Eurographics/ IEEE-VGTC Symposium on Visualization 2008
A. Vilanova, A. Telea, G. Scheuermann, and T. Möller
(Guest Editors)

Volume 27 (2008), Number 3

Virtual Klingler Dissection: Putting Fibers into Context

T. Schultz[1], N. Sauber[1], A. Anwander[2], H. Theisel[3], and H.-P. Seidel[1]

[1]Max Planck Institut Informatik, Saarbrücken, Germany — {schultz,sauber,hpseidel}@mpi-inf.mpg.de
[2]Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany — anwander@cbs.mpg.de
[3]Visual Computing Group, University of Magdeburg, Germany — theisel@isg.cs.uni-magdeburg.de

Abstract

Fiber tracking is a standard tool to estimate the course of major white matter tracts from diffusion tensor magnetic resonance imaging (DT-MRI) data. In this work, we aim at supporting the visual analysis of classical streamlines from fiber tracking by integrating context from anatomical data, acquired by a T_1-weighted MRI measurement. To this end, we suggest a novel visualization metaphor, which is based on data-driven deformation of geometry and has been inspired by a technique for anatomical fiber preparation known as Klingler dissection. We demonstrate that our method conveys the relation between streamlines and surrounding anatomical features more effectively than standard techniques like slice images and direct volume rendering. The method works automatically, but its GPU-based implementation allows for additional, intuitive interaction.

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

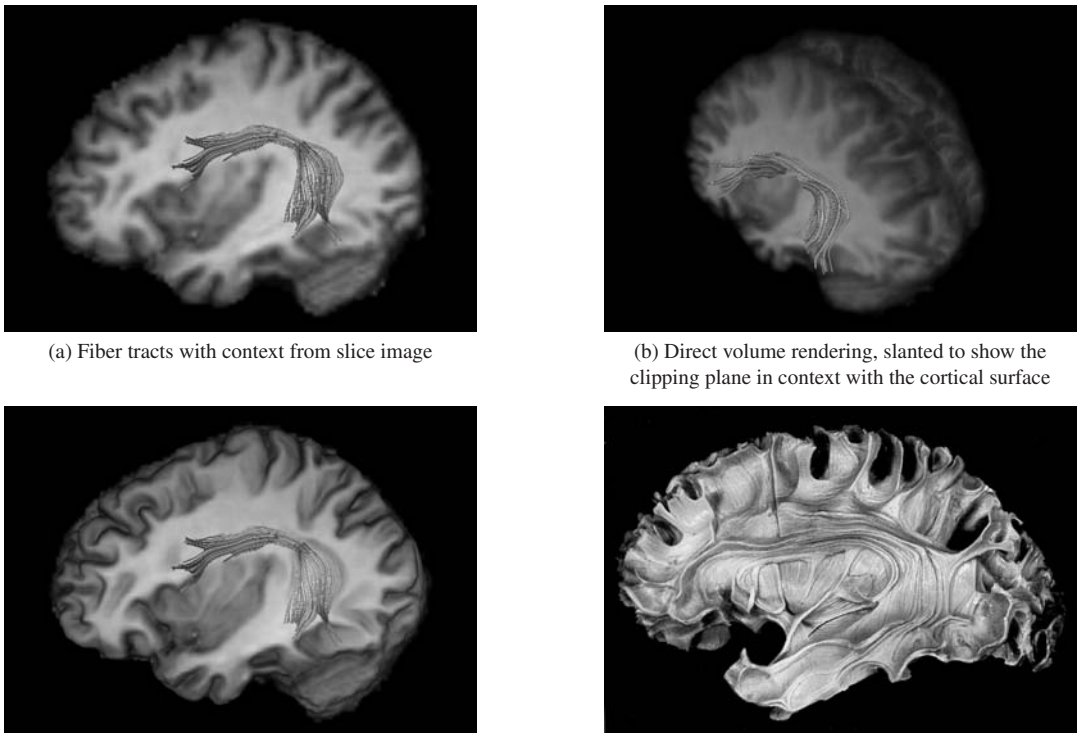
1. Introduction

Among the non-invasive methods for studying the human brain, diffusion tensor magnetic resonance imaging (DT-MRI) [BMB94] is unique in its ability to investigate the major neuronal pathways. A significant amount of research has been devoted to the development of fiber tracking algorithms, which try to reconstruct white matter tracts from DT-MRI data (cf. [MvZ02] for a review), and several approaches have been suggested for hardware-accelerated rendering of the resulting streamtubes (cf. [PFK07] and references therein). However, streamtubes alone are not very informative: Usually, the viewer is interested in understanding the spatial relation of a tract to anatomical landmarks like the gyri and sulci of the cortical surface, or to anomalies like tumors. Such anatomical features are better captured by other measurement protocols, like T_1-weighted MRI, and can be integrated by coregistration with the diffusion dataset. At the current state of the art, streamlines are combined with T_1 data using standard techniques like slice images [SPW*07] or volume renderings with clipping boxes [NGE*06].

