CEIG - Spanish Computer Graphics Conference (2012)
Isabel Navazo and Gustavo Patow (Editors)

Uncertainty Visualization of Brain Fibers

P. Hermosilla[1] and R. Brecheisen[2] and P.-P. Vázquez[1] and A. Vilanova[2]

[1]MOVING Group, Universitat Politècnica de Catalunya, Spain
[2]Eindhoven University of Technology, Nederlands

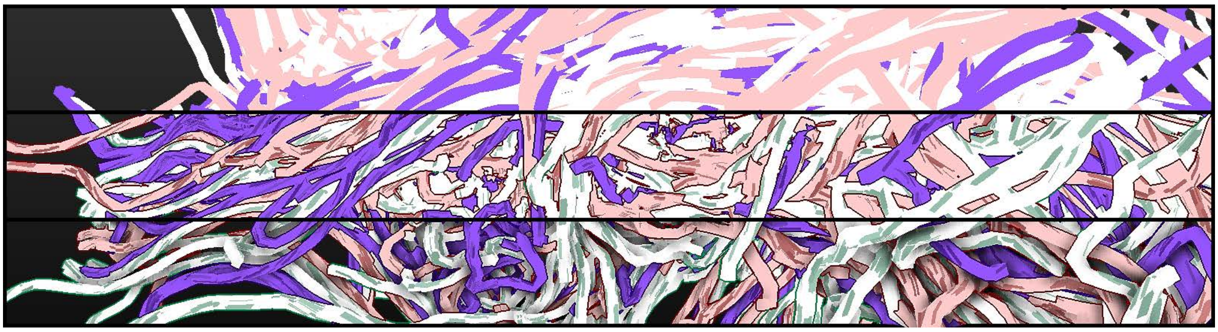


Figure 1: *Illustration of uncertainty in fiber visualization. The top image shows simple coloring based on the uncertainty level, the center image adds texturing in the direction of the fiber and outlines. Finally, the bottom image also shows ambient occlusion.*

Abstract

Diffuse Tensor Imaging (DTI) is an acquisition method based on Magnetic Resonance (MR) that provides information on the white matter fiber pathways in the living human brain. Such knowledge is crucial for understanding the way different parts of the brain work and how they interact with each other. The reconstruction of fiber tracts, however, depends on a number of parameters that introduce a degree of uncertainty in the data. Together with the parameter setting, other elements such as noise, motion, partial volume effects, or image artifacts increase the uncertainty. Therefore, fiber tracking algorithms may produce misleading results. Visualizing such uncertainty is important to avoid taking wrong decisions in medical environments. In this paper we present a set of techniques that provide a better understanding on the visualization of brain fibers by means of textures, silhouettes, ambient occlusion, and animation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms

1. Introduction

DTI fiber tracking is a powerful technique for illustrating how different parts of the brain are connected and how they interact. Unfortunately, its application in clinical practice is still limited. One of the main reasons is the lack of understanding what the diffusion measurement actually means. Although it has been assumed for long time that fibers are aligned with their diffusion directions, it has not been completely validated yet. A second important reason is the amount of uncertainty that burdens fiber tracts: apart from the complexities of the capture process, that may add noise or other imaging artifacts, the reconstruction process is subjected to the selection of several parameters that lead to different results.

The *classical* visualization of fiber tracts using lines or streamtubes usually does not show this uncertainty in data and as a consequence gives the illusion that there is no un-

DOI: 10.2312/LocalChapterEvents/CEIG/CEIG12/031-040

certainty, which may result in a dangerous assumption when the information is further use for surgery planning.

In this paper we present a set of illustrative techniques for the easy visualization of the uncertainty in the fiber data sets. We build upon the work by Brecheisen *et al.* [BPtHRV11], and we extended their DTITool with several visualization algorithms that improve the perception of brain fibers and its uncertainty by applying different visual cues such as textures, silhouettes, ambient occlusion, and animation. As a consequence, more certain fiber tracts are rendered with more eye catching effects such as faster animations and higher contrast colors.

2. Previous Work

Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) [BML94] generates images of the diffusion of water in the brain. However, these images contain artifacts in the form of noise, motion effects and other, that create a certainty degree of uncertainty for fiber tracking algorithms [BPP*00, CMW*06].

Although many efforts have been devoted to DTI visualization, significantly less research has been addressed to the important problem of the visualization of uncertainties generated through the different stages of the DTI processing pipeline. There are different techniques tailored to visualize uncertainty in different areas such as vector fields [BWE05, LPSW96], geospatial information [MRH*05], surfaces [GR04], and even radiosity simulations [PF96]. Pang *et al.* [PWL97] surveys different methods for the visualization of uncertainty in several data types such as surfaces, illumination simulation, flow visualization, and so on.

For the concrete case of fibers, Jiao *et al.* [JPS*10] analyze the uncertainty using different metrics and visualize the differences directly on the fibers by adding some custom widgets that can be directly manipulated by the user. A different approach is taken by Jiao *et al.* [JPGJ12] for the visualization of HARDI data using glyphs. In this case, the uncertainty is shown by superimposing a set of glyphs that represent different levels of uncertainty. For higher uncertainty values, those are rendered with a more transparent and less saturated color. Note that, the higher the uncertainty, the larger the covered volume, and therefore, these glyphs are visible because they bound the more certain glyphs that occupy smaller volumes.

Brechesein *et al.* [BPVtHR09] investigate the uncertainty produced by parameter modifications by generating a sensitivity a streamline superset that covers the whole parameter space of stopping criteria, and then using selective culling to display only specific streamline collections. Similar visualization approaches for multi-valued data are due to Doleish *et al.* [DGH03] and Wenger *et al.* [WKZL04], although they are not specifically focusing on parameter sensitivity. In a further work, Brecheisen *et al.* [BPtHRV11] provide several illustrative motifs for the visualization of different confidence ranges.

