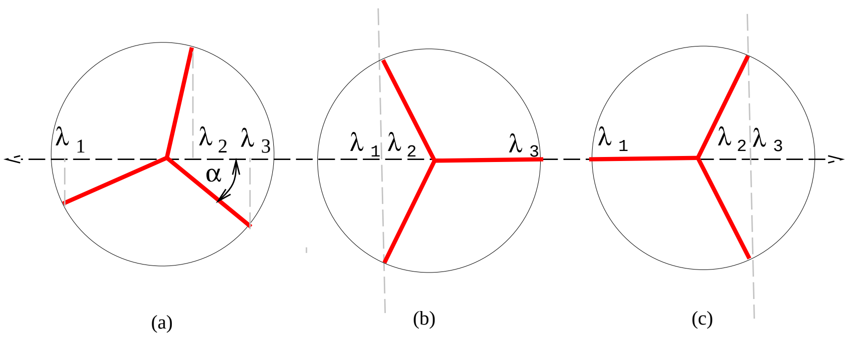


Fig. 1. Cardano's circle. The center of the circle is the mean of the three eigenvalues, and the eigenvalues are the x-coordinates of the line segments. (a) Relative positions of eigenvalues along the x-axis, (b) type L double degenerate point where the minor and medium eigenvalues are equal, and (c) type P double degenerate point where the medium and the major eigenvalues are equal.

3.2 Constraint Functions

In [8], an alternative formulation of degenerate points leading to a more stable numerical solution was presented. We briefly highlight the results here.

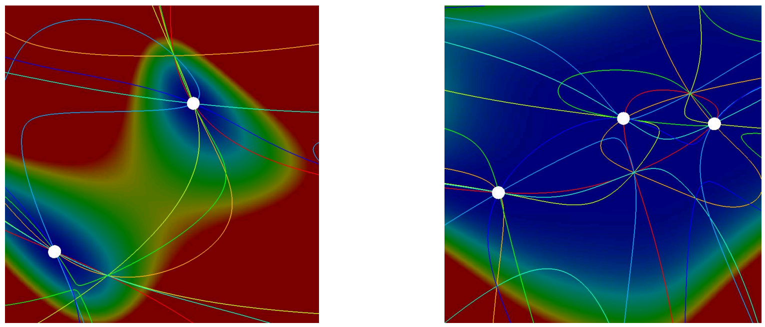
Although Equation 6 provides an elegant representation for evaluating the discriminant without having to perform eigen-decomposition, it is difficult to solve. In Equation 6, the discriminant of a real symmetric tensor is a polynomial of order six. Since it is always non-negative, the degenerate tensor also happens to be its minimum. Rather than using a minimization approach to find the degenerate tensors, the numerical analysis community recommends a root-finding strategy such as conjugate gradient for better numerical stability. A good method widely used to find the root of an equation is to detect the change of signs and then to recursively bisect the domain of interest. But because 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 root of a cubic discriminant stably is very difficult. Instead, we look for another representation of the discriminant.

In our previous investigation, we found that while Hilbert [3] pointed out that not all non-negative 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 [4], but our solution to the 3D case of *seven* equations is optimal [6] in the number of equations. Therefore, the definition of degenerate tensors can also be expressed as the tensors where the seven *constraint functions* are all zero at the same time. We use these seven cubic equations to extract the feature lines from 3D tensor fields. The seven discriminant constraints are:

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

$$\begin{aligned}
D_3(T) = f_x(T)^2 + f_{y1}(T)^2 + f_{y2}(T)^2 + f_{y3}(T)^2 + \\
15f_{z1}(T)^2 + 15f_{z2}(T)^2 + 15f_{z3}(T)^2
\end{aligned} \tag{7}$$

A tensor is degenerate if and only if all of its seven constraint functions are zero. This is the condition that we employ to extract the degenerate 3D tensors. 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 the algorithms that rely on finding degenerate points where the gradients of the discriminants are zeros. Its second advantage is that the constraint functions can be both positive or negative, as opposed to always being non-negative. This property allows us to perform a fast and inexpensive check for the existence of features. And finally, the reformulation also does not require eigen-decomposition.



(a) Two degenerate points (b) Three degenerate points

Fig. 2. 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 pseudo-colored by the discriminant functions. The data is a 2D slice of a randomly generated 3D tensor field. See color plates.

To find the degenerate lines, we visit all the cells in the data. For each cell, we first examine all the faces and find their intersections with the feature lines, if any. After all the degenerate points are extracted, we revisit the cells and connect them to form lines. Note that in our current implementation, we only consider regular hexahedral cells.

