

Visualizing Diffusion Tensor MR Image Using Streamtubes and Streamsurfaces

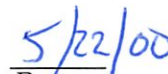
Song Zhang

May 22, 2000

Department of Computer Science
Brown University

Submitted in partial fulfillment of the requirements for the Degree of Master of Science in the
Department of Computer Science at Brown University


Advisor: David Laidlaw


Date

Abstract

Tensor-valued diffusion rate measurements have the potential to elucidate connectivity and other micro-structural information in the nervous system. We present a new method for visualizing 3D volumetric diffusion tensor MRI images. We distinguish between linear anisotropy and planar anisotropy and represent values within the two regimes using streamtubes and streamsurfaces, respectively. Streamtubes represent structures with primarily linear diffusion, typically fiber tracts; streamtube direction correlates with tract orientation. The cross-section shape and color of each streamtube are used to represent additional information of the diffusion tensor matrix at each point. Streamsurfaces represent structures in which diffusion is primarily planar. We also generate anatomical landmarks to identify the positions of outstanding structures, such as eyes, skull surface and ventricles. The final models are 2D surface geometries that can be imported into many interactive environments. Early results show promise for understanding connectivity in a volume and suggest that more anatomical context and additional interactivity will help make exploration more effective.

1 Introduction

In biological tissue, water is ubiquitous and constantly in motion. This motion of water inside biological tissue is called diffusion. Water usually diffuses at different rates both in different locations and along different directions at one location. The difference of diffusion rates at one location is called anisotropy. The diffusion of water is constrained by, and correlated with, micro-structure in the biological tissue. By visualizing diffusion information, we can elucidate the connectivity and micro-structural detail.

With MRI technology, diffusion information can be measured *in vivo* from biological tissues, producing a three-dimensional, second-order tensor field. This tensor field consists of a description of the diffusion information at every sample point in the three-dimensional volume.

Visualizing a diffusion tensor field is difficult because of the rich information it contains. Each sample point in the dataset consists of one diffusion tensor, represented by a 3×3 symmetric matrix. Thus, there are six dependent variables at each sample point. Visualizing these six variables separately loses the relationship among them and makes results difficult to interpret; however, simultaneously visualizing all the information available in a diffusion tensor dataset in one image is almost always impossible. In the literature, complete visualization of diffusion tensor fields is described for two-dimensional datasets. When extended to 3D, these methods all suffer from various limitations.

Our goal is to illustrate biological structures in the tissue. For this purpose, visualizing all the information is often unnecessary. Instead, we can filter out unimportant information and only visualize information that represents connectivity and micro-structural detail within the tissue. Our goal is to visualize this information in a way that mimics the underlying micro-structure.

We present an approach that distinguishes fibrous structures and laminar structures, and allows us to visualize these types of structures using what we call streamtubes and streamsurfaces, respectively. Streamtubes represent fibrous structures, which tend to produce primarily linear diffusion values. The cross-section and color of each streamtube are used to represent the remaining components of the diffusion measurements along the trajectory of fiber tracts in biological tissue. Streamsurfaces represent structures in which diffusion is primarily planar. To provide a context for streamtubes and streamsurfaces, we also generate anatomical landmarks such as eyes, the skull surface, and ventricles.

2 Previous work

In the literature, researchers have successfully designed complete visualization methods for slices of three-dimensional diffusion tensor fields. We discuss two of these methods, then move on to discuss attempts for three-dimensional visualization.

Pierpaoli *et al* use arrays of ellipsoids[6] to represent a two-dimensional diffusion tensor field. A diffusion tensor matrix is symmetric and has positive eigenvalues. These special properties make an ellipsoid a natural geometric representation of the diffusion tensor. Each axis of the ellipsoid represents one eigenvector and its corresponding eigenvalue. Laidlaw *et al* normalized the size of the ellipsoid to give a more continuous visual appearance [2].

Laidlaw *et al* [2][1] borrow concepts from oil painting. Two-dimensional brush strokes are used to represent different aspects of the diffusion tensors. The information is also classified and visualized in different layers such as the underpainting, checkerboard layer, and stroke layer.

These two-dimensional methods visualize a diffusion tensor field by completely visualizing the tensors of discrete sample points. There are two limitations of these methods when applied to a

three-dimensional dataset:

- By visualizing every sample point in the three-dimensional dataset, only the outermost layer of the dataset can be displayed on the screen. The internal data points will be blocked.
- The continuity inherent in biological tissues will not be properly represented in the final image. For example, the neural fibers in the brain that connect different anatomical regions would be difficult to locate within an array of ellipsoids.

To address these problems, several methods for three-dimensional diffusion tensor field visualization have been developed. Each of these three-dimensional method makes different choices about what subset of the tensor information to visualize and how to represent it.

Kindlmann *et al* [4] take a volume rendering approach to the problem. The philosophy behind this method is to display only some of the information, but display that information densely within a volume. The “hue-ball” and a “barycentric map” assign color and opacity to each location in the dataset based on the properties of the diffusion measurements. Although low opacity is assigned to unimportant data to avoid visual cluttering, in many cases there are a number of important data points along the direction of one ray. The compression of all these points into a single pixel makes it difficult for a viewer to extract precise positional information from the final image. This makes it difficult to pick out the path of a certain fibrous structure from its neighborhood. Also, a lack of interactivity limits a user’s understanding of the image.

Xue *et al* [7] use streamlines to visualize the vector field defined by the major eigenvector of the diffusion tensors. In this report, we use the term *major* eigenvector, *medium* eigenvector, and *minor* eigenvector, or the symbols e_1, e_2, e_3 , for the three eigenvectors of the diffusion tensor matrix with decreasing eigenvalues. The major eigenvector is in the direction of fastest diffusion. Xue *et al* reduce the problem of tensor field visualization to vector field visualization. While the result is clear and easy to interpret, it is constrained to a partial visualization of one aspect of the rich information provided by the diffusion tensor field.

Another interesting three-dimensional method uses hyperstreamlines, a visual icon for representing a subset of a three-dimensional tensor field[3]. This method is comparable to using ellipsoids to represent one diffusion tensor. The philosophy behind this method is to visualize *all* of the information but only at *some* of the locations in the volume. The trajectory of the hyperstreamline is generated by the vector field defined by the major eigenvector of the diffusion tensor, which could be interpreted as the path of water diffusing in the tissue. The cross-section shape and color along the trajectory encode information about the other two eigenvectors and the magnitude of the major eigenvector.

Our streamtubes are derived from hyperstreamlines. The hyperstreamline method has been applied to both stress tensor fields and momentum flux density tensor fields, but not to diffusion tensor fields. Several limitations appear when we try to use it for our purpose:

- The cross-section of a hyperstreamline sometimes grow quite large, limiting the density of hyperstreamlines in a scene. This limits the level of detail we can visualize.
- The hyperstreamline method visualizes all of the information along the trajectory of each line, but no information elsewhere. We need an algorithm for placing them so they will cover important information and not block each other.
- While a hyperstreamline is good for visualizing linear structures in biological tissue, it is not as effective for representing planar diffusion, which is constrained to a surface instead of a path.

In spite of these limitations, the hyperstreamline is promising for visualizing continuous micro-structures in biological tissue.

3 Our Approach

Our goal is to visualize the connectivity and micro-structural information in MR diffusion tensor images of biological tissue.

Our solution is to distinguish between linear and planar anisotropy regions, and employ streamtubes and streamsurfaces to visualize these two types of regions, respectively. Section 3.1 talks about classifying different kinds of anisotropies. Then, in section 3.2, we go on to discuss issues with the streamtubes, including their definition, the extension of the trajectories they follow, how to sample the starting points for an initial trajectory set so that all of the details in the data are covered, the strategy for selecting a representative set from the complete trajectory set, and the construction of a streamtube from a trajectory. Section 3.3 explain the strategies for similar problems in generating surfaces in planar anisotropy regions.

Based on the feedback from a preliminary visualization, we also generate geometric representations of anatomical landmarks to provide context. Section 3.4 talks about the method used for generating these landmarks.

Since all of the geometry is generated as two-dimensional meshes, the resulting data model can be displayed and manipulated in an interactive environment.

3.1 Anisotropy classification

Our method uses visual icons to display the micro-structural information contained in a three-dimensional diffusion tensor field. The effectiveness of the visual icon depends not only on how much information it can simultaneously display, but also on how well it visually matches the underlying biological structures, as represented by the diffusion tensors.

By examining the relationship between the three eigenvectors of the diffusion tensor matrix, we can distinguish three classes of diffusion tensors. The diffusion tensor with one eigenvalue much larger than the other two is said to have linear anisotropy. One example of a brain structure producing linear anisotropy is the myelin-encapsulated white matter[10]. The diffusion tensor with two large and one small eigenvalues is said to have planar anisotropy. Sheet-like structure in biological tissues are likely to result in planar anisotropy. Here, the water is constrained to diffuse primarily in the plane defined by the two large eigenvectors. Finally there are diffusion tensors whose three eigenvalues are roughly the same. This type of diffusion tensor is said to be isotropic, representing an underlying structure that is uniformly oriented. Gray matter in the brain produces isotropic diffusion tensor.

These three types of diffusion tensors are best represented using different visual icons. As discussed in the next two sections, we use streamtubes for regions of linear anisotropy, and streamsurfaces for regions of planar anisotropy. We ignore isotropic regions because they do not provide interesting information.