Slice images are the simplest way to put fiber tracts into context. However, they do not convey the three-dimensional shape of structures in the T_1 data (Figure 1 (a)). Volume rendering indicates the location of the clipping plane with respect to the cortical surface, but hardly improves the perception of structures within the plane itself, due to the fact that it is mostly cutting through opaque structures (Figure 1 (b)). Both methods suffer from the fact that the streamlines are not visually connected to the T_1 data. Even when the clipping plane is placed as close as possible to the tract, the streamlines frequently appear to float in mid-air between the T_1 rendering and the viewer, and despite the shading, their exact three-dimensional trajectory is difficult to assess.

In anatomical textbooks, the problem of presenting fiber tracts is effectively addressed in the illustrations produced by Ludwig and Klingler [LK56]. In Klingler's method for fiber tract dissection, the brain first undergoes a preparation process which includes freezing to spread the fibers apart. Afterwards, it is possible to carefully scratch away tissue from one side to follow the course of fiber bundles. This leads to a relief-like surface in which the desired tract is naturally surrounded by its anatomical context (cf. Figure 1 (d)).

In this work, our goal is to generate renderings that put streamlines from DT-MRI data into context with a coregistered T_1-weighted dataset, such that spatial relations between both become apparent. Our method works by deforming a cutting plane through the T_1 data, similar to the way in which the final surface in a Klingler dissection is formed by scratching away tissue. Thus, we refer to it as "virtual Klingler dissection".



(a) Fiber tracts with context from slice image

(b) Direct volume rendering, slanted to show the clipping plane in context with the cortical surface

(c) Virtual Klingler dissection

(d) Example of a Klingler dissection, from [LK56]

Figure 1: *Compared to slice images (a) and direct volume rendering (b), our method for fused visualization of streamlines and T_1 data (c) both relates fiber tracts more clearly to the anatomy and gives a more plastic impression of the cut T_1 volume*

The example result in Figure 1 (c) shows the superior longitudinal fasciculus in its anatomical context: Through the deformed geometry, structures in the T_1 data appear more plastic than with standard methods. Moreover, streamlines produce a visible dent and are rendered more transparently where they are close to the surface, which visually connects them to the T_1 rendering and supports perception of their trajectory. When comparing our result to the real Klingler dissection, one should bear in mind that it is based on an MRI measurement of limited resolution, not on a photograph of a post-mortem preparation, and that the cerebellum (at the bottom right of our image) has been removed by Klingler.

This paper is organized as follows: First, we review related work (Section 2) and give an overview of the steps which are necessary to perform a virtual Klingler dissection (Section 3). Then, we provide details on the use of deformed geometry for visualization (Section 4) and describe implementation issues (Section 5). Finally, we present and discuss a number of results (Section 6), before we conclude the paper and point out directions for future work (Section 7).

2. Related Work

Structural MRI data was previously combined with fiber tracts by Park et al. [PKW*04], who merge streamlines with surfaces from an explicit gray matter reconstruction. This focuses on the endpoints of the fibers, while we are interested in illustrating their full course. Moreover, our method does not rely on an exact segmentation, which requires several hours of pre-computation and manual editing. Catani et al. [CHPJ02] have presented a work with a similar title as ours, but with an entirely different focus: They reconstruct a number of major fiber tracts from DT-MRI data which agree with the results of postmortem studies. To provide context for their streamtube renderings, they employ simple slice images of fractional anisotropy maps.

At the core of our method is the data-driven deformation of a surface. In computer vision, level sets [Set01, OF03] are a standard tool for this task. Since there is no need to handle topological changes in our context, we prefer a faster and simpler approach based on displacing vertices along a vector field. Vector fields have previously been used for mesh deformation by von Funck et al. [vFTS06]. However, their goal is interactive modeling, not visualization, and there are significant differences in how the vector fields are defined. Among others, their vector fields are analytically defined, while ours rely on a sampled representation of volume data.

Deformations which mimic the effect of anatomical dissections have recently been studied by Correa et al.

[CSC06], but they have focused on illustrative visualization of single datasets, and their method is based on volume deformation, while ours works on explicit geometry.

3. Principles of Virtual Klingler Dissection

In this work, we assume that the fibers of interest have been extracted using any of the established tractography methods and are given as input. The tracts shown in our experiments have been generated by the tensorlines algorithm [WKL99]. To put them into context with anatomical T_1 data, our method takes the following steps:

1. Cut the brain along a plane which is aligned with the streamlines.
2. Deform the plane such that streamlines behind it are revealed and that it visually emphasizes features in a coregistered T_1 data set.
3. Volume texture the plane with the T_1 data and render the streamlines on top of it, using variable transparency to convey their proximity.