Rick *et al.* analyze the uncertainty in probabilistic fiber tracts [RvKC*09]. For the visualization, they use simple color change to identify such low probability areas (assigning dark transparent colors to them) and highly probable tracts (that are represented using bright and opaque colors). This technique is further analyzed by Kapri *et al.* [vKRC*10] in order to assess if it clearly communicates visually the uncertainty in different fiber tracts while maintaining interactivity. They conclude that the users would prefer to have direct control over the visualized sets, and that also would like to have a higher amount of contextual information.

Our approach takes into account the results by Kapri *et al.* [vKRC*10]. We provide a higher amount of contextual information, and we give the user a larger amount of flexibility, since she can determine several factors such as the coloring of the fiber tracts and has control over the amount We also provide realtime framerates, shown to be crucial for the tools to be effective ([vKRC*10]).

3. Illustrative Visualization of Uncertainty in Fiber Tracts

In this section we present our approach. First, we detail the requirements and then we give an overview to our implementation. The following sections provide more detail on the implementation of the different effects.

3.1. Requirements

Following the approach by Kapri *et al.* [vKRC*10], we set the following requirements:

- The synthesized images must covey the uncertainty in an intuitive way, that is, users must infer from the renditions which groups of fibers have less uncertainty.
- It is necessary that the spatial arrangements of the fibers is obvious or can be deduced from the visual cues present in the images.
- Real time inspection is compulsory.

In order to fulfill those requirements we incorporate several effects to our renderings: First, we add silhouettes and texturing to the fibers, with higher contrast on the groups of fibers with lower uncertainty. Second, we also compute an ambient occlusion term to improve the 3D perception of the elements in the scene. Finally, we add other cues such as texture animation in order to make more certain fibers to catch the eye attention.

3.2. Visual effects

One of the most important issues in illustrative visualization is the *effective communication* of the features we want to enhance. While other authors improve the 3D perception of neural fiber tracts by the use of line illumination [YDX11], in our case, we want the user to be able to visually distinguish between different levels of uncertainty. We also want to add visual cues that facilitate the proper perception of the overall distribution of fibers in our model. In order to do so, we use two different visual techniques tailored to *classify fibers* and to better convey their *3D position and orientation*.

3.2.1. Visual classification of uncertainty ranges

We are concerned with the easy distinction of the different ranges, therefore, we need to provide visual cues that help the users to distinguish both the position and orientation of the fibers and emphasize the transitions between uncertainty levels. We also want to help the user's attention to focus on fibers with less uncertainty, and we want to provide an enhanced communication of the overall shape of the fibers in the scene. This is achieved by adding a set of visual effects:

- *Color and saturation:* The human eye is very sensitive to color and luminance changes. Therefore, the color will be our basic element for the visual classification of the uncertainty of the fiber tracts. When the user choses a base color for an uncertainty range, it will be used both for tinting the fibers of that range, and for adding other color cues such as texturing and silhouettes. It is well known that saturated colors attract the attention to a higher level than unsaturated colors. Thus, we saturate the picked colors selected for the fiber paths with less uncertainty.
- *Silhouettes:* Silhouettes convey shape. A good way to enhancing the perception of fiber direction and boundaries is to render them together with the geometry. We add silhouettes to the fibers but with the same color than the uncertainty range. This reinforces the position and direction of the fibers of each range. We also add silhouettes between different uncertain ranges to make easily distinguishable the range boundaries.
- *Textures:* We use textures to provide further visual information on the direction of the fiber paths. In order to do so, we generate 3D random textures that are then applied with a sweep in the direction of the fiber path which results in lines in the direction of the path. The random textures store a grey level that is converted onto different levels of the base color inside the corresponding uncertainty range color. Since the fiber color will have a high *value* (from HSV color model), the *stripes* generated using this technique will show a lower *V value*.
- *Animation:* Another important visual effect that attracts user gaze is animation: Quick movements catch eye's attention at a higher degree than slow movements, and both of them are more effective to get the user's attention than no movement. We animate fiber tracts by modifying the fiber color, and this is carried out at different speeds depending on the uncertainty level: less uncertainty implies higher speed. We do this by dynamically modifying the color along the fiber tract. Visually, it is like a white spot moving all along the fiber at a certain speed that depends on its uncertainty level. In order to avoid clutter and a large number of movement patterns, we only animate up to two fiber tract ranges, corresponding to the two more confident fiber tract groups.

3.2.2. Shape perception

Fiber visualization often results in cluttered views because of the high number of fiber tracts and their similarity to each other. These visualizations can be improved with the use of expressive techniques such as halos [MHEI09]. We have added three extra visual cues tailored to better communicate fibers' shapes and orientations as well as their position in space:

- *Silhouettes:* As we have already mentioned, silhouettes are an important cue to provide visual information on the boundaries and orientations of fibers. We have implemented a simple silhouette detection algorithm based on the approach by Saito and Takahashi [ST90], but also adding silhouettes between different uncertainty ranges.
- *Ambient Occlusion:* It is a technique that simulates the indirect light contribution to each point of the scene. Its origin lies on the *obscurances* approach by Zhukov *et al.* [ZSK98]. The main idea is to calculate (e. g. [RMSD*08, Ste03]) or to estimate (e. g. [TCM06, Mit07, FM08]) the amount of occlusion to ambient light produced (mainly) by nearby geometry.
- *Unsharp masking:* Unsharp masking techniques improve the perception of small features by increasing the local contrast [LGK06]. We have implemented unsharp masking as a postprocess for enhancing the perception of the boundaries between fibers of different uncertainty levels.