In the literature, there are many metrics for anisotropy. We want a metric that distinguishes these three classes of diffusion tensors. The metrics are typically defined as functions of the three eigenvalues of the diffusion tensor matrix, $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Some of them, such as rational anisotropy(RA) or fractional anisotropy(FA)[6], measure the variance of the eigenvalues, and provide a directionless scale value. These metrics, though useful for measuring differences in diffusion rates, are not suitable for distinguishing different shapes of anisotropies. One cannot tell if a diffusion tensor's high fractional anisotropy, or high rational anisotropy, results from linear

anisotropy or planar anisotropy. Westin’s metrics[8], on the other hand, measure different kinds of anisotropies, thus satisfying our requirement. The metrics are:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

where c_l represents linear anisotropy, c_p represents planar anisotropy, and c_s represents isotropy. This metric system has the property that $c_l + c_p + c_s = 1$. When the linear anisotropy value c_l is big, the other two values will be small. The same is true with c_p and c_s . Thus it is reasonable to set thresholds on each one of them to distinguish specific kinds of anisotropies. In our case, we do not want to visualize isotropy, so we ignore c_s . We set thresholds α and β such that $c_l > \alpha$ implies linear anisotropy and $c_p > \beta$ implies planar anisotropy. For example, the image shown in Figure 4 uses $c_l > 0.4$ along each streamtube and $c_p > 0.3$ on average planar anisotropy within each streamsurface.

This metric system is used to associate streamtubes with regions of linear anisotropy, and streamsurfaces with regions of planar anisotropy.

3.2 Linear anisotropy representation (streamtubes)

Linear structures, such as neural fibers in the human brain, constrain the diffusion of water along their axonal lengths. These structures give rise to diffusion tensors that exhibit linear anisotropy. The natural choice of a geometric primitive for representing linear anisotropy region is something with a linear geometric appearance, such as streamlines [7]. Additional criteria are the ability to carry additional information provided by the diffusion tensors, and the potential to avoid visual cluttering. Based on these criteria, we propose streamtubes as the geometric primitive for representing linear anisotropy.

The visual mapping of the streamtube is much like that of the hyperstreamline. The trajectory sweeps along the major vector field, and the cross-section shape is an ellipse representing the other two eigenvectors. We normalize the length of the medium eigenvector to a constant value so that the size of the streamtube is predictable, while the ratio between medium and minor eigenvalues remains intact. The color of the streamtube is related to the linear anisotropy value, with saturated red for the maximum, and white for none. Table 1 summarizes the visual mapping for a streamtube.

Trajectory	e_1 field
Cross-section	e_2 and e_3 , normalized
Color	c_l

Table 1: Visual mapping of diffusive parameters to streamtube attributes

The shape and size of a streamtube is determined primarily by its trajectory. In other words, the major eigenvector field is the primary information visualized by streamtubes. Our strategy

is to first fill the linear anisotropy regions with a dense set of streamtubes, and then select a representative set from it. There are several issues regarding streamtubes which we will discuss next. First, we explain how to track streamtube trajectories in the major eigenvector field. Second, we discuss how to pick the starting points to generate a dense set of trajectories that covers all of the linear anisotropy regions. Third, we show how to select a representative set of trajectories from the dense set. Finally we show how to visualize additional information such as direction of medium and minor eigenvectors.

3.2.1 Calculating a trajectory

Each streamline begins from a starting point, and extends both forward and backward, following the major eigenvector field. Our goal is to find the integral curve passing through the starting point. Because streamtubes are used for representing regions of high linear anisotropy, their trajectories should be restricted to these regions. Also, the streamtubes should be clipped to the data volume and to regions with high signal-to-noise ratios. To enforce these requirements, we integrate and extend trajectories only so long as:

1. The extended point is within the data volume.
2. The signal-to-noise ratio at the extending point is not too low. We use a T2-weighted intensity image as a measure for the signal-to-noise ratio of the diffusion tensor data. T2-weighted intensity image is a scalar dataset produced along with the diffusion tensor dataset. It indicates the intensity of the signal in the diffusion tensor field. If the T2-weighted intensity is low at one location, the signal-to-noise ratio of the diffusion tensor at the same location is also low.
3. The linear anisotropy value at the extended point is above the threshold. i. e. $c_l > \alpha$.

There are many ways to integrate the trajectory. The Euler method is the simplest method and essentially links step-size vectors together head to tail. However, the Euler method induces an error of $O(h^2)$, where h is the step length. As h becomes bigger, the integration will soon wander away from the desired curve.

To reduce the error, we use the second-order Runge-Kutta method, which reduces the error term to $O(h^3)$. We could also use a higher order Runge-Kutta method or a number of other method, to achieve more accuracy; however, there is a tradeoff between accuracy and running time.

3.2.2 Starting points sampling

Streamtube starting points should be chosen so that streamtubes will pass through all regions of high linear anisotropy. Recall that we will reduce the number of trajectories later, so it is acceptable to generate many trajectories now.

The starting points are not restricted to the sample points. At any point within the data volume, we can interpolate the tensor value using neighboring data points, so there is no limit on how many starting points we could use. However, while we would like to cover the voxel size level details, sampling too many points within one voxel does not make a lot of sense. We must also consider the computational expense. For the $256 \times 256 \times 144$ dataset shown in Figure 4, there will be nearly 10 million sample points. With a relatively large three-dimensional dataset like this one, the expense of starting a trajectory at every sample point is significant.

Now consider the sample points close to (for example, within one voxel of) a generated trajectory. If we generate a new trajectory from this point, it will be similar to the existing one with

high probability. Using this fact, we generate a starting point for each sample point, but remove those on the paths of existing trajectories. Thus, we cover all the data points, but with a reduced cost.

3.2.3 Trajectory selection

Once we have generated a dense set of trajectories, we have to decide which ones should be selected and visualized. With our sampling method, we produce more than 150,000 trajectories on an example human brain dataset. Visualizing them all is not only expensive, but also undesirable, as the inclusion of too many streamtubes in the scene will generate visual cluttering. Thus, a selection algorithm is necessary to filter out the duplicate trajectories and keep only a representative set. We use three criteria to measure the importance of each trajectory:

- *Length*—Trajectories shorter than a given threshold are removed. Short trajectories could be the result of either noise or reflection of short linear structures. For our purpose, short structures are less significant than long ones.
- *Average linear anisotropy*—By averaging the linear anisotropies at each point on a trajectory, we can get an idea of whether the trajectory passes through a region of high anisotropy. If so, it is assumed to represent a preferred direction of diffusion, and it is kept for visualization.
- *Similarity to previously selected trajectories*—We drop the trajectories that are too “close” to selected ones, thus removing the “duplicates”. We define the distance between two curves to be:

$$D = \begin{cases} 0 & \text{if } \int_{s_0}^{s_1} \max(\text{sgn}(\text{dist}(s) - T_l), 0) ds = 0 \\ \frac{\int_{s_0}^{s_1} \max(\text{dist}(s) - t, 0) ds}{\int_{s_0}^{s_1} \max(\text{sgn}(\text{dist}(s) - T_l), 0) ds} & \text{otherwise} \end{cases}$$

We integrate the shortest distance between two curves along the arc length s of the shorter curve, and divide the integration by the length of the shorter curve. In some cases, two curves run along each other for most of the arc length but differ significantly over a short section. In this case, we keep both because of their significant difference. D therefore only measures the significant distance between two curves, not their average distance.

We drop a trajectory if its distance from a selected trajectory is too small.

The algorithm for selecting a representative set from all trajectories is included as Algorithm 1. The values of t_l , t_a , and t_s are three thresholds set on the above metrics.

3.2.4 Creating tubes for trajectories

The trajectory of a streamtube represents part of the major eigenvector field extracted from the diffusion tensor field. The streamtube can also carry more of the information provided by the diffusion tensor. We use ellipsoid-shaped cross-sections to represent the medium and minor eigenvectors. One disadvantage of this approach is that these two eigenvalues can vary a lot between different regions, resulting in large cross-sections which could block other streamtubes and thus limit the number of streamtubes we could put in a scene. We address this problem by normalizing the medium eigenvector to a constant length to keep the cross-section from getting too big, yet we retain the ratio between the medium and minor eigenvalues. The mesh construction process is illustrated in Figure 1. Points are sampled on the cross-sections and then they are connected regularly


```

input: trajectory set  $U$ 
FOR each  $u \in U$  DO
  IF  $\text{length}(u) < t_l$  or  $\int c_l(u)du < t_a$  THEN
    erase  $u$  from  $U$ ;
  END IF
END FOR
WHILE  $U \neq \emptyset$  DO
  select  $v \in U$  with  $\max(\text{length}(u) \times \int c_l(u)du)$ , put  $v$  in  $V$ 
  FOR each  $u \in U$  DO
    IF  $\text{dist}(u, v) < t_s$  THEN
      erase  $u$  from  $U$ .
    END IF
  END FOR
END WHILE
output:  $V$ 

```

Algorithm 1: Trajectory selection algorithm.

to form a mesh. To avoid distortion, we calculate the binormal for each vertex on a streamtube, pick the sample point on the cross-section that is closest to the binormal, and use it as the starting point for generating the triangle list.

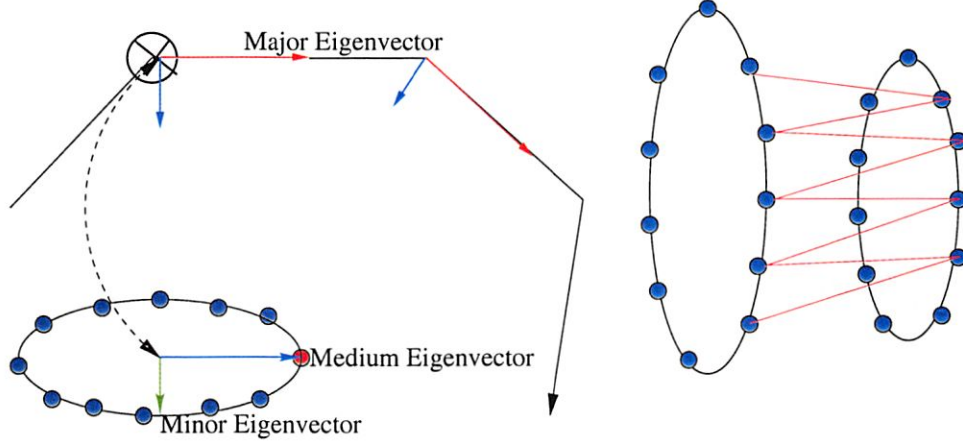


Figure 1: Generating mesh from trajectory and cross-section

Color on the streamtube is determined by the value of c_l . Color ranges from white, representing a minimum c_l value 0 to saturated red representing a maximum c_l value 1.