To find the degenerate points on a 2D face, we employ an iterative root finding method that satisfies all the seven constraint functions simultaneously. Note that although we are looking at a 2D face, the tensors are still 3D. Assume the tensor at location X is denoted by $T(X)$. For the feature points X^*, we have $\overrightarrow{CF}(X^*) = CF_i(X^*) = 0$, for $i = 1, ..., 7$, where $\overrightarrow{CF}(X)$ is an assembly of the seven constraint functions into one vector function. Using the Newton-Raphson method and an initial guess of X_n, we have the following conceptual algorithm,

$$\begin{aligned} X_{n+1} &= X_n - \left(\frac{\partial \overrightarrow{CF}}{\partial X}\right)^{-1} \cdot \overrightarrow{CF}\Bigg|_{X=X_n} \\ &= X_n - \left(\frac{\partial \overrightarrow{CF}}{\partial T} \cdot \frac{\partial T}{\partial X}\right)^{-1} \cdot \overrightarrow{CF}\Bigg|_{X=X_n} \end{aligned} \tag{8}$$

Note that we calculate the $\frac{\partial \overrightarrow{CF}}{\partial X}$ 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 the grid using finite difference methods for higher precision. $\frac{\partial \overrightarrow{CF}}{\partial T}$ 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.

However, Equation 8 does not work because on a cell face, X is only 2D while $\overrightarrow{CF}$ is 7D. Thus, $\frac{\partial \overrightarrow{CF}}{\partial X}$ is a 7×2 matrix. There are a number of ways to deal with such a system. In our case, we find that the least square estimator involving the transpose of the matrix works quite well.

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

This new hybrid algorithm minimizes the square error terms among the seven constraints. Using the center of each cell as the initial guess for an intersection point, we find that this method converges to the actual intersection point within five iterations in most non-degenerate cases with precisions up to 10^{-9}, and it almost never misses a feature point if it exists. Even if it happens, as evidenced by disconnected feature lines, the missing points can be recovered by subdividing the cell face or tracing the tangents of the feature lines [9]. This Newton-Raphson based method on constraint functions is superior in speed, accuracy and precision compared to other methods developed directly based on the cubic discriminants. For example, we also implemented a comparison 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%.

4 Geometric Approach

Since we want to extract the degenerate tensors in a root-finding framework, it is desirable to have a 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 discriminant is underspecified since there is only one equation but with two unknowns. An equation based on constraint functions is over-specified because there are seven equations with two unknowns. The formulation on constraint functions is better than its discriminant counterpart numerically because an over-specified 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.

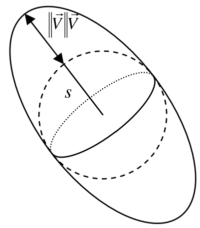


Fig. 3. Relationship between s and V and the degenerate tensor glyph.

Theorem 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 Figure 3)*.*

A linear tensor transforms all vectors onto a line. 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,

$$T = sI \pm V \cdot V^T \tag{10}$$

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, they only need to keep 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. A degenerate tensor on a 2D patch of a typical 3D real symmetric tensor field can be found by solving:

$$T(x, y) = sI \pm V \cdot V^T \tag{11}$$

Finding the location (x, y) of a four-parameter degenerate tensor on a 2D patch means we will have six unknowns. Since there are six independent components in real symmetric tensors, we can write a system of six equations with six unknowns. Such well-defined systems can be solved using any standard numerical method such as Newton-Raphson or one of its variants.

And since the problem is well-defined, we also expect it to have stable and isolated solutions.

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 $(\lambda_1 \leq \lambda_2 \leq \lambda_3)$ 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 four parameters (s, V), we use the following heuristic,

$$s_0 = \frac{\lambda_2 + \lambda_3}{2} \tag{12}$$

$$V_0 = \sqrt{s_0 - \lambda_1} \cdot e_1 \tag{13}$$

Using s_0 and V_0 for the initial guess, we iteratively update the six parameters using the Newton-Raphson method until convergence to a solution. 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. Besides 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 algorithms built on the constraint functions in situations when the features are very close to triple degeneracy.

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,

$$s = \frac{V_x^2 + V_y^2 + V_z^2}{3} \tag{14}$$

Substituting this term back into Equation 11 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 terms of numerical stability and convergence speed.