3.3. Algorithm Overview

Before the rendering can start, we need to prepare the input data, as depicted in Figure 2. This is carried out as a preprocess with the following steps:

- *Geometry:* In this step, we generate the geometry that will be used to effectively render the fiber tracts. As input, we have a line set with an uncertainty value per line. For the geometry generation, we first group all fibers with the same uncertainty level. Then, we generate rectangular prism per fiber, and each vertex is assigned a vector with the fiber direction and an integer with the uncertainty value.
- *Texture:* In order to enforce the perception of fibers' directions, we texture them with stripes. We use an auxiliary 3D texture with some texels in different levels of gray, and the remaining texels in white. To avoid a large memory consumption, the texture is stored as a 2D texture.
- *Color:* Once the fibers have been created, we assign a different color for each uncertainty value. We let the users to select the base color, but, in order to improve perception,

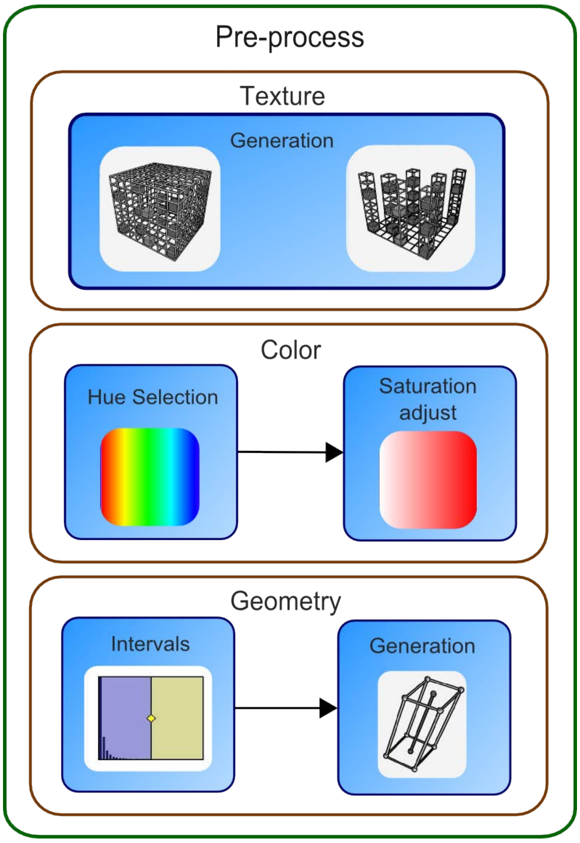


Figure 2: *Preprocessing steps.*

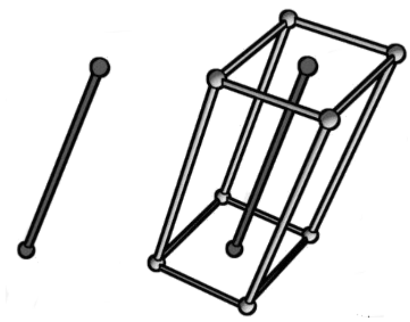
we keep the hue value selected by the user is used, but the saturation is changed, giving high saturation to higher confident ranges and low saturation to uncertain values.

The rendering stage performs several steps at each frame that will be detailed later. Each step requires rendering either the fiber geometry or a simple quad covering the whole viewport and they implement the different algorithms in a modular fashion so that we can toggle on and off each effect at any moment. These major rendering stages are:

- Geometry pass: This stage generates partial information required for the final composition. It generates the texture values, id of the uncertainty range, depth, fiber direction, and distance to the beginning of the fiber (for the animation).
- Ambient occlusion: This process calculates ambient occlusion. In order to do so, we calculate an approximation of the occlusion factor per pixel in the scene.
- Final composition: This final rendering step is used to compose the final image. It performs the following tasks:
  - Sweep: The fragment shader uses the fiber direction to query the buffer of the previous render step and sweep the visited pixels in order to build lines in the fiber direction.
  - Silhouette: Silhouette is computed and it is painted with the color of the closest uncertainty range.
  - Color and animation: With the stored id, we assign the base color to the fiber. For the first or two first levels of uncertainty (lower uncertainty values), we generate a simple animation that changes the color of the fiber (to white) along the fiber with the time. Final color is modulated with the occlusion.
  - Unsharp masking: In order to improve the visual contrast between different uncertainty ranges, we also perform a final unsharp masking of the generated image.

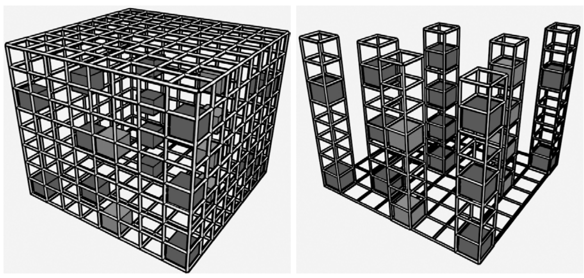
4. Preprocess

As already stated, the input data must be prepared before starting the rendering stage. In this preprocess we perform two steps: a) 3D model generation from the line set, and b) Pseudo-random texture creation for fiber texturing



(a) Original fiber (b) New geometry

Figure 3: *Geometry generation around the original fiber segment.*



(a) Original 3D texture (b) Equivalent 2D texture

Figure 4: *Compacting a 3D texture in a 2D texture.*

The input data is a set of lines, each one formed by a set of segments, and the attributes that indicate their uncertainty level. Since these fibers do not have volume, we *generate a 3D model* from those by building an oriented prism with 4