3.3 Planar anisotropy representation (Streamsurfaces)

Diffusion tensors with planar anisotropy could be the result of a surface structure, a boundary between different materials, or an intersection between two linear features. Visualizing planar anisotropy regions helps to illustrate these kinds of structures. However, a linearly continuous visual icon such as the hyperstreamline or streamtube is inappropriate for regions of this nature. Therefore, we introduce the streamsurface.

The streamsurface is the approximation of the integral surface in both the major eigenvector field and the medium eigenvector field. Given a starting point in the volume, we expand a streamsurface following these two vector fields. First, we generate many streamsurface to represent as many details as possible. Then a set of them are selected for display. Colors are mapped to the surfaces to represent the values of c_s .

3.3.1 Surface generation method

From a high level, we integrate to extend a starting point. This generate a set of points that surround the initial point and a set of triangles connecting them. We then repeat for each point until we hit termination criteria. The pseudo code is given in Algorithm 2.

```

input: Starting point  $p_i$ 
 $P$ : List of points to be extended
 $Q$ : List of points in the final surface
 $T$ : List of triangles in the final surface
 $P \leftarrow \{p_i\}, Q \leftarrow \{p_i\}$ 
WHILE  $P \neq \emptyset$  DO
  FOR each  $p \in P$  DO
    extend  $p$  to a set of new points  $S$ 
     $P \leftarrow S$ 
     $Q \leftarrow S$ 
    FOR each  $s \in S$  DO
      Attach  $s$  to the existing surface by generating triangles  $t$ .
       $T \leftarrow t$ 
    END FOR
    delete  $p$  from  $P$ 
  END FOR
END WHILE
output:  $Q, T$ 

```

Algorithm 2: Surface mesh generation algorithm.

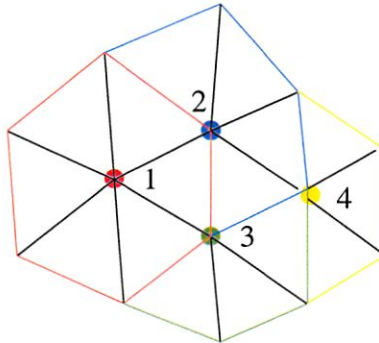


Figure 2: Generating surface mesh

The key steps are extending points and connecting them together in triangles. Figure 2 gives an idea of the surface tracking process. Now, we explain these steps and the extending condition:

- *Extending a point*—For every point, the algorithm maintains a list of existing triangles attached to it.
 1. Project these triangles to the plane perpendicular to the minor eigenvector, and draw a unit circle on the projection plane centered at the current point. Calculate the segments of arc on the unit circle that are not covered by those existing triangles.
 2. Mark points on the uncovered arc to divide it into equal segments; the number of divisions is chosen to make each segment of the arc be as near as possible to $\frac{\pi}{3}$.
 3. Extend in the direction of each marked point. This confines the extension to the plane defined by the marked point and the minor eigenvector of the original point. At each point in this plane, there is a vector defined by the crossing of this plane and plane expanded by the first two eigenvectors of the diffusion tensor at that point. The extension is performed in this two-dimensional vector field using the Runge-Kutta method.
- *Connecting points*—Extended points are connected to their adjacent points on the projection plane.
- *Extending condition*—Since the surface is used for representing regions of high planar anisotropy, the vertices on the surface should be restricted to these regions. Also, the extension should be clipped to the data volume and to high signal-to-noise regions.

The surface we describe may not be well defined. We believe that if we calculate the bracket $[X, Y] = XY - YX$ of each point in our tensor field using the major and medium eigenvectors as X and Y , and the result lies within the plane spanned by the vectors, then the surface is well defined. We are still investigating whether this is true.

3.3.2 Starting points sampling

Our choice of initial starting points for generating streamsurfaces is similar to that used for streamtubes. We loop through every sample point in the dataset, ignore sample points that lie within the voxel space of any point on a previously generated surface.

3.3.3 Surface selection

Similar to streamtube selecting, we use the following three criteria in surface selection:

- *Area of surface*—We favor relatively large pieces of surface, which represent large continuous surface structures.
- *Average planar anisotropy*—This value indicates whether the surface is constructed within a region of high planar anisotropy region.
- *Similarity to selected surfaces*—If a surface is too similar to a selected surface, we consider it a duplicate and get rid of it. The distance between two surface is defined as

$$D = \begin{cases} 0 & \text{if } \iint_{\Omega} \max(\text{sgn}(\text{dist}(\omega) - t), 0) d\omega = 0 \\ \frac{\iint_{\Omega} \max(\text{dist}(\omega) - T_s, 0) d\omega}{\iint_{\Omega} \max(\text{sgn}(\text{dist}(\omega) - T_s), 0) d\omega} & \text{otherwise} \end{cases}$$

Similarly to the definition of distance between curves, this metric gives the average of the minimum above threshold distance between two surfaces. The surfaces will be deemed different by this metric if they differ significantly over some area.

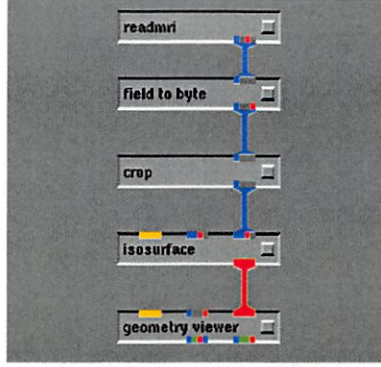


Figure 3: AVS network for generating anatomical features geometries as isosurfaces from T2-weighted images

Thresholds are set on each of the three values to filter out less representative surfaces.

3.3.4 Color scheme

Color on the streamsurface is mapped to planar anisotropy values. Color ranges from white which represents planar anisotropy value 0 to saturated green which represents planar anisotropy 1.

3.4 Anatomical landmarks

Based on the feedback from preliminary results, we found that biologists tended to explore the image more efficiently if they can easily identify some obvious large structures that they are familiar with in the image. For example, if they are looking at an image of the human brain, knowing the position of the skull, eyes, and ventricles make it easier to find structures in the rich information provided by our diffusion tensor visualization method. For this reason, we provided anatomical landmarks in our image. These landmarks are created by generating isosurfaces from T2-weighted images using AVS[11]. The AVS network is shown in Figure 3.

Water has a higher T2-weighted intensity than white matter and gray matter; white matter and gray matter, in turn, have higher T2-weighted intensities than air. So, we can generate an approximate boundary between fluid and other tissue that will echo the shape of the ventricle by generating isosurfaces at a certain level between the intensity of water and other tissue. While these shapes are not precise without more robust classification and segmentation, their general shapes makes the images much easier to interpret.

4 Results

We applied our method to a human brain dataset and a mouse embryo dataset. The human brain dataset has $256 \times 256 \times 40$ voxels, with a resolution of $0.89 \text{ mm} \times 0.89 \text{ mm} \times 3.2 \text{ mm}$. We interpolated and resampled the data isotropically in all 3 dimensions. The resampled dataset has $256 \times 256 \times 144$ voxels and the resolution is $0.89 \text{ mm} \times 0.89 \text{ mm} \times 0.89 \text{ mm}$. Table 2 shows the important parameters used in visualizing the human brain dataset.

Figure 4 shows the human brain from the eye to the top of the head. Through the semi-transparent skull surface, we can see the ventricle (colored blue) in the middle, and the streamtubes both around it and through the space. The two bulging spheres at the bottom of picture show the

positions of eyes. According to our feedback, these are the most useful landmarks in this dataset. User feedback indicates that the corpus callosum is depicted by the streamtubes running near the top of the ventricle. The corpus callosum contains almost all the neurons that cross the brain from one side to the other, which is illustrated by the paths of these streamtubes. The internal capsule is also shown in the picture. This is more or less the second most obvious white matter structure that is visible on a myelin stain. The internal capsule contains a huge number of neurons; it is the major tract that carries information between the cortex and the brainstem. The cerebral peduncles are the continuation of the internal capsule as it runs down into the midbrain. The white matter disappears off the bottom of the image as it descends into the brainstem; this is going to become the pyramidal tract, which carries basically all the descending motor information, and is therefore an obvious bundle of white matter at all points along its length.

The green surfaces in the picture show the laminar structures in the tissue where water diffusion is constrained in the surface.

We know from feedback that “in general, the strategy of revealing the ventricle surface along with the streamlines worked extremely well; the utility of these images went up by orders of magnitude with the ventricles as landmarks”. However, in our method, we cut off the ventricle in a box to get rid of a great number of small ventricles. Our feedback suggests that a more complete representation of the ventricles would help identify structures in the image.

Streamtube length	> 26.70 mm
Average linear anisotropy	> 0.40
T_l used in D for streamtubes	1.34 mm
Distance between lines	> 44.5 mm
Surface size	> 7.00 mm ²
Average planar anisotropy	> 0.30
T_s used in D for streamsurfaces	1.34 mm
Distance between surfaces	> 8.9 mm

Table 2: Parameters used in Figure 4 and Figure 5

Figure 5 shows another view of the human brain image. The user can interactively change the view.