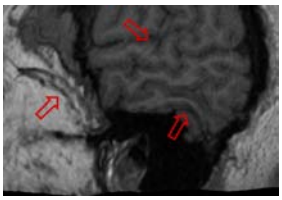
These three steps execute fully automatically in less than a second and in many cases, they already yield a satisfactory rendering. However, our implementation additionally allows the user to modify the result, both by adjusting parameters and by direct interaction with the surface.

4. Visualization by Data-driven Surface Deformation

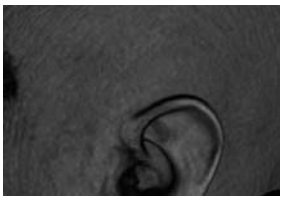
Inspired by the way in which surfaces evolve in the course of a dissection, we suggest a novel metaphor for volume visualization, in which the volume is thought to possess mechanical resistance. When the initial cutting plane is moved in some direction, the spatially varying resistance deforms the geometry. Similar to direct volume rendering (DVR), in which a transfer function assigns optical properties like color and opacity to materials in the volume, the resistance of the material is defined as a function of the data.

From this scalar resistance measure $r \in [0,1]$, we define an effective velocity $\mathbf{v}$, parallel to the original surface normal, at which a massless particle may traverse the volume. Its magnitude is given by $\|\mathbf{v}\| = 1 - r$. Formulating the problem in terms of a velocity field allows us to employ a standard tool for flow visualization: One way to visualize a 3D flow is to release a surface into the fluid at some instant, to let it move with the flow, and to observe the evolution of the resulting *time surface* with time t. Our deforming geometry is given as such a time surface, with the velocity defined by the transfer function. The fact that the velocity is non-negative implies that, like in a real Klingler dissection, we always remove material, never add any.

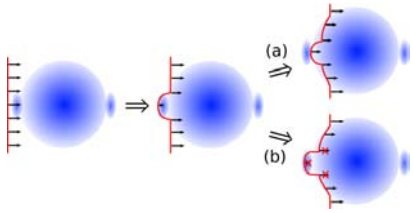
Like an undeformed cutting plane, the final surface is textured with the local values of the volume. However, this only results in an expressive rendering when the shape of the generated surface has a clear connection to its texture. After large integration times, when the surface has been pushed through significant structures, this may no longer be the case: Figure 2 (a) presents a closeup of a human head, in which the surface was shaped by its way through skin, skull, and blood vessels. Since the resulting features are not related to the displayed brain tissue, they appear as shading artifacts.



(a) Artifacts in a surface at integration time $t = 40$



(b) A resistance threshold stops the surface at the skin layer



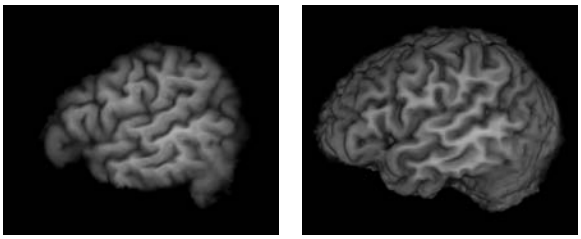
(c) Schematic illustration of the surface evolution leading to Figures (a) and (b). The head is now viewed from the top

Figure 2: *Pushing a surface through structures like skull and blood vessels can lead to shading artifacts (a) which are avoided by limiting the total resistance (b)*

To avoid this problem, we limit the total resistance which a surface point may overcome, i.e., a point on the time surface is stopped when the integral $\int_0^s r(u)du$ along its path reaches a threshold θ_r. We have fixed this resistance threshold empirically at $\theta_r = 0.7$. Figure 2 (b) demonstrates that the threshold avoids the artifacts by stopping the surface at the skin layer. Figure 2 (c) illustrates the integration process in both cases: Crossed out vectors denote velocities which are ignored because of the threshold.

4.1. Relation to Standard Methods

Even though the deformation process creates geometry from volumetric data, our method is more closely related to direct volume rendering [DCH88] than it is to surface extraction methods like isosurfacing [LC87]. It is the goal of surface extraction to reconstruct the geometry of some object of interest, like the contour of an organ or a tumor. Thus, rendering the extracted shape alone already provides an expressive visualization. In contrast to this, our deformed geometry is only used to select and shade a part of the volume. However, this process does not simply result in an improved shading of the original plane (which could be achieved using bump mapping or 2D image filters), but reveals parts of the volume which are behind it (cf. Figure 3).