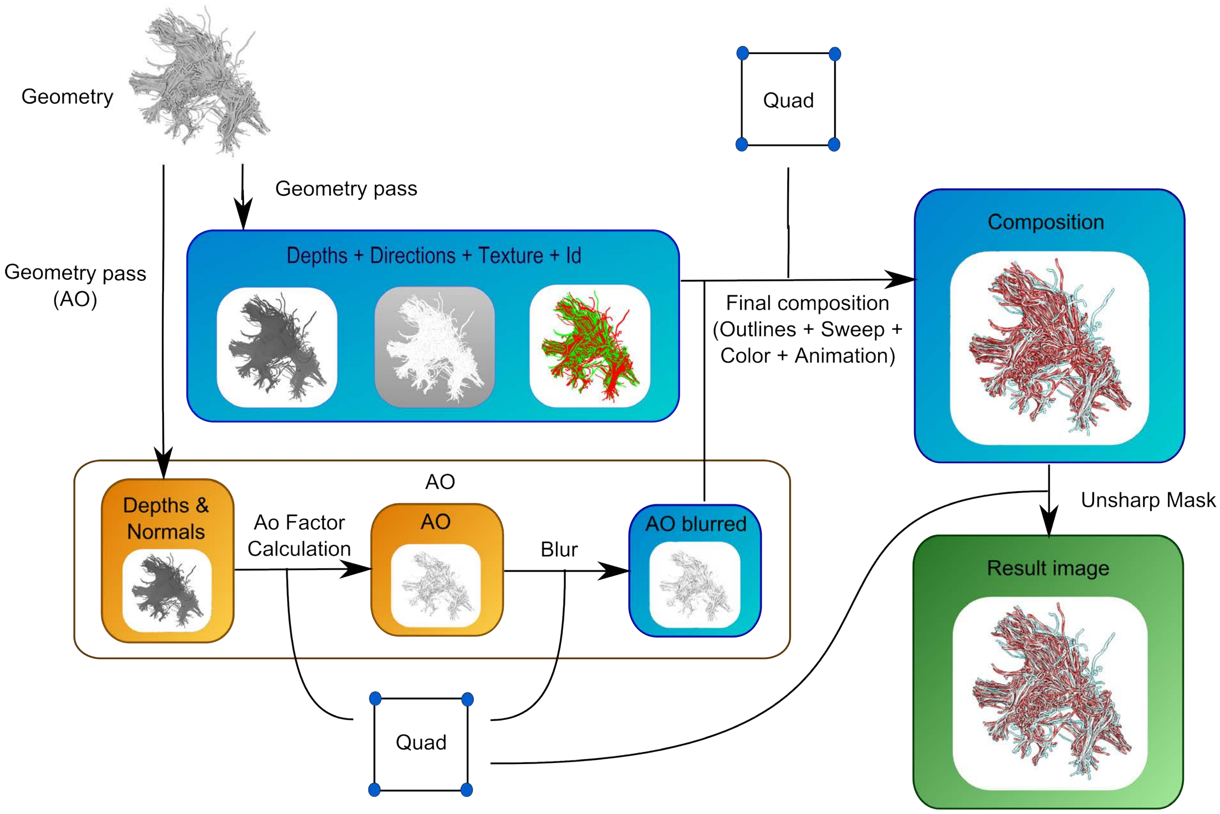


Figure 5: *Overview of the rendering algorithm.*

quads around the original fiber as its faces, and another couple of quads closing the top and bottom parts. The original segment will indicate the direction of the fiber, necessary for the posterior texturing. This process is depicted in Figure 3.

In order to texture the interior of the fibers, we *create an auxiliary 3D random texture* stored as a 2D texture (see Figure 4). Initially, all texels of the 2D texture contain white. Then, we randomly select a set of texels that are filled with a random grey level. A grey color in the (x,y) 2D position corresponds to the color of the column (x,y) in the 3D texture. When we have the pixels of the 2D texture defined as grey or white we randomly define the distance between two consecutive dots in this column and the size of the grey dots. This information is stored as follows in the texture:

- *Red channel*: Stores the grey level (between 0.0 and 1.0).
- *Green channel*: Stores the size of each grey point. Size is encoded in a way that value 0.0 corresponds to the minimum size of the axis and the bounding box of the model, and 1.0 is the maximum value.
- *Blue channel*: Stores the difference between each grey point throughout the column, encoded in the same way than the green channel.

The cost at rendering time to obtain the right values is negligible, and the memory savings are important.

5. Rendering

In this section we describe the process executed to draw a single frame. It is composed by several rendering steps, concretely, two of them are essentially geometry passes, that require rendering the whole geometry, and the other two are composition steps, that use only as input a quad covering the whole viewport. This is depicted in Figure 5.

5.1. Geometry pass

The first step consists in a rendering that populates the GPU memory with several auxiliary textures to be used in the following steps. This process renders the whole geometry at full screen size and generates a depth map, an image where the fibers are textured, a texture with the fiber directions, and another texture containing the ids of each level of uncertainty. To avoid wasting texture memory, the whole information is stored only into two auxiliary buffers.

5.2. Ambient Occlusion calculation

Ambient occlusion rendering improves the perception of depth as well as the spatial arrangement of the fibers. In fiber rendering, ambient occlusion has been used to guide the hatching when simulating hand-drawn illustration for the inclusion of context, concretely, they augment fiber stylized rendering with information of the brain, also using non-photorrealistic techniques [SEI10].

We implement a simple screen-space approach, the *classical* Screen Space Ambient Occlusion as developed by Crytek [Mit07] (SSAO) due to its simplicity. The algorithm has 3 steps: First, we store the depth of each pixel and the normal of the projected geometry at each fragment. In a second step, we use this information to compute the occlusion factor. Finally, we blur the occlusion factor in order to correct sampling artifacts. This is carried out in two separate (vertical and horizontal) passes.

The more complex step is the initial ambient occlusion factor estimation. We proceed the following way: For each pixel we obtain the position of that pixel in camera space using the depth value. Then, we visit a set of neighbors using a set of displacement random vectors previously computed that are stored in texture. These displaced new positions are then projected back and the stored depth values of those new pixels are compared with the depth of moved positions. With these depth differences, we obtain the occlusion factor.

The random sampling scheme used in this method generates noise. Thus, a further blurring is required ([Mit07]). We achieve this with two more render passes to blur the occlusion factors, taking into account the depth values to prevent mixing occlusion factors of distant and near geometry.