Parameters should be chosen so that enough geometries are placed in the scene to represent the dataset, but not too many so as to generate the visual cluttering and impair the interactivity. To explore the parameter space, we vary each parameter and generate a set of pictures for the human brain dataset. From these pictures, we observe that relaxations on thresholds result in more geometries. But the main structures in the human brain could be observed with the initial setting of parameters in Table 2. We plan to work with biologists to experiment more on the effect of the parameter changes.

We also run our method on an embryonic mouse dataset; parameters are shown in Table 3.

Result is shown in Figure 14

In these pictures, we can see the surfaces around the skin of the embryo, suggesting that the boundary between different materials could result in planar anisotropy.

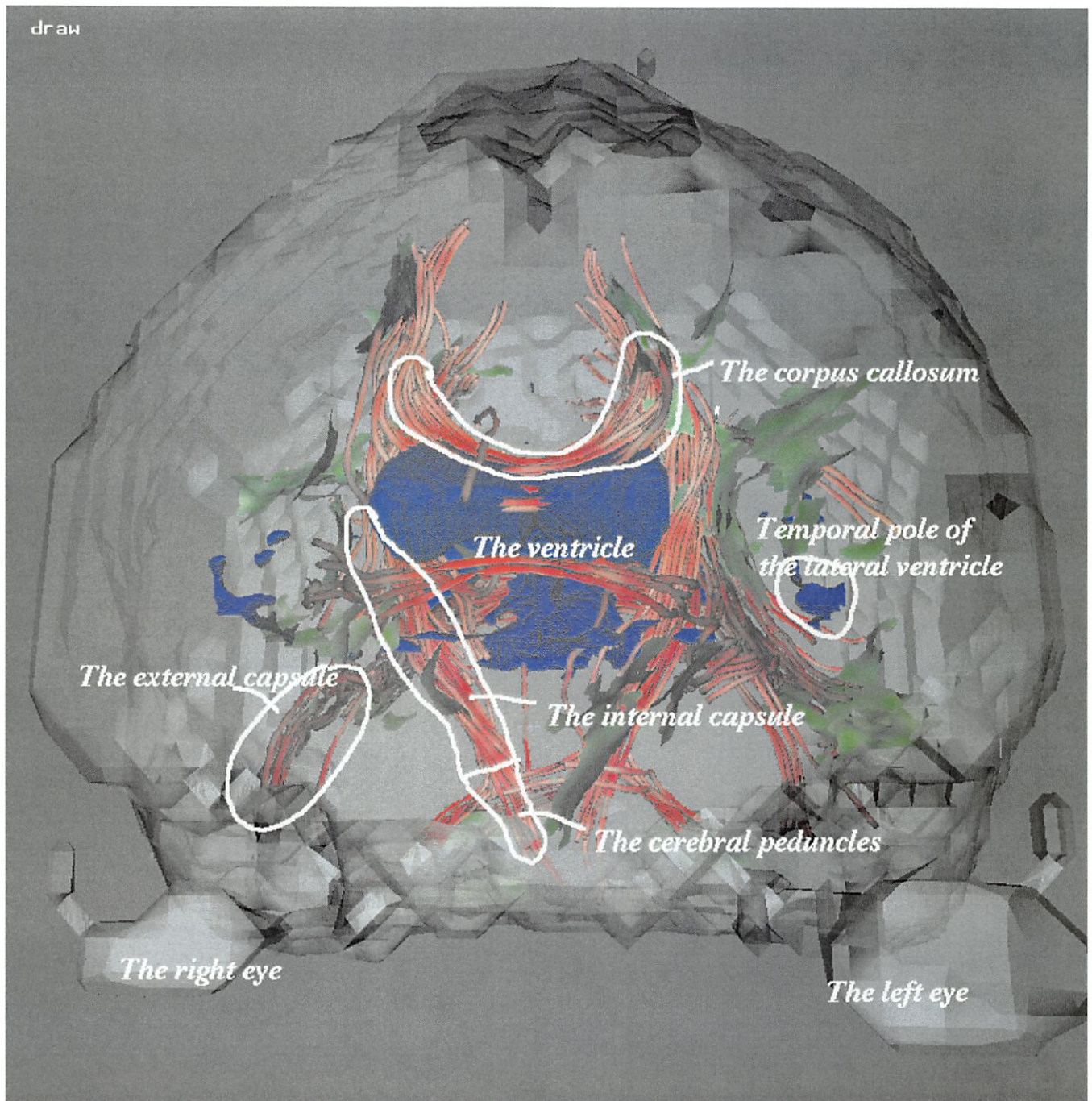


Figure 4: Human brain image, generated with parameters listed in Table 2

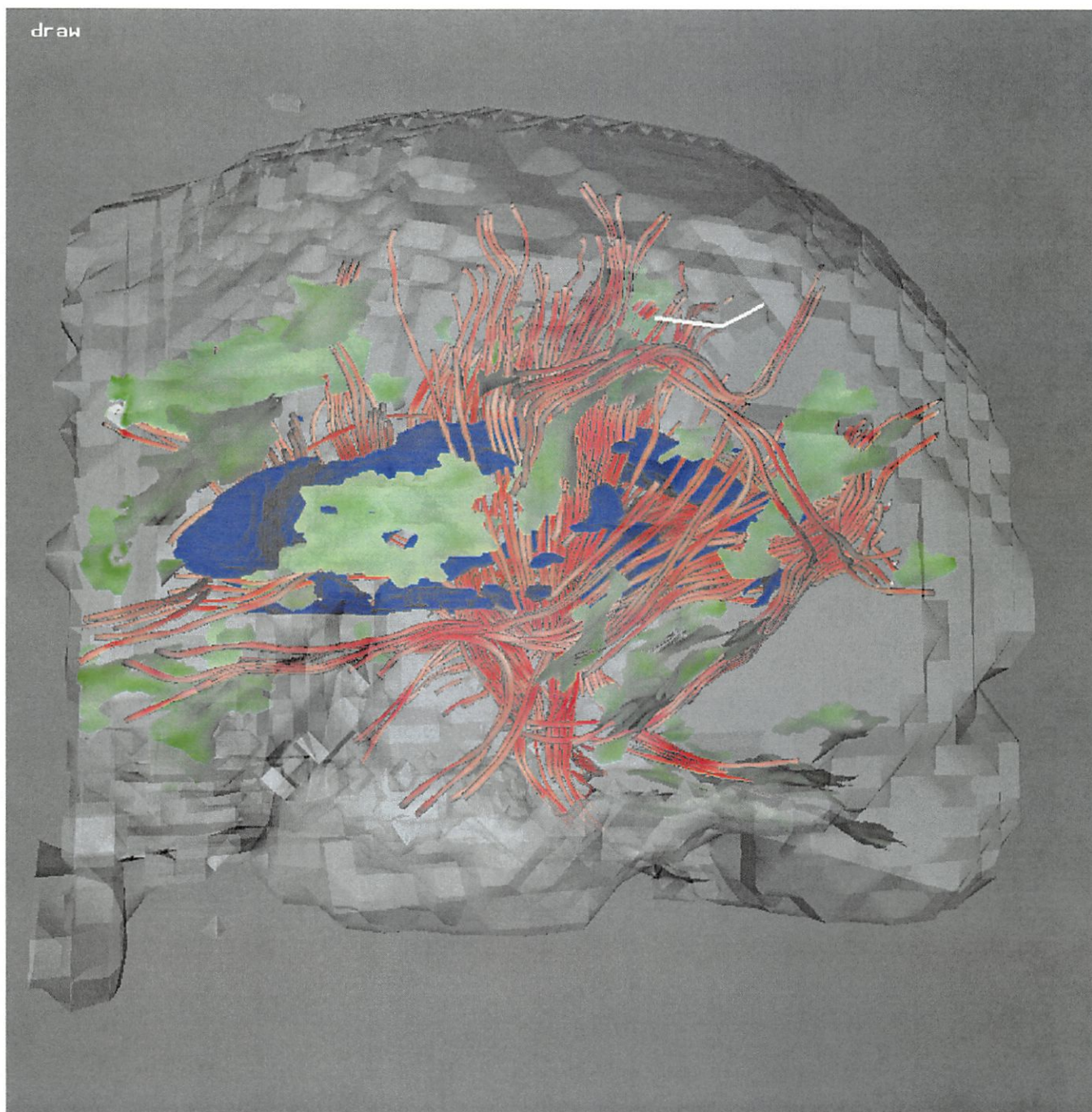


Figure 5: Another view of human brain image of Figure 4

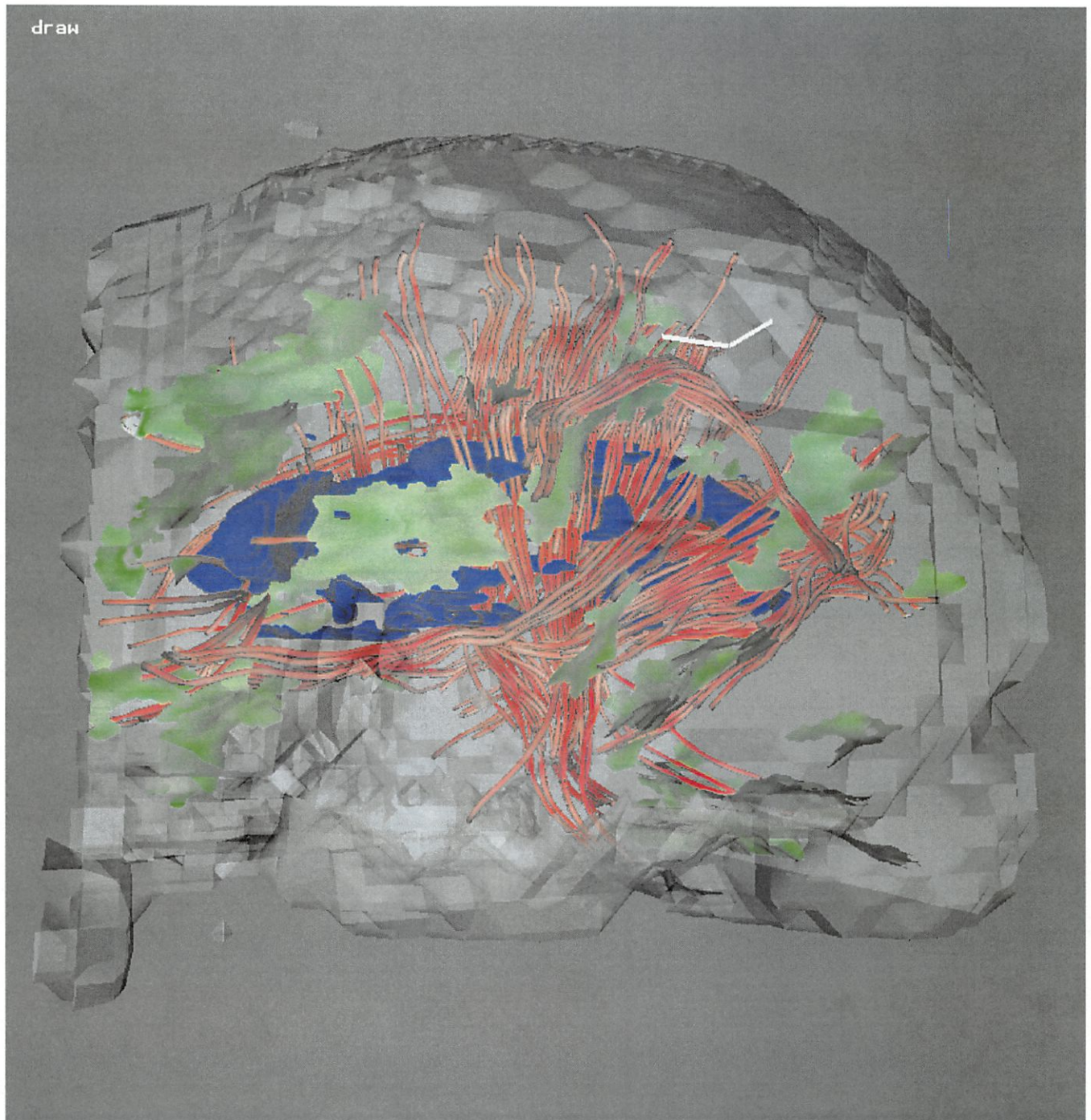


Figure 6: Human brain image, with one parameter changed in Table 2: average linear anisotropy > 0.20

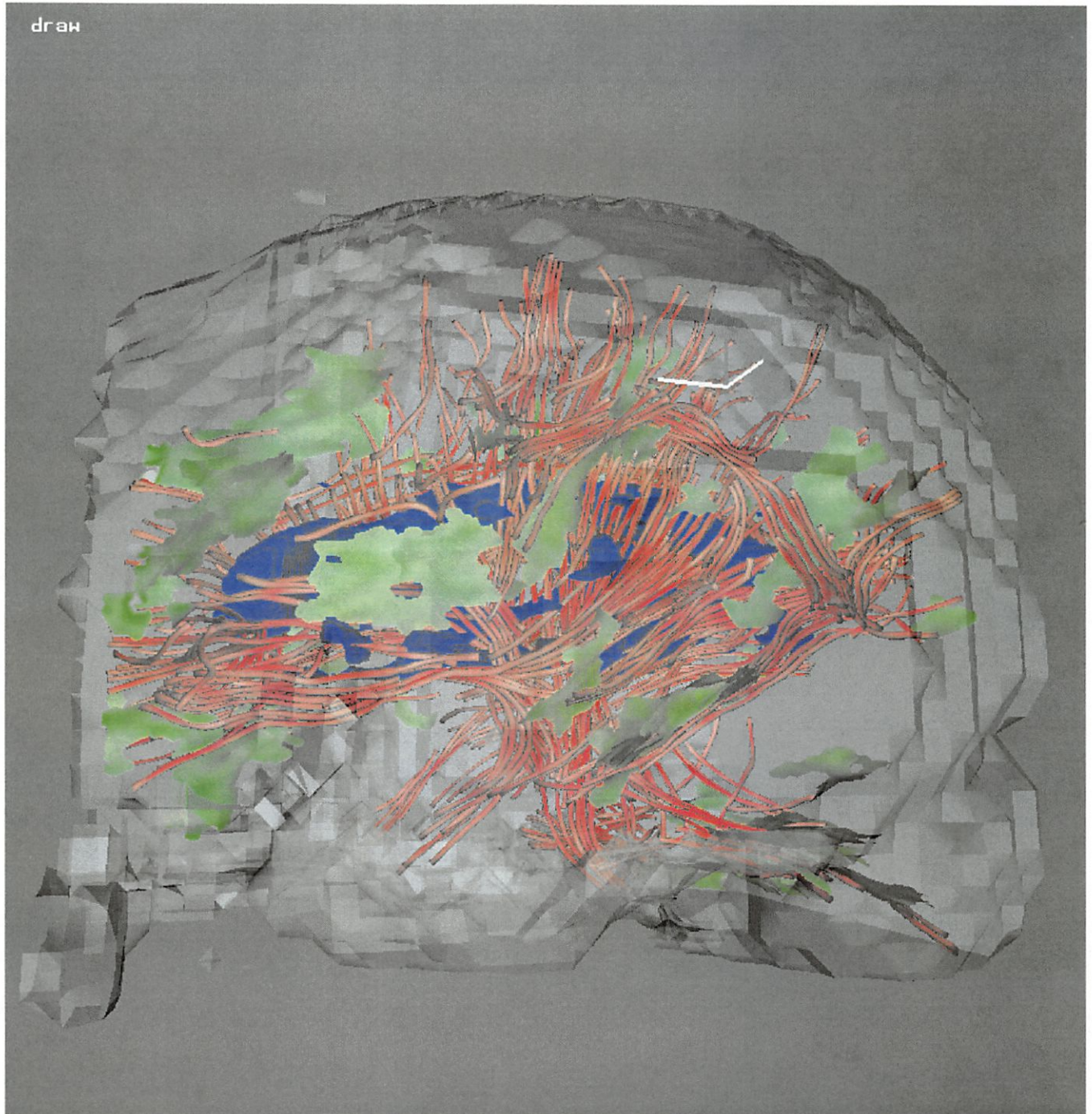


Figure 7: Human brain image, with one parameter changed in Table 2: line length > 8.9 mm

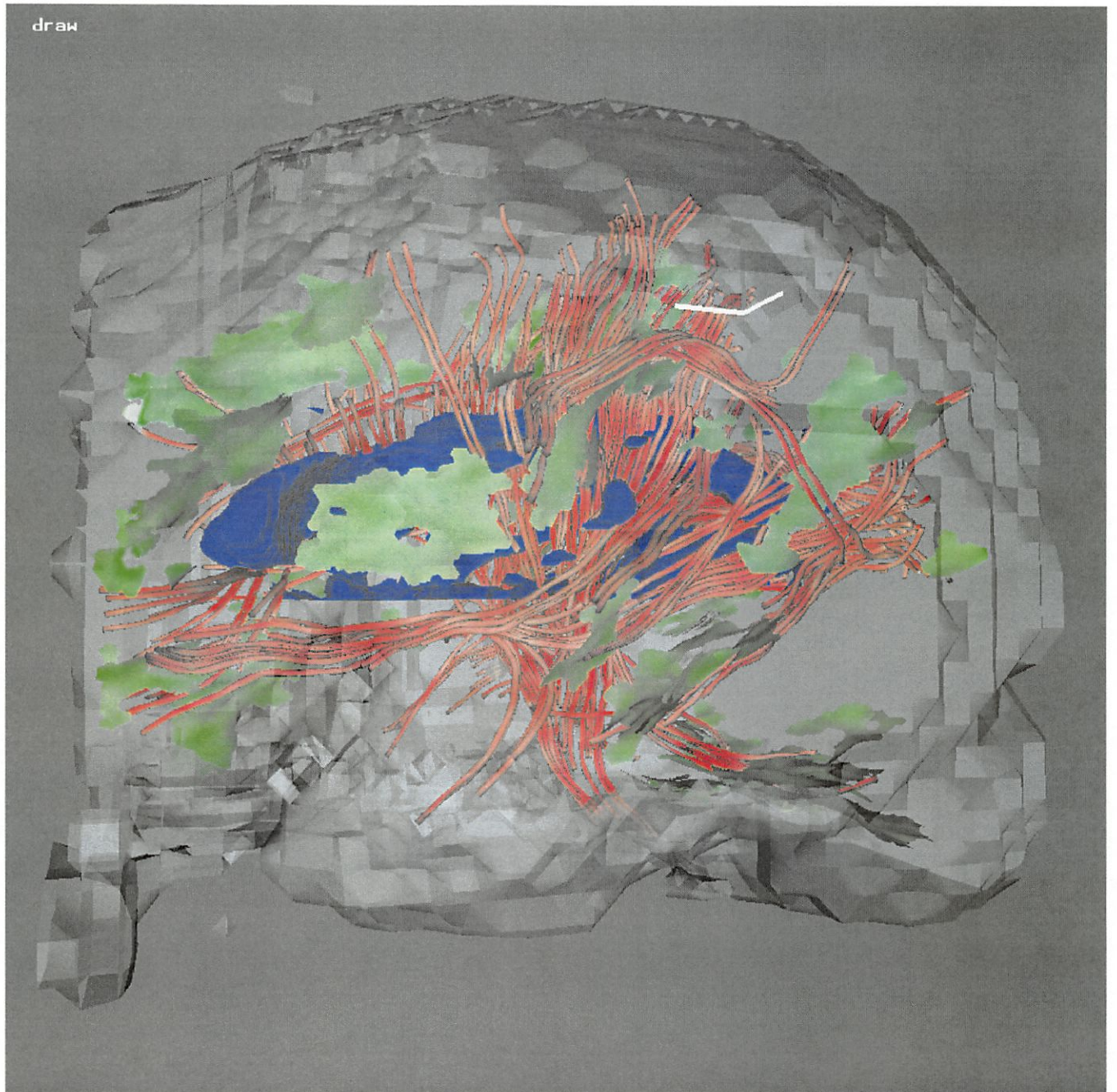


Figure 8: Human brain image, with one parameter changed in Table 2: $T_l = 2.68$ mm