5 Topological Feature Lines

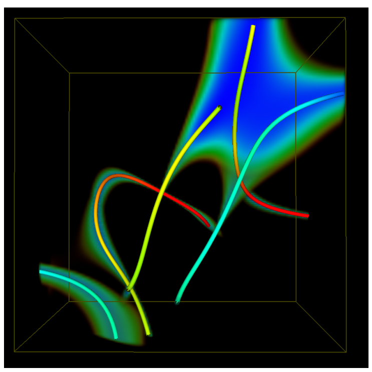
Now that we have obtained the degenerate points using one of the methods described in the two previous sections, the next step is to form the topological feature lines. The general idea is to connect the degenerate points on cell faces with those at neighboring cell faces. However, as we can clearly see in Figure 2, some cells may have more than one degenerate point, and hence more than one feature line going though them. We therefore use a multi-pass approach to connect these degenerate points. The procedure proceeds by examining only those candidate cells that contain degenerate points (i.e. intersection points of feature lines with the face) on at least one of their six faces. In the first pass, all candidate cells containing exactly two intersection points are processed by: (a) simply connecting those two points, (b) recording the orientation of the line segment as tangents at the end points, and (c) marking the cell as processed. In each subsequent pass, the number of candidate cells are further reduced by connecting the remaining intersection point(s) to points in neighboring cells that have been processed earlier, and therefore have tangent information. If a face has multiple degenerate points, we select the point that minimizes the angle between the resulting feature line and the previously computed tangent. Each candidate cell is marked as processed, and the procedure continues until there are no more candidate cells.

In our current implementation, we use this iterative method to generate the tangent lines on topological feature points and ultimately resolve the line connections between multiple points. In the future, we plan to calculate the tangent of the degenerate tensor line at a specific feature point analytically instead of this post-processing method.

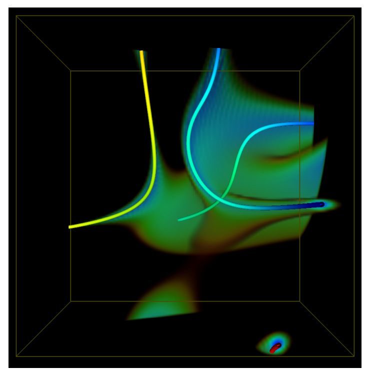
6 Results

We experimented with four data sets to test out our degenerate tensor extraction algorithm using the geometric approach. In the experiments, we use a pre-filtering algorithm that is similar to the one used in [8]. The first is a 2D rectangular patch with randomly set symmetric 3D tensor values at the four corners (see Figure 2). 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 hexahedral cell also with randomly set symmetric 3D tensors values at its eight corners (see Figure 4). It is resampled into a finer resolution for smoother features lines. The third is the stress tensor data in a semi-infinite volume with two point loads (see Figure 5). The fourth is the deformation tensors in the computed flow past a cylinder with hemispherical cap (see Figure 6). From Figures 4 to 5, the colors of the volumes are mapped to the tensor discriminant (Equation 6) with blue mapped with lower transparency to zero and warmer colors with higher transparency mapped to higher values. Degenerate tensors can be found in the blue regions. Additional digital images can be accessed online at: www.cse.ucsc.edu/research/avis/tensortopo.html.

Figure 4 shows degenerate tensors in a 3D cell form feature lines (rendered as tubes). Note that the feature lines are *not* hyperstreamlines, rather they are where the major and medium, or the medium and minor, or all three hyperstreamlines intersect each other. Only the faint green is visible in the vicinity of the tubes because the tubes are in the blue regions. The color of the tubes are such that type P points with very different minor value are mapped to warmer colors, and type L points with very different major value are mapped to cooler colors. The milder colors are where the other eigenvalue is not as different as the degenerate pair. We see that complex feature lines can form even from a simple linearly interpolated random tensor field.



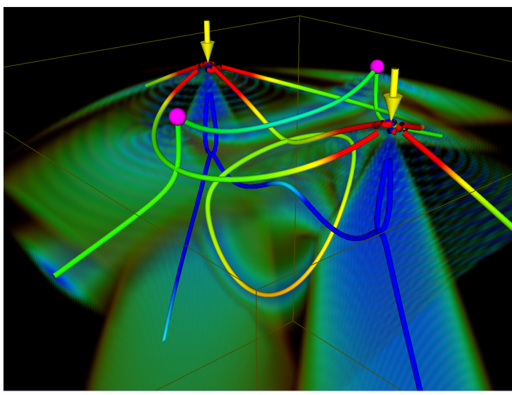
(a) First set



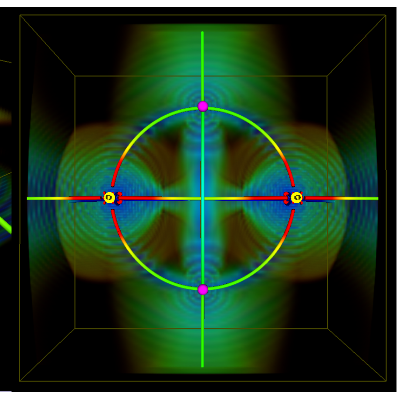
(b) Second set

Fig. 4. 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 pseudo-colored by the discriminant using cool colors for low discriminant values (closer to feature lines) and warm transparent colors for distant values. See color plates.