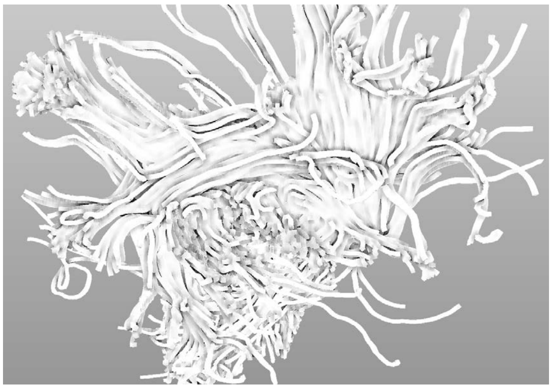


Figure 6: *Computed weights for the Ambient Occlusion calculation.*

This is a computationally expensive algorithm. To reduce its cost, we execute this algorithm using a buffer with $1/4th$ of the viewport size. To make the up-sampling we use a simplified version of the Bilateral Upsampling algorithm [SGNS07], where we get, for each pixel in the high-res image, the near value, from the low-res image, with the lowest depth difference. The values computed in this step (see Figure 6) are used to darken the final color.

5.3. Silhouette rendering

Silhouette helps to understand the fiber's shape. Therefore, we visually illustrate the extent of the fibers by adding outlines. Our algorithm is based in the approach by Saito and Takahashi [ST90]. Silhouettes are detected by analyzing the depth map. A fragment shader visits the neighboring pixels using a radial kernel and if strong depth differences between pixels are detected, the pixel is marked as belonging to a silhouette. For this silhouette detection algorithm, we do not need to render the whole geometry. On the contrary, we only render a quad covering the whole viewport.

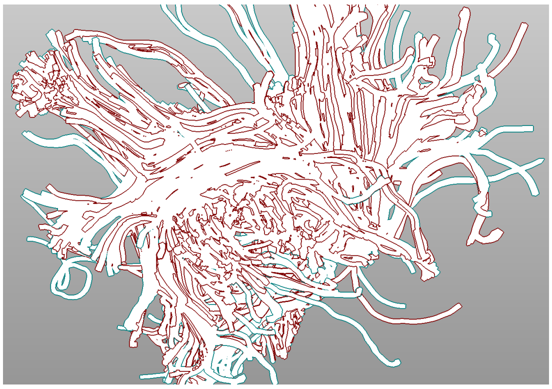
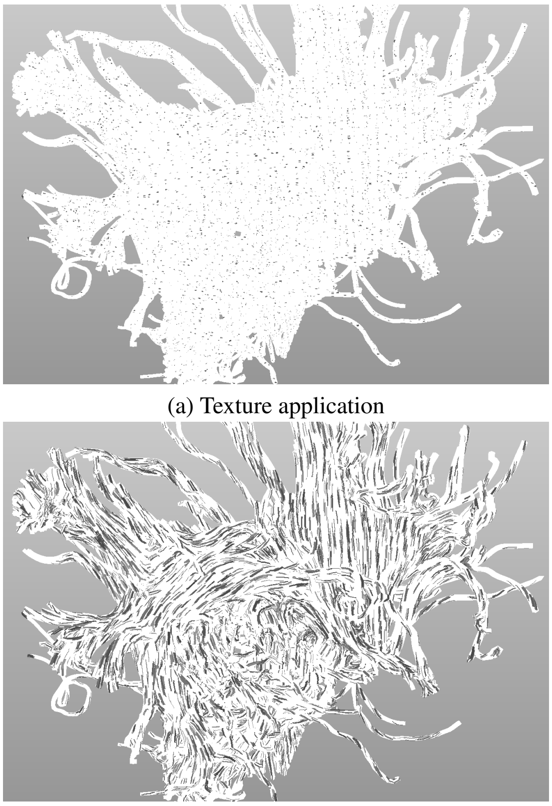


Figure 7: *Rendering outlines for fibers enhancing the external silhouette and the changes of uncertainty level in the interior of the model.*

Since the geometry step also generated an id texture, we can use it in order to determine uncertainty range changes in image space. In this case, we also label the pixel as silhouette. The result helps to identify the model shape, and it also permits a better visual distinction between uncertainty ranges. The results are shown in Figure 7. In contrast to the approach by Brecheisen *et al.* [BPtHRV11], our silhouettes are computed by taking into account depth and confidence change.

5.4. Final composition

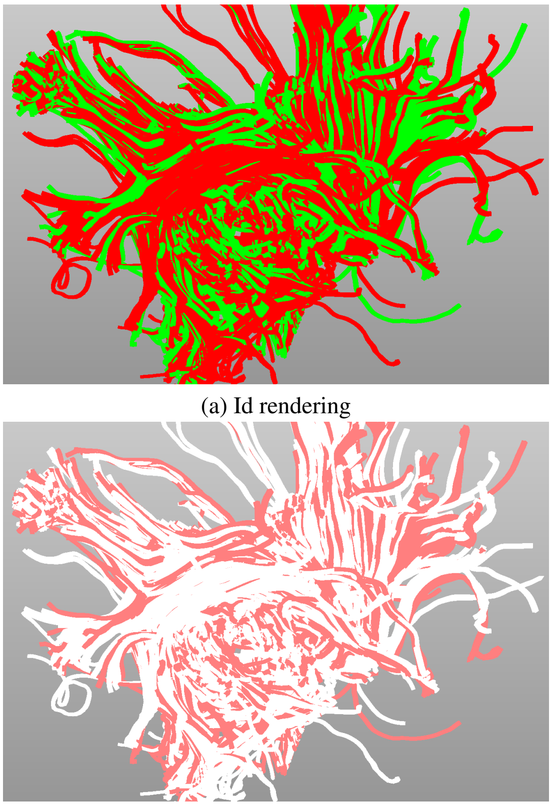
The interior part of the fibers is textured using a two-pass method: First, the values of the 3D texture are applied to each fiber by querying the corresponding value of the 3D texture using the fiber position, as shown in Figure 8-top. This step only requires rendering a quad covering the whole viewport. Then, we perform a sweep in screen space where the textured values are swept in the main direction of the fiber. Again, this can be accomplished by simply rendering a quad that covers the whole viewport. A simple fragment



(a) Texture application

(b) Sweep

Figure 8: *Fiber texturing in two passes.*



(a) Id rendering

(b) Base color

Figure 9: *Initial color application.*

shader analyzes fiber direction and performs the corresponding sweep (Figure 8-bottom). These swept colors are mixed with the uncertainty color (Figure 9-bottom), using the uncertainty range id (Figure 9-top) selected from the buffer of the first step. This second rendering pass is the same used for silhouette detection, and the result is mixed then with the ambient occlusion factor to get the final composition (Figure 10-top).