(a) (b)

Figure 3: *Unlike a simple 2D embossment filter (a), our method reveals structures behind the original plane (b)*

It deserves further discussion whether introducing geometry which does not correspond to a boundary in the volume improves its visualization or rather leads to a false perception of shape. The effectiveness of a Klingler dissection does not suffer from the fact that the resulting surface is made by the hands of an anatomist rather than a natural boundary in the living brain. However, a dissection is created by an expert, while our surface deformation happens automatically, based on properties of the volume. In this respect, a more suitable analogy is sandblasting an object which is composed of different materials: Since softer materials erode faster than harder ones, this turns a planar surface into a relief. Even though the evolving surface is not a natural boundary, we can expect its shape to reflect the structure of the object.

Experimentally, we found the resistance threshold described in the previous section essential for obtaining artifact-free results. Interestingly, this threshold creates a theoretical link to thin slab volume rendering [YNR96]. Optical models for direct volume rendering typically include an absorption term which attenuates light intensity $I(s)$ depending on a spatially varying extinction coefficient $\tau(u)$ and distance s [Max95]:

$$I(s) = I_0 \cdot \mathrm{e}^{-\int_0^s \tau(u)du} \tag{1}$$

Assume we volume render the slab from which we created the deformed geometry using an orthographic projection perpendicular to the slab and using our resistance term $r(u)$ as the extinction coefficient $\tau(u)$. Then, the distance at which we stopped the surface deformation because of the resistance threshold ($\int_0^s r(u)du = \theta_r$) coincides with the point at which transparency in the volume rendering reaches $T(s) = \mathrm{e}^{-\theta_r}$. In particular, the selected threshold $\theta_r = 0.7$ corresponds to transparency $T = 0.5$. This means that in the limit, our deformed geometry converges to a surface of constant accumulated opacity. So far, such surfaces have not been extracted explicitly, but previous work [RSK06] has demonstrated how the points of accumulated opacity $A = 0.95$ can be used to separate semantic layers in certain types of data.

This insight only establishes a similarity to volume ray casting, not an equivalence: Our surface deformation is much faster than a full evaluation of the volume rendering integral, since the integration does not have to be performed per-pixel and does not involve a lighting computation in each step. Also, the visual appearance differs, as observed in Figure 1 (c): Since we do not perform a compositing, the surface has a clear appearance, while the volume rendering looks more blurry. Moreover, the lighting based on the normal of the surface produces a plastic impression, while the volume rendering has a flat look, which is typical of interfaces to clipping geometry, due to the fact that correct lighting in such areas needs to employ the normal of the geometry rather than the gradient of the volume [WEE03]. Note that Figures 1 (a)–(c) show the same T_1 data with the same resolution and trilinear interpolation; the improved impression of Figure 1 (c) is entirely due to the introduced geometry.

4.2. Mesh-based Implementation

When the surface is represented as a classic triangular mesh, with vertices m_i at initial positions $\mathbf{x}_0(m_i)$, deformation is performed by solving the following initial value problem:

$$\frac{d\mathbf{x}(t;m_i)}{dt} = \mathbf{v}(\mathbf{x}(t;m_i)) \qquad \mathbf{x}(t_0;m_i) = \mathbf{x}_0(m_i) \tag{2}$$

Equation (2) describes advection of a massless particle along a static flow field $\mathbf{v}(\mathbf{x})$ over time t. To solve it numerically, we have tried both fixed stepsize Euler integration and an adaptive fifth-order Runge-Kutta scheme [PTVF02], set for 0.5% accuracy in the final result. In our experiments, both methods achieved similar efficiency and visual results. To speed up the computation, we implemented a GPU solver similar to the one described by Krüger et al. [KKKW05].

In strongly deformed regions, it can become necessary to refine the sampling of the mesh. To this end, we employ a remeshing scheme similar to the one by von Funck et al. [vFTS06], which splits edges that have become too long on the undeformed mesh, and integrates the new vertices along the vector field.

Finally, using equilateral triangles instead of a triangulated rectangular grid for the initial sampling (cf. [MS05]) allowed us to reduce the vertex count and associated integration costs by around 20% at a comparable visual quality.

4.3. Implementation as Height Field

Since we are only considering planar starting geometry in this work and restrict ourselves to deformations in orthogonal direction, the results can alternatively be represented as height values over the original surface. It is possible to store the height field as a texture in graphics memory and to use vertex buffers along with the programmable vertex units of modern GPUs to rapidly transform it into textured geometry. We also implemented this approach, following the description of McGuire and Sibley [MS05].