Figure 5 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 as alluded to in [2]. Other very interesting feature lines are also extracted: (1) 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. This loop feature is also stable in the sense that it persists even as the relative magnitudes of the two point loads are varied. (2) how the blue feature line below each of the point load 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 (b), we see another interesting feature which is the circular feature line that connects the two point loads and the two triple degenerate points. It is worth noting that in the vicinity below each load point, the stress tensors are similar to those found in single point load data sets where we have observed the degenerate tensors to form a conical *surface*. One can also make out these degenerate conical surfaces, particularly the one under the more distant point load in (a). Since our algorithm is designed for extracting features lines, it produces artifacts when the features form a surface or subvolume.



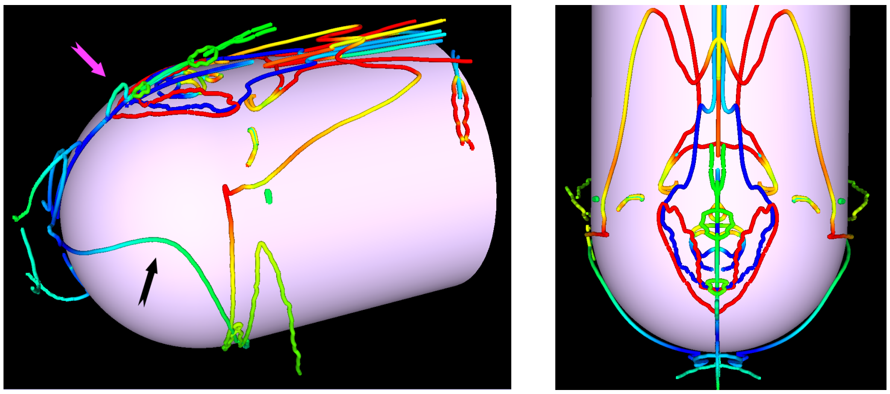
(a) Oblique view



(b) Top view

Fig. 5. Double point load data. Yellow arrows indicate point load, while the 2 magenta spheres show the location of the triple degenerate points. Color scheme is the same as Figure 4. See color plates.

Figure 6 shows degenerate lines in the deformation tensors of the computed flow at an angle past a cylinder with a hemispherical cap. Only a portion of the data close to the cap is shown because most of the interesting features are found there. A little bit of asymmetry is apparent on the left side of Figure 6(b) where the seam of the curvilinear grid wraps around. We see a curved line on the cap shown by the black arrow. It matches some of the patterns of the velocity topology from the same data set. There are more features at the upper half of the data because the flow there is more turbulent. Figure 6(a) is from an oblique view. Most of the features are close to the geometry of the object except a complicated branch structure which extends away from the geometry. It contains a small cyan horn shape pointed by the pink arrow, two green ring shapes, and a bifurcating structure. Figure 6(b) is from a top view. Recall that lines with cool colors have type L degeneracies while lines with cool colors have type P degeneracies. While we can see some lines that are greenish in color, we did not find any triple degeneracies. Also, while we can see very interesting complicated line structures, we have yet to explore their full significance as we still need to develop methods for extracting the separating surfaces in order to describe the full topology.



(a) Oblique View (b) Top View

Fig. 6. Degenerate lines in deformation tensors of flow past a cylinder with a hemispherical cap. Feature lines are colored as in Figure 4. See color plates.

The time statistics for the different data sets are summarized in Table 6. All the results are generated on a Dell Dimension 8100 with a single 1.5 GHz Pentium 4, 1Gb of memory, and an nVidia GeForce2 Ultra.

Data Set	Time
2D Random Patch (1×1)	0.05 (millisec)
3D Random Cell ($32 \times 32 \times 32$)	0.9 (sec)
Double Point Load ($64 \times 64 \times 64$)	4.2 (sec)
Hemisphere ($72 \times 110 \times 84$)	23 (sec)

Table 1. Time to extract degenerate tensors in different datasets

7 Open Problems

There are many open problems that need to be investigated. We highlight a few of them here.
First, since all the algorithms are built on extracting lines, they have problems dealing with features that form surfaces and volumes. Finding the separatrices of these features would also need to be addressed.