In this second rendering pass we also apply an animation to the two lowest uncertainty levels. This animation is intended to make the attention of the user to focus on these levels. The animation consists in generating a fading out region that paints the fiber in white and moving this region along the fiber direction at a certain speed (larger for more certain fibers). We can simulate this effect by storing in a auxiliary buffer the distance of each fragment to the first fiber endpoint. This is carried out in the first rendering pass. Then, in this composition pass, this distance and the time since application started are used to determine which zones will be modified. The colors of these zones are modified using the equations bellow. First we compute the modification factor that depends on the distance to the first endpoint of the fiber and the moving speed:

$$factor = (dist + time * CSpeed * ISpeed) \% CRange \quad (1)$$

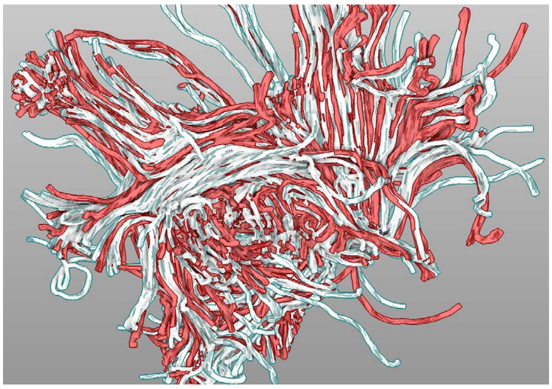
$$factor = \frac{2 \mid factor - CRange * 0.5 \mid}{CRange} \quad (2)$$

$$factor = \min\left\{\frac{2factor}{CFadeOut * ISpeed}, 1.0\right\} \quad (3)$$

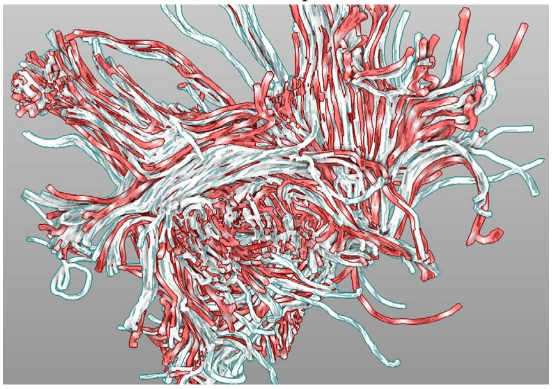
Then, the color is modified by using this factor:

$$color = (color * factor) + vec4(1.0 - factor) \quad (4)$$

where *CSpeed*, *CRange* and *CFadeOut* are constants that define the speed, distance between animated section and the animated section size. *time* is the time since the application started and *dist* is the distance to the first fiber endpoint. Finally, *ISpeed* is the uncertain range speed. Although the result of the animation can hardly be seen in an image, we show an example in Figure 10-bottom, that can also be seen in the accompanying video.



(a) Final composition



(b) Animated

Figure 10: *Composition and animation.*

5.5. Unsharp Mask

Contrast enhancement is a popular 2D image processing tool that improves the image appearance. Concretely, it helps users to understand complex models by emphasizing small features. Several experiments have proven that users prefer enhanced scenes over the non-enhanced ones [LGK06, IRS*09]. Unsharp masking increases high frequency components of a signal by adding back a filtered version of the signal in which high-frequency components are enhanced. In our case, we want the user to easily distinguish between fibers of different uncertainty levels. Unsharp masking is usually achieved through image post-processing. This is the approach we take. Like in other previous cases, we only need to render a quad covering all the viewport.

The color modification is performed in CIELab format. Once the sample point is evaluated, the resulting color is transformed to CIELab yielding a triplet $[L,a,b]$. We also sample the neighboring pixels and average these colors. The resulting color is another triplet, also in CIELab format: $[L_{Avg}, a_{Avg}, b_{Avg}]$. The unsharp color in CIELab ($U(C)$) is built using the following formula:

	Ranges			
Distance	**1**	**2**	**3**	**4**
Near	41.195	41.354	41.353	41.516
Medium	51.157	52.402	52.145	52.145
Far	59.960	59.968	59.969	59.966

Table 1: *Frames per second.*

$$U(C) = [L + (\lambda(L - L_{Avg}), ka, kb)], \quad (5)$$

where λ is the weight we give to the difference. If we use a high value, the image results in too overexposed, while a too low value does not emphasize the existing features. Like in [TLB*09], we add the k factor in order to guarantee that the saturation is preserved. The value k is computed as:

$$k = (L + \lambda(L - L_{Avg}))/L. \quad (6)$$

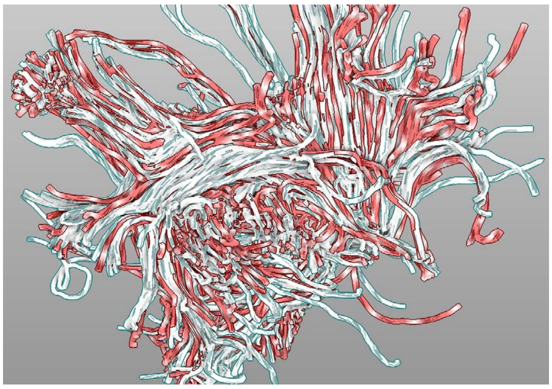


Figure 11: *Unsharp masking for enhancing local contrast.*

The results are shown in Figure 11. In this case, the image contains all the previously presented effects including animation. The unsharp effect enhances the perception of the silhouette edges.