Table 1 compares the performance of both implementations on a 2 GHz Athlon 64 with a GeForce 6600 graphics

Slice Size	Vertices	fps GPU	fps CPU
93×116	49 648 / *27 120*	36.2 / *13.3*	6.0 / *7.3*
256×156	103 680 / *44 521*	18.5 / *7.8*	2.6 / *3.7*
256×256	170 112 / *66 998*	11.7 / *6.7*	1.7 / *2.4*

Table 1: *Surface deformation and rendering can be done in real time. Results of the mesh-based approach are in italics*

board. Reported timings include surface deformation to integration time $t = 10$ (which was used for all examples in this paper), normal estimation, and rendering on a 1300×1000 viewport. Results of the mesh-based approach are given in italics, those of the height field are in normal print. In both GPU-based implementations, the user may change the starting geometry and observe the modified result in real time.

Since the height field implementation currently does not involve a refinement step, it has to employ a much finer sampling to avoid shading artifacts. However, the results indicate that the overhead from remeshing and GPU–CPU communication clearly outweigh the reduced vertex count when the GPU is used for integration. When all processing is done on the CPU, using an adaptive mesh is slightly more efficient.

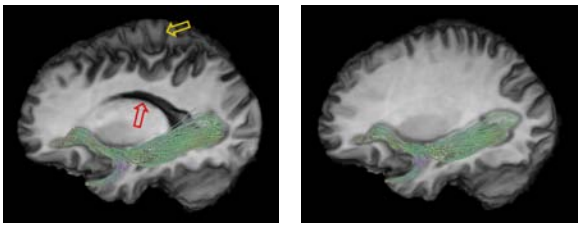
5. Implementing the Individual Stages

5.1. Finding a Suitable Cutting Plane

The initial cutting plane should be well-aligned with the streamlines. On the other hand, physicians are very much used to inspecting axis-aligned slices. A suitable cutting plane will offer a compromise between the partly contradictory goals of streamline and axis alignment. This is reflected by real Klingler dissections, which are usually close to an axis-aligned view, but rarely coincide precisely with it.

Assuming an evenly-spaced discretization of the streamlines, we can reduce them to their vertices to reduce the problem of finding an initial cutting plane to the standard task of fitting a plane to a point cloud. However, the given streamlines will not in general be closely aligned to a plane, so we need to employ a robust estimator which is tolerant against gross outliers. In computer vision, random sample consensus (RANSAC) [FB81] is a popular tool for such tasks. It repeatedly uses a minimum set of random samples to parametrize the model (i.e., three points in case of a plane), estimates the quality of the fit by counting the number of points that are within a predefined distance to the resulting model, and stores the best result. Once a sufficiently good initial estimate has been found, the least squares problem is solved on the inlier points only.

This simple procedure relies on the fact that it will sooner or later draw three inliers from the point cloud and can use the resulting plane to filter out outliers. Moreover, it easily allows us to integrate the preference for axis-aligned views by evaluating planes by a score $S = C \cdot \max_i |\mathbf{n} \cdot \mathbf{e}_i|$, where C is the number of points within 3 mm around the plane, $\mathbf{n}$ is the surface normal, and $\mathbf{e}_i$ are the axes. To ensure interactive response, we let RANSAC run for a fixed period of time (0.5 s) and use the best result so far.



(a) (b)

Figure 4: *Our robust estimator (b) is less likely to propose unusable views than a simple least squares approach (a)*

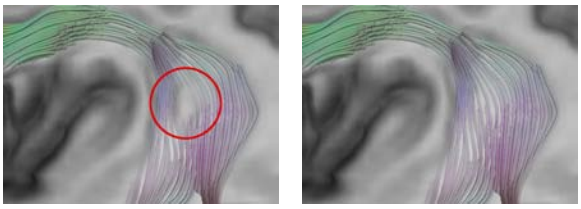
Figure 4 presents an example where our modified robust estimator is crucial for obtaining a useful result. In Subfigure (a), a tractography of the inferior fronto-occipital fasciculus and the uncinate fasciculus is shown in context of a plane which has been chosen based on a simple least squares fit on all vertices. The tract is visible, but an expert would find the specific plane, which cuts diagonally through parts of the ventricle (red arrow) and through a part of the opposite hemisphere (yellow arrow) confusing rather than helpful. In contrast, the result of our robust estimator in Subfigure (b) is more closely aligned to a standard sagittal view. If the user is still not entirely satisfied with the suggested plane, she may move and rotate it manually.

5.2. Deforming the Geometry

Deformation of the original cutting plane follows two goals: First, the surface should be retracted when streamlines are in its vicinity. This is akin to the way an anatomist would follow the course of a fiber tract in a Klingler dissection: It reveals fibers which would otherwise be occluded and introduces dents in the surface which give visual cues about the immediate proximity between T_1 data and streamlines. Second, a surface whose curvature enhances the appearance of features in the anatomical data is preferred over a flat one.