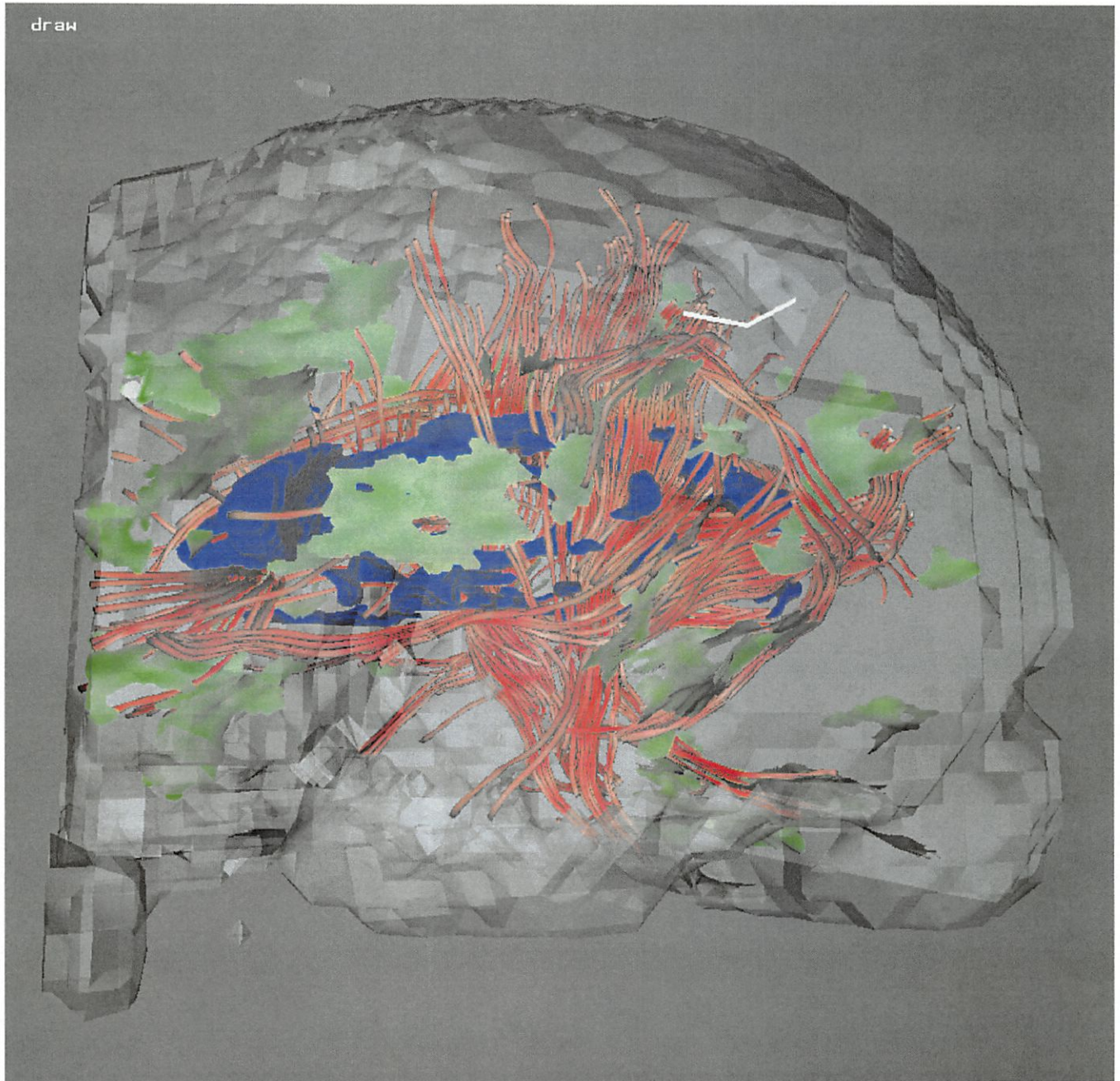


Figure 9: Human brain image, with one parameter changed in Table 2: distance between lines > 26.70 mm

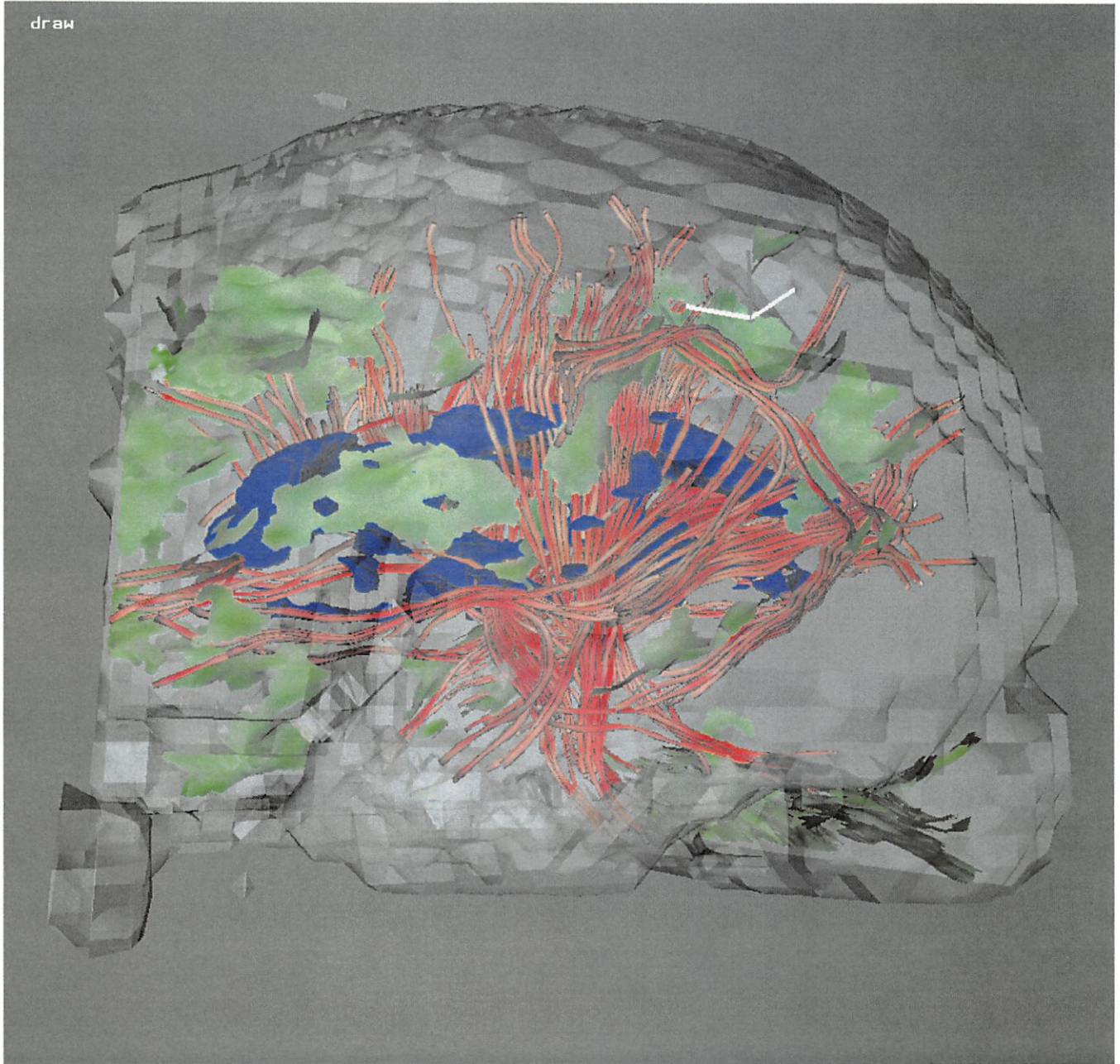


Figure 10: Human brain image, with one parameter changed in Table 2: surface sizes $> 3.5 \text{ mm}^2$