6. Results

We can see how the different effects are applied to fiber tracts where two, three, and four different uncertainty levels are respectively shown in Figures 11, 12, and 13.

We show the framerates obtained in table 1. They were obtained using three different configurations: the first one shows the fibers at a close distance, thus covering most of the viewport, the second configuration shows fibers at a medium distance, so we can see the whole bunch of fibers. Finally, the *far* configuration moves the fibers off the observer and a good portion of the viewport is free. In all configurations, we use 4 uncertainty ranges and we obtain a framerate with a lower bound of, roughly 40 fps. The results were tested on a i7 Intel CPU at 3.26GHz equipped with 12GB of RAM memory and a NVidia GeForce GTX 570 with 1280MB of GDDR5 memory. Images were generated at a resolution of 1381×906. We also tested the application on lower range systems, such as a notebook, equipped with a i7 CPU and a NVidia GTX 540M GPU obtaining framerates over 15 fps (though with a larger resolution). Since the fps increase with the camera distance, the bottleneck is likely to be at the pixel shading stage.

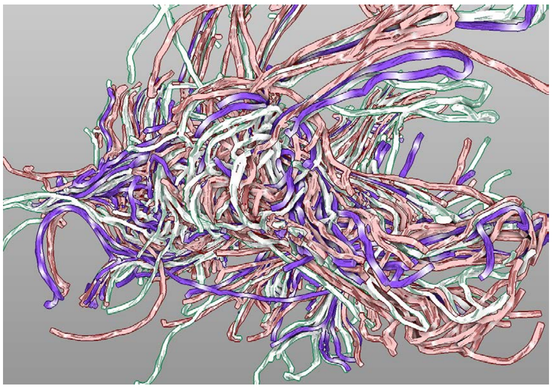


Figure 12: *Fiber rendering using 3 uncertainty ranges.*

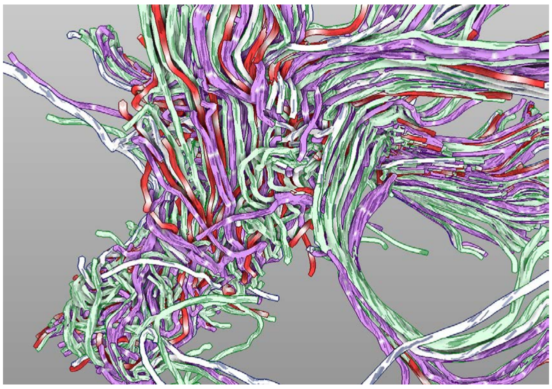


Figure 13: *Fiber rendering with 4 uncertainty ranges.*

7. Conclusions and Future Work

We have presented a rendering technique for brain fiber visualization that adds a set of non-photorrealistic techniques tailored to improve the awareness and perception of the user on the uncertainty of the different fibers. We worked upon the idea by Brecheisen *et al.* [BPtHRV11] but added different visual cues more oriented to perception such as texturing, animation, ambient occlusion, and unsharp masking. The main idea behind the different visual cues added to the fiber rendering algorithm is to guide the user's attention to the more important fibers of the model, the ones with higher confidence, and to aid in the process of distinguishing the different ranges by enhancing the boundaries. We built the rendering pipeline in a modular way, so that the user has complete flexibility in activating or deactivating any of the visual cues. She also has the possibility of changing the base colors of the different uncertainty levels and deciding which levels are to be shown. Despite the number of techniques included, that require multiple rendering passes, we achieve realtime framerates.

For the more confident fiber clusters, we have used rendering techniques that have been proved as methods that catch the user's attention (contrast, movement, and so on...). However, a further user study is compulsory to analyze the real value of the proposed techniques. We believe that the most important outcome to obtain is the set of techniques that allow for the fast determination on the most confident fiber bundles only from the visualization. For these studies both real and synthetic data can be used. On the other hand, the user study cannot be carried out with many people, since they should be specialists (medical researchers, surgeons, etc.). Other users could also be tested to increase the population, but the presence of specialists is compulsory. We expect the conclusions of the user study will help us improve the rendering effects if required.

Apart from the user study, there is still room for improvements. We have planned to modify the texturing process to make it independent of the projection size of the model. Moreover, other effects can also be tested, such as the use of transparency or blurring to focus the user attention to the more important fiber ranges and use the texturing process to create a pencil rendering illustration, that has been classically used in professional medical illustration. Pencil rendering could be used both for the fiber tracts or to enrich the renderings with contextual information [SEI10].

Acknowledgments

The authors want to thank the anonymous reviewers for their valuable comments. This paper has been supported by project TIN2010-20590-C01-01 of the Spanish Government.

References

[BML94] BASSER P. J., MATTIELLO J., LEBIHAN D.: Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of magnetic resonance Series B 103*, 3 (1994), 247–254.

[BPP*00] BASSER P. J., PAJEVIC S., PIERPAOLI C., DUDA J., ALDROUBI A.: In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine 44*, 4 (2000), 625–32.

[BPtHRV11] BRECHEISEN R., PLATEL B., TER HAAR ROMENY B., VILANOVA A.: Illustrative uncertainty visualization for DTI fiber pathways. In *EuroVis 2011 Poster* (2011).

[BPVtHR09] BRECHEISEN R., PLATEL B., VILANOVA A., TER HAAR ROMENY B.: Parameter sensitivity visualization for dti fiber tracking. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1441–1448.

[BWE05] BOTCHEN R. P., WEISKOP D., ERTL T.: Texture-based visualization of uncertainty in flow fields. In *IEEE Visualization 2005* (2005), pp. 647–654.