Second, in our numerical algorithms, we define the degenerate tensors as points with two equal eigenvalues without considering the eigenvectors. But we are originally interested in these points because they are where hyperstreamlines can cross each other, and hyperstreamlines are defined on eigenvectors. Therefore, a proper definition or an extraction algorithm based on both eigenvalues and eigenvectors could provide more insight into this problem. This is possible through a quantity solely defined on eigenvectors and is similar to the index of 2D critical points.

At each point along the degenerate lines, we can project the 3D tensors onto a 2D plane perpendicular to the eigenvector with distinct eigenvalue [7]. The projected 2D tensors also show a degenerate pattern. One can extract the separatrix of the 3D tensor field by calculating the separatrices of the projected 2D tensors at each point and connecting them together. With the 3D degenerate tensors and their separatrix surfaces, we will have a complete topological structure of 3D tensor fields.

Simplification and tracking of the topological structure proved to be useful for 2D tensor fields (see Chapter 13). In 3D, they are more difficult since the degenerate features become lines instead of points.

Finally these techniques should be extended to other application domains such as diffusion tensor (DT) MRI data described in other chapters of this book. While the main attribute of interest is usually the path of the fibers, highly linear fibers are where we would likely find feature lines. Hence, it is possible that topological visualization of DT-MRI data would also be beneficial. For this approach to be of real practical interest, the issue of topological artifacts induced by noisy data must be addressed as well. DT-MRI data are considered as typical tensor fields since there is no inherent constraint on the eigenvalues that may result in different codimensions. Although it could have large regions where the tensors are close to degeneracy, if one examines those points with very high precision and checks their persistence against noise, the results still form lines. That is, the topological features in such tensor fields form feature lines that are stable even in the presence of noise [8]. In practice, if an algorithm detects these nearly degenerate regions, such as in isotropic regions, it can mark off that region and skip the subsequent topological analysis because these regions contain very little topological information. Alternatively, those regions can be removed by post-processing or simplification. While we have found that feature lines are stable even in noisy environment, the topological structure (both for vectors and tensors) is quite sensitive. This can be used to our advantage, for example, when comparing topological structures from two tensor fields.

8 Conclusion

Double degenerate points in 3D real symmetric tensor fields form lines. These are the stable features. In this paper, we presented an intuitive geometric approach used in extracting these feature lines. It provides an alternative to the constraint functions and the discriminant function formulations. In all these three approaches, degenerate points are first extracted on each face of a candidate hexahedral cell. These points are then connected in an iterative fashion to generate feature lines. We applied this algorithm on several data sets including randomly generated tensor fields, the analytic double point load data set, and the computational data set on a flow past a cylinder with a hemispherical cap. The results from the double point load stress tensor field reveals new insight on a thoroughly studied data set by Hesselink et al. At the same time, the results also point to several areas that require further 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 designing strategies for seeding hyperstreamlines, topology simplification, and tracing topology in time-varying data.

9 Acknowledgment

This work is supported by NSF ACI-9908881. The flow data is courtesy of NASA. We would also like to thank Peter Lax and Beresford Parlett for their correspondence.

References

1. T. Delmarcelle and L. Hesselink. The topology of second-order tensor fields. In R.D. Bergeron and A.E. Kaufman, editors, *Proceedings IEEE Visualization '94*, pages 140–148. IEEE Computer Society Press, 1994.
2. L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, Jan-Mar 1997.
3. D. Hilbert. Über die Darstellung definiter Formen als Summen von Formenquadraten. *Math. Annalen*, 32:342–350, 1888.
4. P. D. Lax. On the discriminant of real symmetric matrices. *Communications on Pure and Applied Mathematics*, 51(11-12):1387–1396, 1998.
5. Peter Lax. *Linear Algebra.* Wiley, 1996.
6. B. N. Parlett. The (matrix) discriminant as a determinant. *Linear Algebra and its Applications*, 355:85–101, 2002.
7. Wei Shen. Seeding strategies for hyperstreamlines. Master's thesis, University of California, Santa Cruz, 2004.
8. Xiaoqiang Zheng and Alex Pang. Topological lines in 3D tensor fields. In *Proceedings of Visualization 04*, pages 313–320, Austin, 2004.
9. Xiaoqiang Zheng and Alex Pang. Topological lines in 3D tensor fields and discriminant hessian factorization. *IEEE Transactions on Visualization and Computer Graphics*, 2005. to appear.