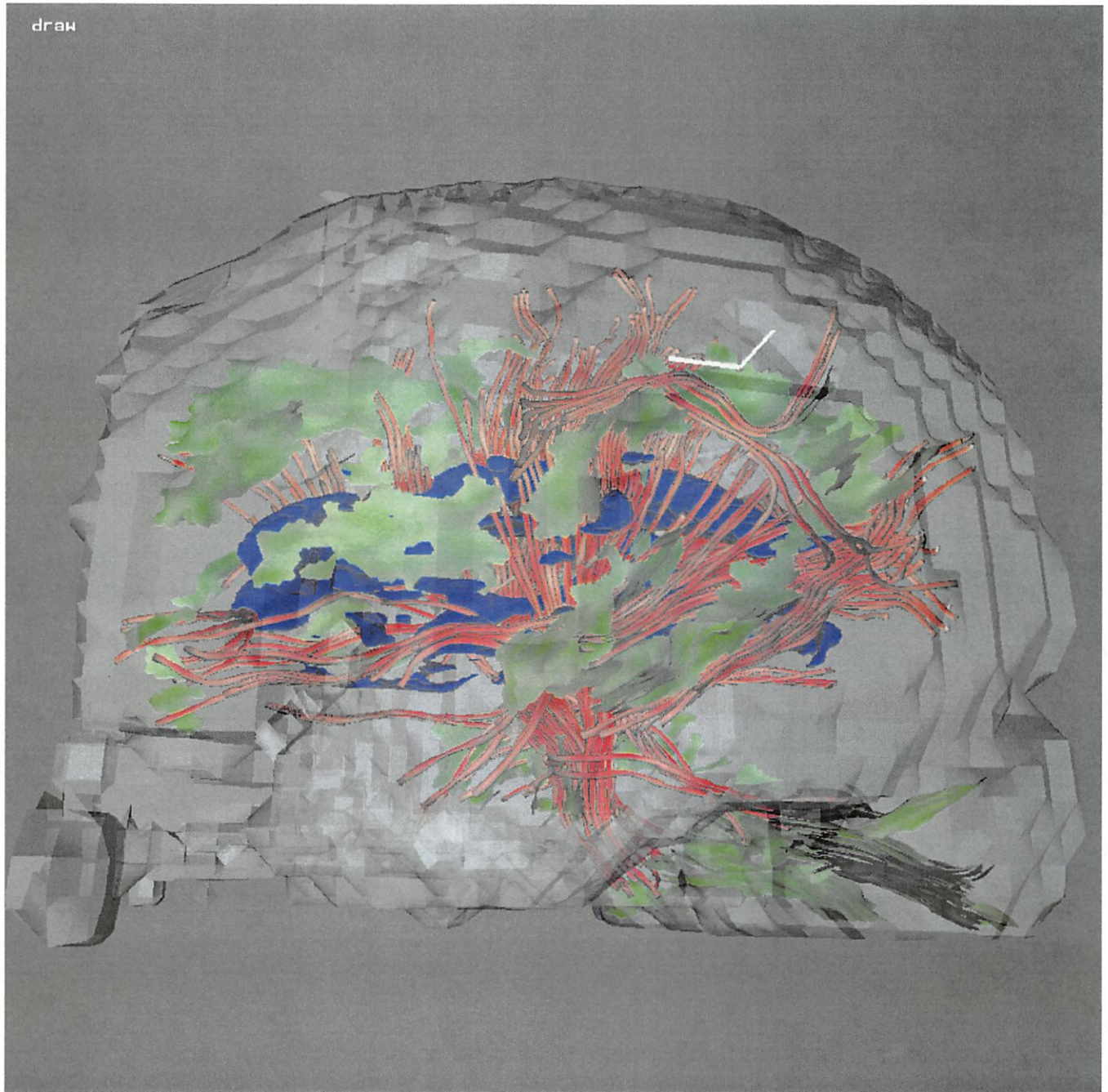


Figure 11: Human brain image, with one parameter changed in Table 2: average planar anisotropy > 0.20

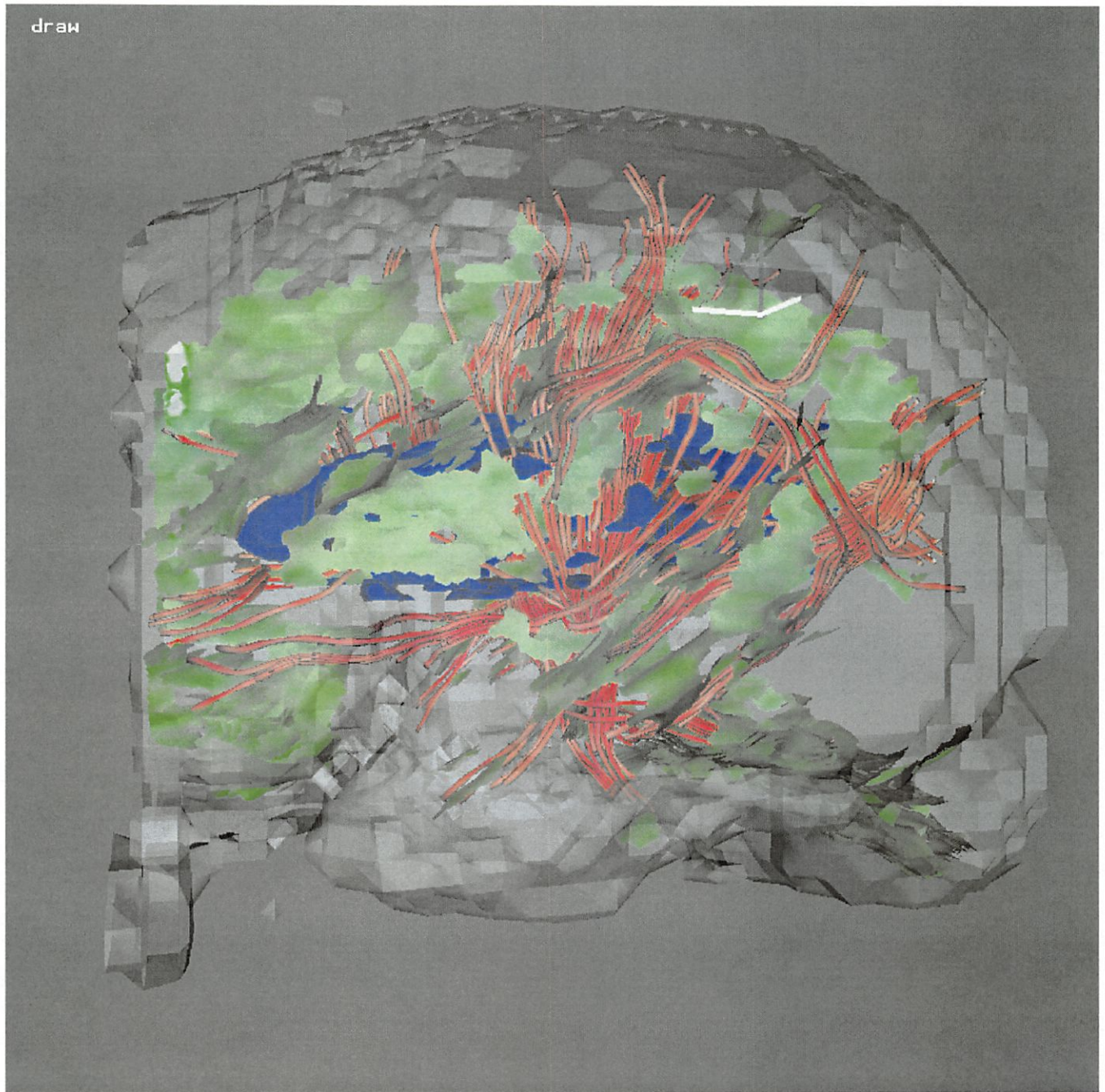


Figure 12: Human brain image, with one parameter changed in Table 2: distance between surfaces > 4.5 mm