To pursue the first goal, a voxel-wise streamline density ρ is derived from the given tracts. It is approximated by normalizing the length of each streamline segment by the volume of one voxel and counting it towards the density of the voxel which contains the midpoint of the segment. The resulting field is convolved with a narrow Gaussian kernel to ensure a smooth deformation. Resistance $r = 1$ should be assigned to $\rho(u) = 0$, while r should tend to zero for $\rho(u) \to \infty$. This is accomplished by taking the difference of unity and the scaled arc tangent of $\rho(u)$.

A second transfer function g is used to take influence of the T_1 data into account. Our implementation lets the user



(a) (b)

Figure 5: *In some cases, the T_1 surface may occlude part of a fiber tract even after deformation (a). Such problems are easily resolved by a local, interactive deformation (b)*

define g as an arbitrary piecewise linear function of scalar value $f(u)$, but all demonstrated results use a simple linear mapping of f to $[0,1]$. The combined resistance $r(u)$, which is used to deform the surface within the framework of Section 4, allows it to move when either of the individual terms indicates low resistance:

$$r(u) = g\left(f(u)\right) \cdot \left[1 - \frac{2}{\pi} \arctan\left(\lambda \cdot \rho(u)\right)\right] \qquad (3)$$

To obtain a meaningful quantity, the absolute streamline density $\rho(u)$ has to be normalized by the seed point density ρ_s, which is an arbitrary parameter in fiber tracking. Additionally, it depends on the distance of the surface to surrounding fibers how much it has to deform to reveal them. The scaling parameter λ takes care of both facts. Consider the streamline vertices in some corridor around the surface and let σ be their standard deviation from the surface. Then, the following choice of λ reveals streamlines within 3σ in areas where $\rho(u) = \rho_s$:

$$\lambda = \frac{1}{\rho_s} \tan\left(\frac{\pi}{2} \frac{3\sigma}{3\sigma + \theta_r}\right) \qquad (4)$$

If the user is not entirely satisfied with the deformation, she can alter the suggested settings of g and λ. Moreover, the surface can be further deformed interactively, by clicking and moving the mouse over it. In this case, integration is continued with resistance threshold θ_r disabled, but resistance r still in effect. To keep the deformation local, integration time t decreases with distance from the surface point below the mouse pointer. For example, this intuitive tool allowed us to transform Figure 5 (a), where the automatic deformation had failed to reveal a small part of a tract in a region of low streamline density, to Figure 5 (b), which resolves this problem, within few seconds.

5.3. Distance Cueing

Depth cueing is a standard computer graphics technique which supports depth perception by blending object colors with the background, depending on the distance from the viewer. We adopt this idea to visually connect the streamlines with the T_1 surface by blending them with the surface when they come close to it. This is done by assigning an opacity value which decreases linearly with distance to the surface when it is less than a threshold $\theta_d = 5\,\text{mm}$.

The approximate distance of a point to the surface is found by computing the orthogonal distance to the initial plane and subtracting the interpolated height value from the height field representation. If a mesh representation is used, it is first converted into a height field by rendering it to an off-screen buffer and reading back the OpenGL depth buffer.

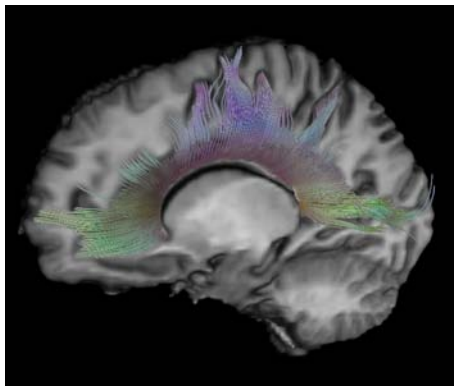
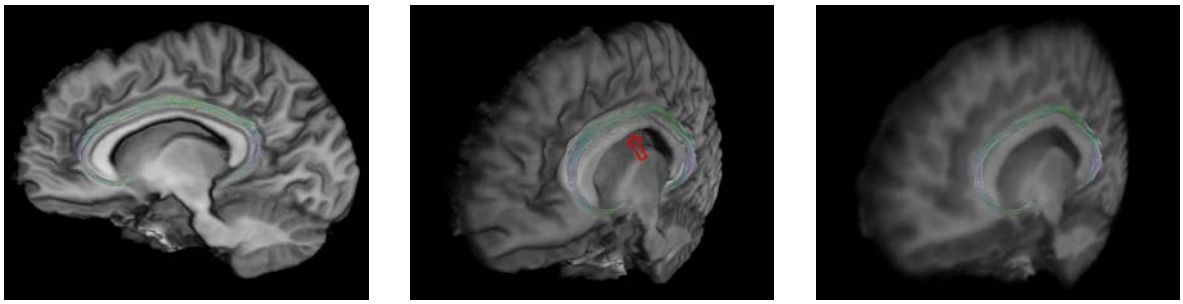