[CMW*06] CHENG P., MAGNOTTA V. A., WU D., NOPOULOS P., MOSER D. J., PAULSEN J., JORGE R., ANDREASEN N. C.: Evaluation of the gtract diffusion tensor tractography algorithm: a validation and reliability study. *Neuroimage 31*, 3 (July 2006), 1075–1085.

[DGH03] DOLEISCH H., GASSER M., HAUSER H.: Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the symposium on Data visualisation 2003* (Aire-la-Ville, Switzerland, Switzerland, 2003), VISSYM '03, Eurographics Association, pp. 239–248.

[FM08] FILION D., MCNAUGHTON R.: Effects & techniques. In *ACM SIGGRAPH 2008 classes* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 133–164.

[GR04] GRIGORYAN G., RHEINGANS P.: Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics 10*, 5 (Sept. 2004), 564–573.

[IRS*09] IHRKE M., RITSCHEL T., SMITH K., GROSCH T., MYSZKOWSKI K., SEIDEL H.-P.: A perceptual evaluation of 3d unsharp masking. In *Human Vision and Electronic Imaging XIV, IS&T/SPIE's 21st Annual Symposium on Electronic Imaging* (San Jose, USA, 2009), Rogowitz B. E., Pappas T. N., (Eds.), vol. 7240 of *Annual Symposium on Electronic Imaging*, SPIE, pp. 1–12.

[JPGJ12] JIAO F., PHILLIPS J., GUR Y., JOHNSON C.: Uncertainty visualization in hardi based on ensembles of odfs. In *Proceedings of the 5th IEEE Pacific Visualization Symposium (PacificVis 2012)* (February 2012), p. (accepted).

[JPS*10] JIAO F., PHILLIPS J. M., STINSTRA J. G., KRGER J., KOMMARAJU R. V., HSU E. W., KORENBERG J. R., JOHNSON C. R.: Metrics for uncertainty analysis and visualization of diffusion tensor images. In *MIAR* (2010), pp. 179–190.

[LGK06] LIN W., GAI Y., KASSIM A.: Perceptual impact of edge sharpness in images. *Vision, Image and Signal Processing, IEE Proceedings - 153*, 2 (April 2006), 215–223.

[LPSW96] LODHA S. K., PANG A., SHEEHAN R. E., WITTENBRINK C. M.: Uflow: visualizing uncertainty in fluid flow. In *Proceedings Visualization '96* (1996), pp. 249–254.

[MHEI09] MAARTEN H. EVERTS HENK BEKKER J. B. T. M. R., ISENBERG T.: Illustrative rendering of dense line data. In *IEEE Visualization 2009* (2009).

[Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *ACM SIGGRAPH 2007: Courses* (New York, NY, USA, 2007), ACM, pp. 97–121.

[MRH*05] MACEACHREN A. M., ROBINSON A., HOPPER S., GARDNER S., MURRAY R., GAHEGAN M., HETZLER E.: Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science 32*, 3 (July 2005), 139–160.

[PF96] PANG A., FREEMAN A.: Methods for comparing 3d surface attributes. In *In SPIE Vol. 2656 Visual Data Exploration and Analysis III* (1996), pp. 58–64.

[PWL97] PANG A., WITTENBRINK C., LODHA. S.: Approaches to uncertainty visualization. *The Visual Computer 13*, 8 (Nov 1997), 370–390.

[RMSD*08] ROPINSKI T., MEYER-SPRADOW J., DIEPENBROCK S., MENSMANN J., HINRICHS K. H.: Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Eurographics 2008) 27*, 2 (2008), 567–576.

[RvKC*09] RICK T., VON KAPRI A., CASPERS S., EICKHOFF S. B., ZILLES K., KUHLEN T.: Poster: Interactive visualization of uncertainty in probabilistic tractography of brain's white matter pathways as assessed by diffusion tensor imaging. In *Proc. IEEE Visualization* (2009).

[SEI10] SVETACHOV P., EVERTS M. H., ISENBERG T.: Dti in context: Illustrating brain fiber tracts in situ. *Comput. Graph. Forum 29*, 3 (2010), 1023–1032.

[SGNS07] SLOAN P.-P., GOVINDARAJU N. K., NOWROUZEZAHRAI D., SNYDER J.: Image-based proxy accumulation for real-time soft global illumination. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2007), PG '07, IEEE Computer Society, pp. 97–105. URL: http://dx.doi.org/10.1109/PG.2007.35, doi:10.1109/PG.2007.35.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90* (1990).

[Ste03] STEWART A. J.: Vicinity shading for enhanced perception of volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003* (Washington, DC, USA, 2003), IEEE Computer Society, p. 47.

[TCM06] TARINI M., CIGNONI P., MONTANI C.: Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 1237–1244.

[TLB*09] TAO Y., LIN H., BAO H., DONG F., CLAPWORTHY G.: Feature enhancement by volumetric unsharp masking. *Vis. Comput. 25*, 5-7 (2009), 581–588.

[vKRC*10] VON KAPRI A., RICK T., CASPERS S., EICKHOFF S. B., ZILLES K., KUHLEN T.: Evaluating a visualization of uncertainty in probabilistic tractography. In *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling* (2010), Wong K. H., Miga M. I., (Eds.), vol. 7625, SPIE, pp. 762534–762546.

[WKZL04] WENGER A., KEEFE D. F., ZHANG S., LAIDLAW D. H.: Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (Nov. 2004), 664–672.

[YDX11] YUAN Z., DAI F., XU D.: Improved visualization of fiber tracts in whole brain using illuminated line rendering. In *BMEI* (2011), Ding Y., Peng Y., Shi R., Hao K., Wang L., (Eds.), IEEE, pp. 328–332.

[ZSK98] ZHUKOV S. I. A., KRONIN G.: An ambient light illumination model. In *Eurographics Rendering Workshop 98* (1998).