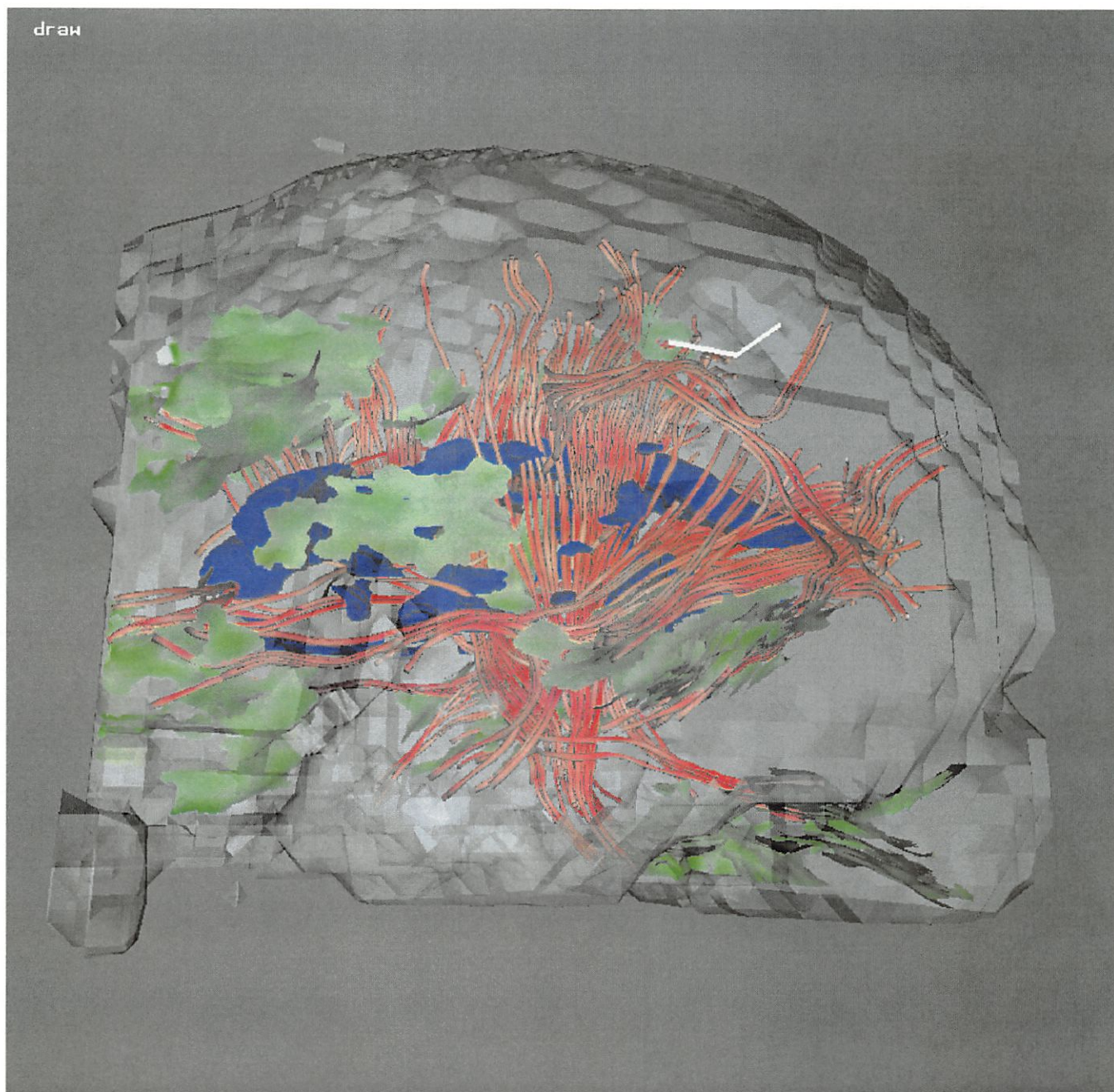


Figure 13: Human brain image, with one parameter changed in Table 2: $t > 2.68$ mm

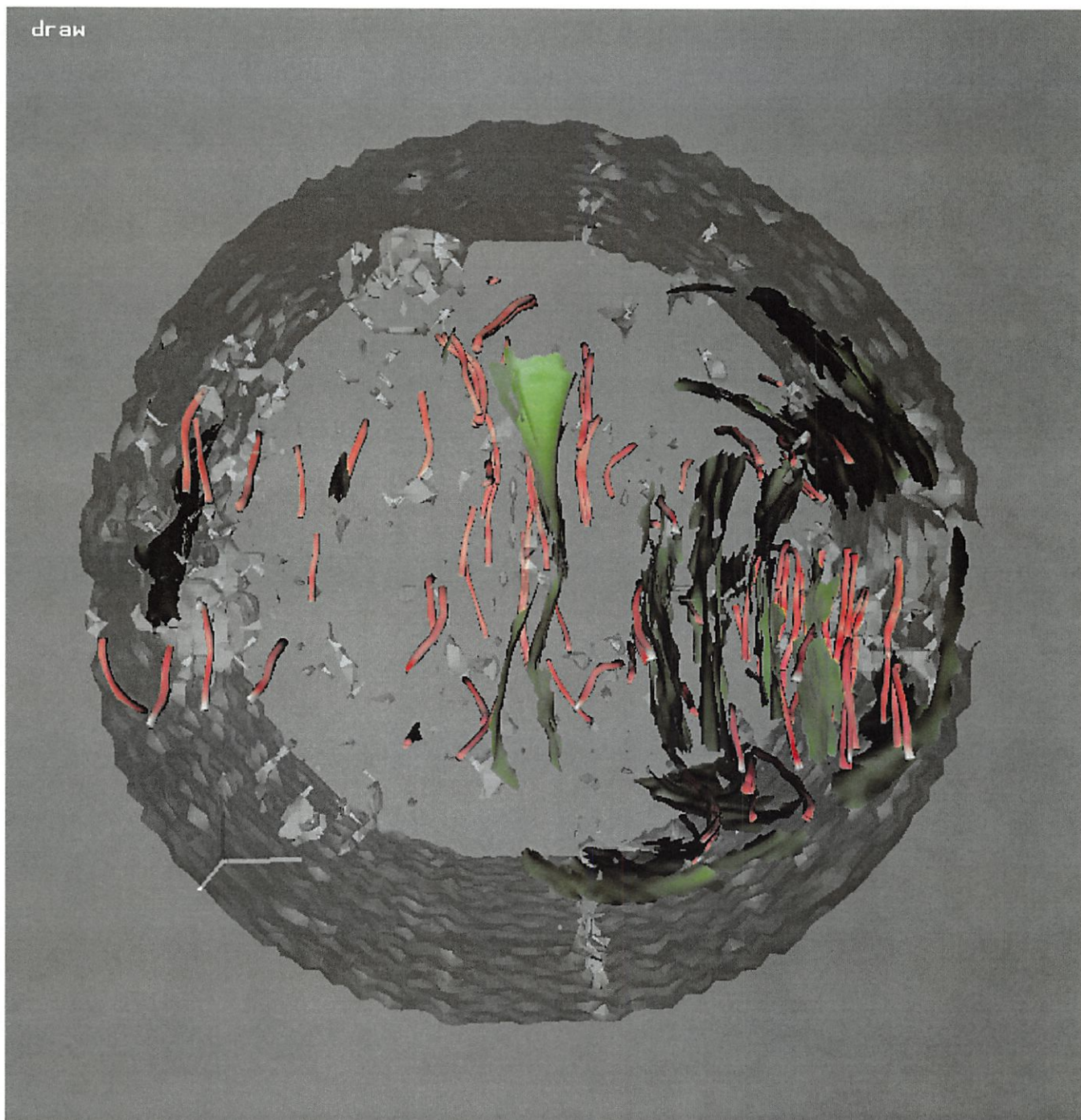


Figure 14: Result: Mouse embryo image, generated with parameters listed in Table 3

Line length	$> 0.84 \text{ mm}$
Average linear anisotropy	> 0.30
T_l used in D for streamtubes	0.03 mm
Distance between lines	$> 0.63 \text{ mm}$
Surface size	$> 0.00028 \text{ mm}^2$
Average planar anisotropy	> 0.30
T_s used in D for streamsurfaces	0.03 mm
Distance between surfaces	$> 0.21 \text{ mm}$

Table 3: Parameters used in Figure 14

5 Conclusion and future work

In this project, we present a new method of visualizing second-order diffusion tensor fields. We separate different kinds of anisotropy and map them to different visual icons, streamtubes and streamsurfaces. We generate a set of visual icons and select from them a representative set for display. The visual icons are generated in mesh format, facilitating user interaction with the object. Also, anatomical landmarks are generated for the purpose of orienting and positioning for the dataset. The final images show the paths of fibrous structures running through high linear anisotropy region, which illustrate the connectivity between different parts of the tissue. User feedback also shows that anatomical context and interactivity will help user explore the image effectively and efficiently. More and higher resolution landmarks are needed to further assist interpretation of the image.

In the future, we would like to verify our algorithm for generating streamsurfaces. We would like to generate more landmarks, and also make the generation fast and interactive. Also, we would make more experimentation with different parameter choices. And we would like to get more feedbacks from biologists, revise our work to help them better understand the images.

6 Acknowledgments

Thanks to my advisor David Laidlaw for all the support in this project. Thanks to Dr. Susumu at Johns Hopkins University who kindly sent us the human brain dataset and Eric Ahrens for the mouse embryo dataset. Also thanks to Dan Morris, David Berson, Eric Ahrens, Russell Jacobs, Seth Ruffins, and Rusty Lansford for providing feedback and evaluation on our results. And thanks to Cindy Grimm Dan Morris, and David Tucker for help in reviewing this report.

References

- [1] David H. Laidlaw, David Kremers, Eric T. Ahrens, Matthew J. Avalos. *Applying Painting Concepts to the Visual Representation of Multi-Valued Scientific Data*
- [2] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Russell E. Jacobs, and Carol Readhead. *Visualizing Diffusion Tensor Images of the Mouse Spinal Cord*. In IEEE Visualization 98, pages 127-134, 1998.
- [3] T. Delmarcelle and L. Hesselink. *Visualization of second order tensor fields and matrix data*. In IEEE Visualization 92 Proceedings, pages 316-323, 1992.

- [4] Gordon Kindlmann and David Weinstein. *Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields*. IEEE Visualization '99.
- [5] David Weinstein, Gordon Kindlmann, Eric Lunderg. *Tensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields* IEEE Visualization '99.
- [6] C. Pierpaoli and P.J. Basser. *Toward a Quantitative Assessment of Diffusion Anisotropy*. Magnetic Resonance Magazine, pages 893-906, 1996.
- [7] Rong Xue, Peter C.M. van Zijl, Barbara J. Crain, Meiyappan Solaiyappan, and Susumu Mori. *In Vivo Three-Dimensional Reconstruction of Rat Brain Axonal Projections by Diffusion Tensor Imaging* Magnetic Resonance in Medicine Volume 42, Number 6, December 1999.
- [8] C. F. Westin, S. Peled, H. Gubjartsson, R. Kikinis, and F.A. Jolesz. *Geometrical diffusion measures for MRI from tensor basis analysis* Proceedings of ISMRM, 1997.
- [9] M. Zöckeler, D. Stalling, and H-C. Hege. *Interactive visualization of 3d-vector fields using illuminated streamlines*. IEEE Visualization 96 Proceedings, pages 107-113, 1996.
- [10] J. Cormans, R. Luypaert, F. Verhelle, T. Stadnik and M. Osteaux. *A METHOD FOR MYELIN FIBER ORIENTATION MAPPING USING DIFFUSION-WEIGHTED MR-IMAGES* MAGNETIC RESONANCE IMAGING, pages 443-454, 1994
- [11] Craig Upson, Thomas Faulhaber, Jr., David Kamins, David Laidlaw, David Schlegel, Jeffrey Vroom, Robert Gurwitz, and Andries van Dam. *The application visualization system: A computational environment for scientific visualization*. IEEE Computer Graphics and Applications, 9(4):30-42, July 1989.