Figure 6: *Despite the non-planar shape of the corpus callosum, fitting a cutting plane produces a sensible result*

6. Results and Discussion

The cutting planes suggested by the program could be used without modification in Figures 4 (b), 5, 6, 7, and 8. To test the robustness of the estimator, we tried it on two highly non-planar inputs: For fibers from the corpus callosum, which has a saddle-shaped geometry, the estimator suggested a plane that provides good context for the left half (Figure 6) and clearly shows how the tract passes above the ventricle and projects to the cortical surface. In a tractography of both cingulum bundles, the estimator chose the vertices of the right bundle as inliers and generated suitable context for them (not shown, but very similar to Figure 7). In both cases, a least squares approach produced a result near the mid-sagittal plane, at a high distance to almost all streamlines.

Within this project, our experience with the novel visualization metaphor suggested in Section 4 has been encouraging. With the resistance threshold θ_r enabled, we have not observed any cases in which the deformed geometry would have introduced artifacts or a false perception of shape, and even though the interpretation as a surface of constant opacity only holds when viewed in perpendicular direction, we found that the surface can be rotated without losing its expressiveness. For example, Figure 7 shows that changing the viewpoint can help to understand how the cingulum bundle runs over the top of the corpus callosum (red arrow). Such an exploration is not limited to the framerates reported in



(a) Medial view on the cingulum bundle (b) Rotated view on the same surface (c) Clipped DVR for comparison

Figure 7: *Rotating the created geometry interactively (from (a) to (b)) supports the spatial impression. Due to its flat appearance, rotating a clipped volume rendering is less helpful (c)*

Table 1, since it does not involve a re-integration. For direct comparison, Subfigure (c) presents a clipped volume rendering from the same viewpoint.

The resistance term suggested in Equation (3) proved effective for our application. In particular, we found the combination of a streamline-based and a T_1-based resistance to be superior to a deformation which only depends on one of the input datasets. Figure 8 illustrates this by showing results based on streamline density (a) and T_1 value (b) only, as well as the proposed combined term (c).

The specified defaults for the scalar transfer function g and the streamline density weight λ were appropriate in most cases. Only in Figure 8 (c), they failed to reveal a small portion of the tract, which was easily fixed interactively (cf. closeup in Figure 5). Overall, our framework produces usable results automatically, but makes it easy for the user to refine them to her liking.

7. Conclusion and Future Work

Understanding the spatial relation of streamlines from DT-MRI to anatomical structures in T_1 data is an important aspect in the visual analysis of fiber tracking results. In this work, we proposed a fused visualization method, in which both types of data interact to reveal their spatial relation. We presented an efficient implementation and results on four different fiber tracts in the human brain. Our results have been validated by one of our co-authors, who is a domain expert. Additionally, a physician who is working with DT-MRI data, but not affiliated with our team has confirmed that our renderings convey an impression of the relative fiber positions, while standard methods fail to connect the streamlines to their context.

Part of our contribution lies in the way in which established methods, like RANSAC or depth cueing, are modified and combined to solve a specific problem. However, the data-driven surface deformation, which is at the core of our method, is a novel visualization tool in itself. Despite initial doubts about the appropriateness of creating "artificial" geometry for visualization, the valid and expressive results achieved in our experiments support the arguments which are in favor of such an approach. We are confident that the method from Section 4 can help to create more plastic renderings in other situations where the flat appearance caused by clipping geometry may be undesired.

To improve the virtual Klingler dissection further, it would be interesting to consider non-planar starting surfaces to fit the complex geometry of some neuronal pathways and to facilitate illustrating multiple tracts in a single rendering.

Acknowledgements

We would like to thank Carsten Stoll (MPI Informatik) for advice on GPU programming and Michael Scheel (Charité, Berlin, Germany) for providing feedback on our results. We are grateful to Timm Wetzel and Enrico Kaden (MPI CBS) for acquisition and preprocessing of the DWI images.

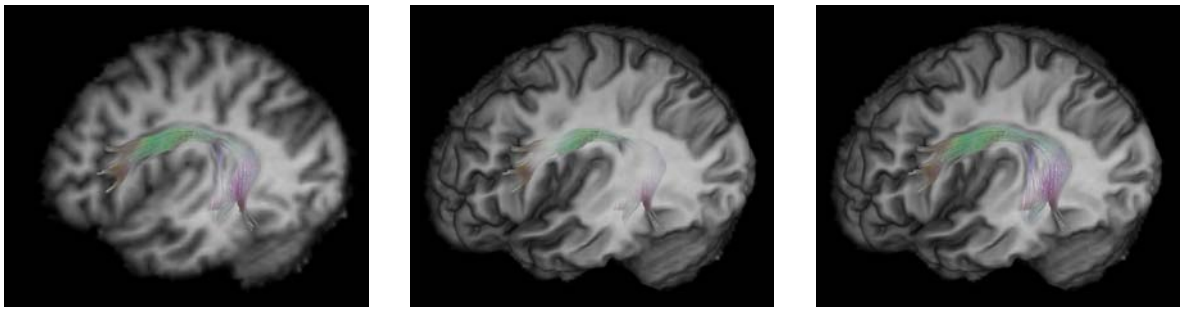
Figure 1 (d) is reproduced from [LK56] with the kind permission of S. Karger AG, Basel. Figure 2 uses "Bruce Gooch's Brain". Figure 3 (a) has been created using the GIMP. Our implementation makes use of the teem library. This work has partially been funded by the Max Planck Center for Visual Computing and Communication (MPC-VCC).

References

[BMB94] BASSER P. J., MATTIELLO J., BIHAN D. L.: Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance B*, 103 (1994), 247–254.

[CHPJ02] CATANI M., HOWARD R. J., PAJEVIC S., JONES D. K.: Virtual in vivo interactive dissection of white matter fasciculi in the human brain. *NeuroImage 17* (2002), 77–94.

[CSC06] CORREA C. D., SILVER D., CHEN M.: Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1069–1076.



(a) Deformation based on fibers only (b) Deformation based on T_1 data only (c) Proposed combined method

Figure 8: *Deforming the cutting plane based on streamline density (a) or T_1 value (b) alone is less effective in conveying their relation than a combined approach (c)*

[DCH88] DREBIN R. A., CARPENTER L., HANRAHAN P.: Volume rendering. In *Proc. ACM SIGGRAPH* (1988), pp. 65–74.

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[KKKW05] KRÜGER J., KIPFER P., KONDRATIEVA P., WESTERMANN R.: A particle system for interactive visualization of 3D flows. *IEEE Transactions on Visualization and Computer Graphics 11*, 6 (2005), 744–756.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. ACM SIGGRAPH* (1987), pp. 163–169.

[LK56] LUDWIG E., KLINGLER L.: *Atlas cerebri humani*. S. Karger AG, Basel, 1956.

[Max95] MAX N.: Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 99–108.

[MS05] MCGUIRE M., SIBLEY P.: *A Heightfield on an Isometric Grid*. Tech. Rep. CS-05-14, Department of Computer Science at Brown University, 2005.

[MvZ02] MORI S., VAN ZIJL P. C.: Fiber tracking: principles and strategies – a technical review. *NMR in Biomedicine 15* (2002), 468–480.

[NGE*06] NIMSKY C., GANSLANDT O., ENDERS F., MERHOF D., HAMMEN T., BUCHFELDER M.: Visualization strategies for major white matter tracts for intraoperative use. *International Journal of Computer Assisted Radiology and Surgery 1*, 1 (2006), 13–22.

[OF03] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153 of *Applied Mathematical Sciences*. Springer, 2003.

[PFK07] PETROVIC V., FALLON J., KUESTER F.: Visualizing whole-brain DTI tractography with GPU-based tuboids and LoD management. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1488–1495.

[PKW*04] PARK H.-J., KUBICKI M., WESTIN C.-F., TALOS I.-F., BRUN A., PEIPER S., KIKINIS R., JOLESZ F., MCCARLEY R., SHENTON M.: Method for combining information from white matter tracking and gray matter parcellation. *American Journal of Neuroradiology 25* (2004), 1318–1324.

[PTVF02] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in C++: The Art of Scientific Computing, Second Edition*. Cambridge Univ. Press, 2002.

[RSK06] REZK-SALAMA C., KOLB A.: Opacity peeling for direct volume rendering. *Computer Graphics Forum (Proc. Eurographics) 25*, 3 (2006), 597–606.

[Set01] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods*, 2nd ed. Cambr. Univ. Press, 2001.

[SPW*07] SCHMAHMANN J. D., PANDYA D. N., WANG R., DAI G., D'ARCEUIL H. E., DE CRESPIGNY A. J., WEDEEN V. J.: Association fibre pathways of the brain: parallel observations from diffusion spectrum imaging and autoradiography. *Brain 130*, 3 (2007), 630–653.

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH) 25*, 3 (2006), 1118–1125.

[WEE03] WEISKOPF D., ENGEL K., ERTL T.: Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 298–312.

[WKL99] WEINSTEIN D., KINDLMANN G., LUNDBERG E.: Tensorlines: advection-diffusion based propagation through diffusion tensor fields. In *Proc. IEEE Visualization* (1999), pp. 249–253.

[YNR96] YEN S. Y., NAPEL S., RUBIN G. D.: Fast sliding thin slab volume visualization. In *Proc. Symposium on Volume Visualization* (1996), pp. 79–86.